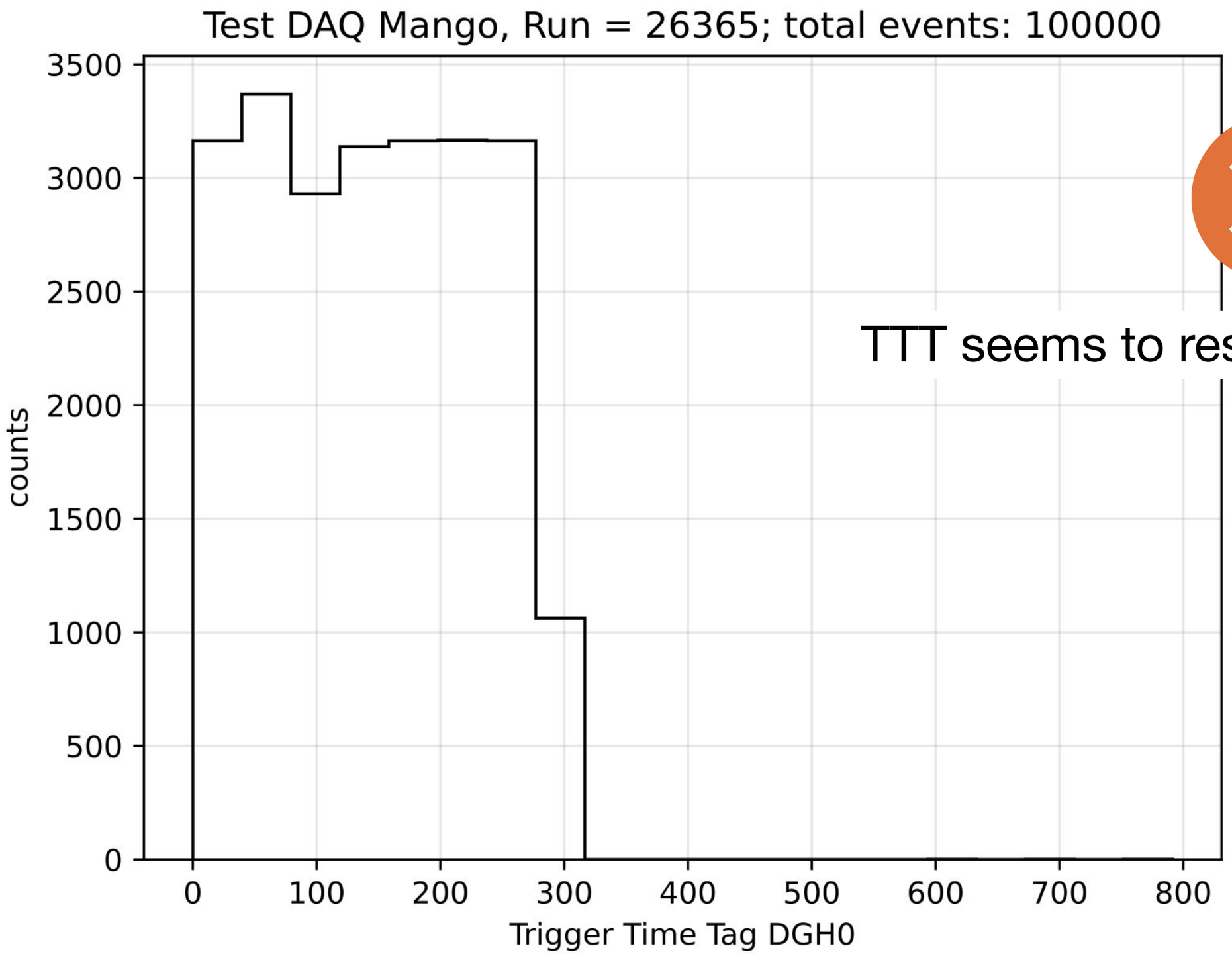

# Mango DAQ update

## Trigger Time Tag improvements

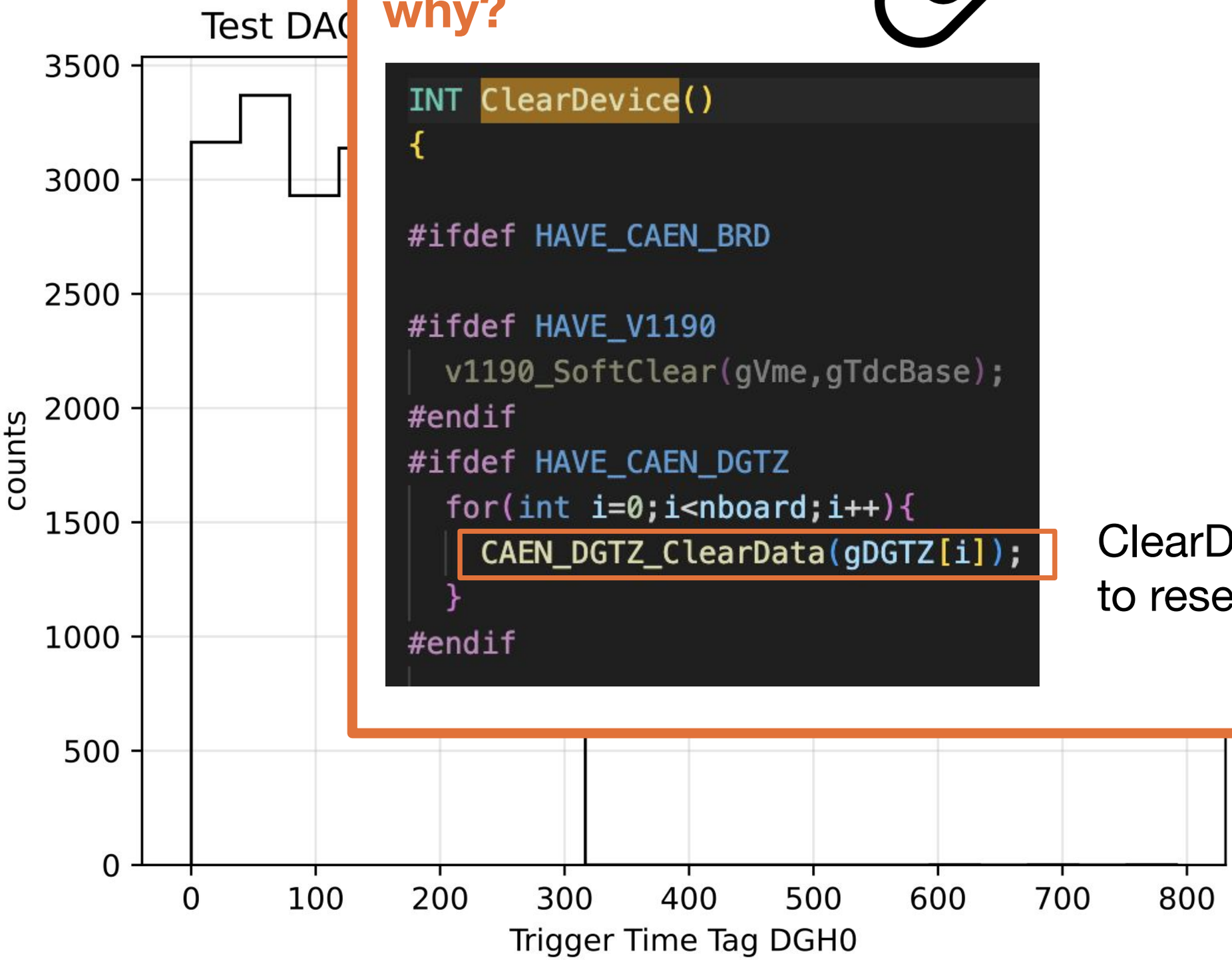

Vito Monno 22/04/2025

# Previously on **Mango:** TTT distribution




TTT seems to reset each photo!

**Test on this:** TTT distribution




## why?

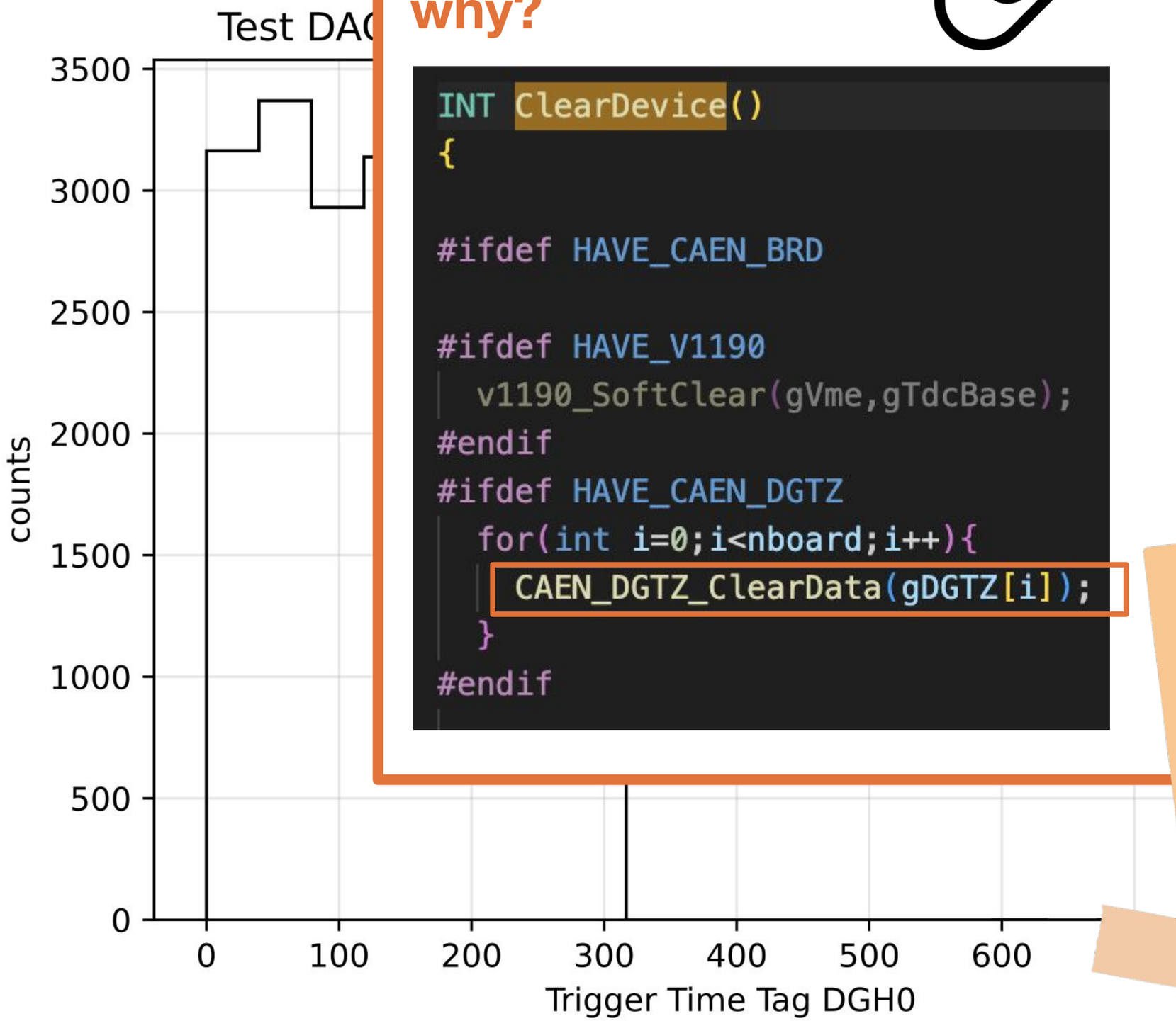
```
INT ClearDevice()
{

#ifdef HAVE_CAEN_BRD

#ifdef HAVE_V1190
  v1190_SoftClear(gVme,gTdcBase);
#endif
#ifdef HAVE_CAEN_DGTZ
  for(int i=0;i<nboard;i++){
    CAEN_DGTZ_ClearData(gDGTZ[i]);
  }
#endif
```

ClearData seems to reset it

**Test on this:** TTT distribution




## why?

```
INT ClearDevice()
{

#ifdef HAVE_CAEN_BRD

#ifdef HAVE_V1190
  v1190_SoftClear(gVme,gTdcBase);
#endif
#ifdef HAVE_CAEN_DGTZ
  for(int i=0;i<nboard;i++){
    CAEN_DGTZ_ClearData(gDGTZ[i]);
  }
#endif
```

***we opened a ticket to CAEN***

# *CAEN*'s answer:

Message from **Support** 11/04/2025 16:18:55

Il buffer viene svuotato al momento del readout, usando la funzione ReadData() della libreria CAENDigitizer.
Non è necessario usare la ClearData() durante l'acquisizione.
Mentre la ReadData() svuota il buffer, questo viene riempito ovviamente da nuovi eventi. Si possono accettare nuovi trigger solo se c'è spazio nel buffer, cioè se la rilettura è veloce rispetto al riempimento del buffer stesso (come detto in precedenza).

So, we just need to avoid using `ClearData()` when used in `read_event()`

( we want to still use it in `begin_of_run()` )

# Solution:

`ClearData()` is called in `ClearDevice()` function

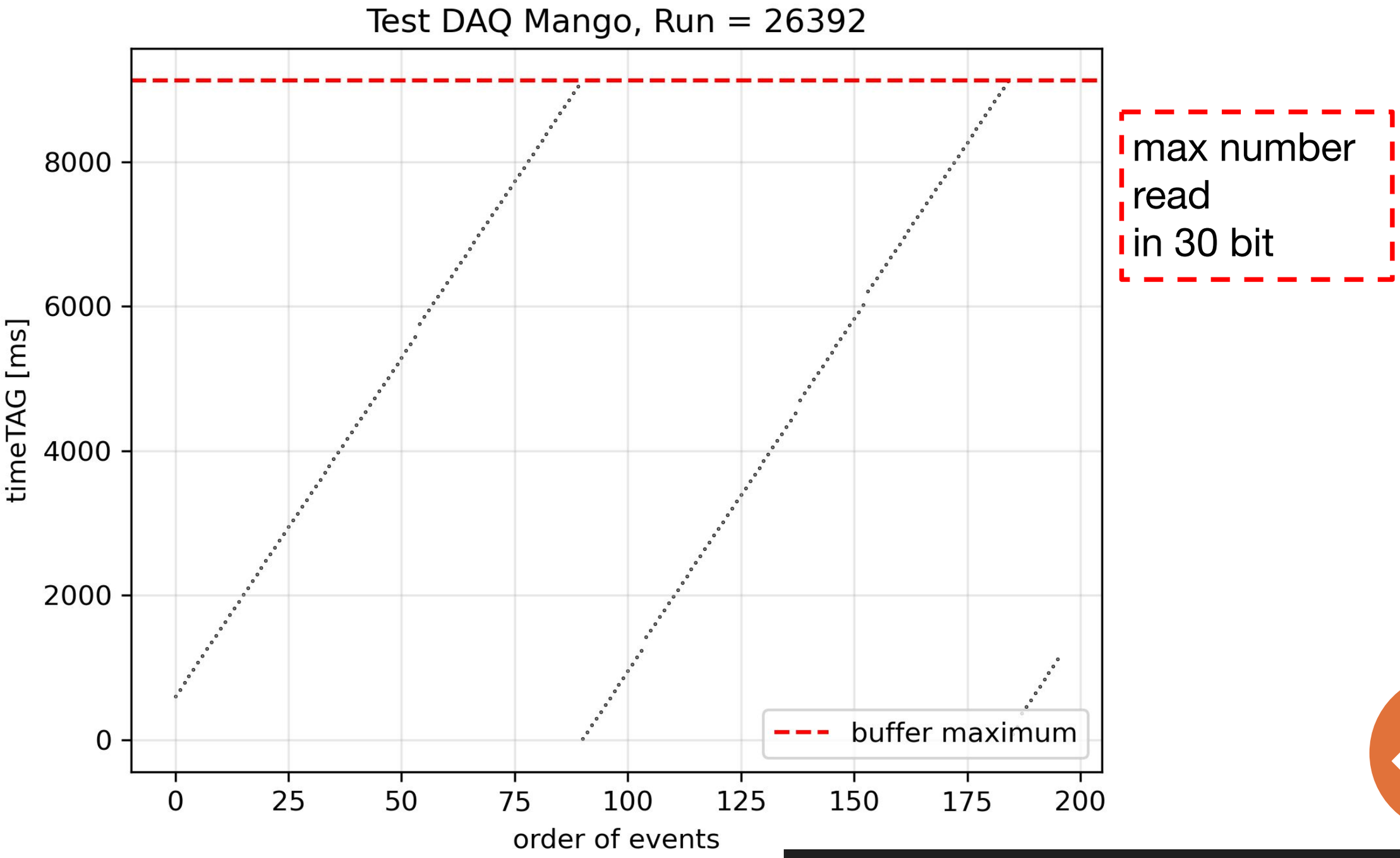
we added a BOOLEAN argument to `ClearDevice()`
which allow control on `ClearData()` use.

```
INT ClearDevice(BOOL clear_dgtz_data)
{
```

```
#ifdef HAVE_CAEN_DGTZ
   if (clear_dgtz_data) {
      for(int i=0;i<nboard;i++){
         CAEN_DGTZ_ClearData(gDGTZ[i]);
      }
   }
#endif
```

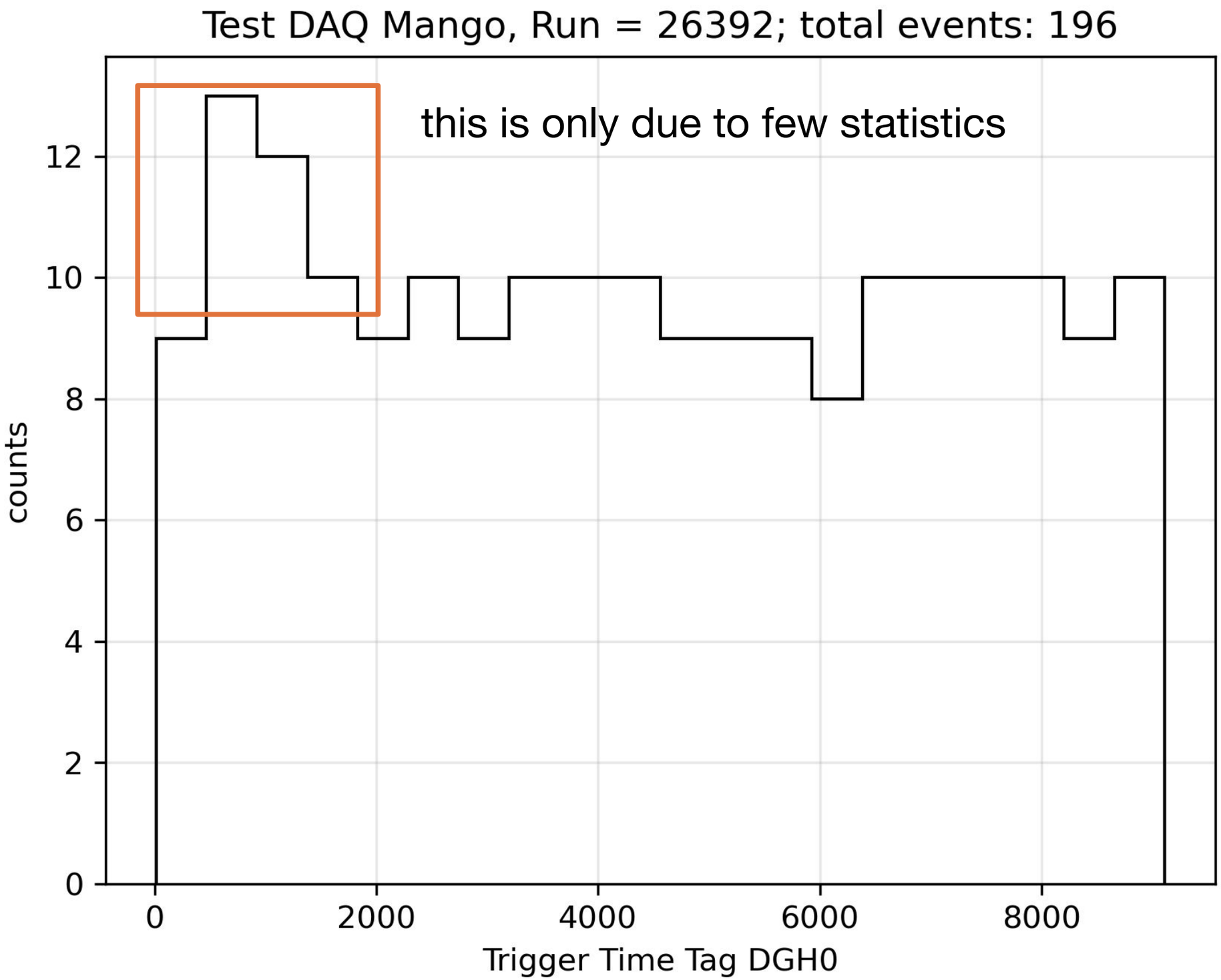
➔ ClearData( ) is called in begin_of_run( )
➔ ClearData( ) is not called in read_event( )

# **Test on this:** TTT distribution




```
the max readable timeTAG is:  9.1268
the max read timeTAG is: 9.1145
```

**Test on this:** TTT distribution



# New Goal: 60 bit TTT

# **First of all:** getting dgtz fw version

```
//VITO: print the firmware version
CAEN_DGTZ_BoardInfo_t BoardInfo;

for(int i=0;i<nboard;i++){
  CAEN_DGTZ_GetInfo(gDGTZ[i], &BoardInfo);
  std::cerr << "Digitizer " << i << " (" << BoardName[i] << ") - Firmware: " << BoardInfo.ROC_FirmwareRel << " / " << BoardInfo.AMC_FirmwareRel << std::endl;
}
```

```
cygno01@mango01:~/daq/online$ ./cygnus_fe > useless.txt
configuring dgtz...
Digitizer 0 (V1742) - Firmware: 04.21 - Build 3B12 / 01.03 - Build 4410
```

60 bit TTT is not available for this version 1.03 (need >1.06)

| | |
|---|---|
| [20] | Extended Group Trigger Time Tag flag (EGTTT). If enabled, the Group Trigger Time Tag information is extended to 60 bits (8.5 ns resolution). Referring to the Group n Data in the event structure of the digitizer and taking channel groups 0-1 as an example, the MSB of the EGTTT (common to Group0 and Group1) is the 30-bit Group Trigger Time Tag of Group1, while the LSB is the 30-bit Group Trigger Time Tag of Group0. The same for the EGTTT common to Group2 and Group3 (VME only). Options are:<br>0 = 60-bit trigger timestamp disabled (default);<br>1 = 60-bit trigger timestamp enabled.<br>NOTE: Bit[20] setting is indifferent when enabling only GROUP 0 and/or GROUP 2 (VME only), as the timestamp significant value remains at 30 bits. Instead, enabling only GROUP 1 and/or GROUP 3 (VME only), bit[20] must not be 0, otherwise the timestamp is inconsistent.<br>NOTE: This bit is valid from AMC FPGA firmware revision 1.06 on |

# **First of all:** getting dgtz fw version

```
//VITO: print the firmware version
CAEN_DGTZ_BoardInfo_t BoardInfo;

for(int i=0;i<nboard;i++){
  CAEN_DGTZ_GetInfo(gDGTZ[i], &BoardInfo);
  std::cerr << "Digitizer " << i << " (" << BoardName[i] << ") - Firmware: " << BoardInfo.ROC_FirmwareRel << " / " << BoardInfo.AMC_FirmwareRel << std::endl;
}
```

```
cygno01@mango01:~/daq/online$ ./cygnus_fe > useless.txt
configuring dgtz...
Digitizer 0 (V1742) - Firmware: 04.21 - Build 3B12 / 01.03 - Build 4410
```

60 bit TTT is not available for this version 1.03 (need >1.06)

```
cygno01@mango01:~/daq/online$ ./cygnus_fe > useless.txt
configuring dgtz...
Digitizer 0 (V1742) - Firmware: 04.29 - Build 8716 / 01.06 - Build 6530
```

**now updated :)**

# Enabling EGTTT

| | |
|---|---|
| [20] | Extended Group Trigger Time Tag flag (EGTTT). If enabled, the Group Trigger Time Tag information is extended to 60 bits (8.5 ns resolution). Referring to the Group n Data in the event structure of the digitizer and taking channel groups 0-1 as an example, the MSB of the EGTTT (common to Group0 and Group1) is the 30-bit Group Trigger Time Tag of Group1, while the LSB is the 30-bit Group Trigger Time Tag of Group0. The same for the EGTTT common to Group2 and Group3 (VME only). Options are:<br>0 = 60-bit trigger timestamp disabled (default);<br>1 = 60-bit trigger timestamp enabled.<br>NOTE: Bit[20] setting is indifferent when enabling only GROUP 0 and/or GROUP 2 (VME only), as the timestamp significant value remains at 30 bits. Instead, enabling only GROUP 1 and/or GROUP 3 (VME only), bit[20] must not be 0, otherwise the timestamp is inconsistent.<br>NOTE: This bit is valid from AMC FPGA firmware revision 1.06 on |

60 bit EGTTT

100111010101…10101010000

MSB 30 bit
Group[1]

LSB 30 bit TTT
Group[0]

# Enabling EGTTT

| | |
|---|---|
| [20] | Extended Group Trigger Time Tag flag (EGTTT). If enabled, the Group Trigger Time Tag information is extended to 60 bits (8.5 ns resolution). Referring to the Group n Data in the event structure of the digitizer and taking channel groups 0-1 as an example, the MSB of the EGTTT (common to Group0 and Group1) is the 30-bit Group Trigger Time Tag of Group1, while the LSB is the 30-bit Group Trigger Time Tag of Group0. The same for the EGTTT common to Group2 and Group3 (VME only). Options are:<br>0 = 60-bit trigger timestamp disabled (default);<br>1 = 60-bit trigger timestamp enabled.<br>NOTE: Bit[20] setting is indifferent when enabling only GROUP 0 and/or GROUP 2 (VME only), as the timestamp significant value remains at 30 bits. Instead, enabling only GROUP 1 and/or GROUP 3 (VME only), bit[20] must not be 0, otherwise the timestamp is inconsistent.<br>NOTE: This bit is valid from AMC FPGA firmware revision 1.06 on |

in `ConfigDgtz()`, bit [20] enables EGTTT:

```
//VITO: ENABLING EGTTT 60 bit
for(int i=0;i<nboard;i++){
  // Read the register than turn on the bit 20
  uint32_t enable_egttt;
  CAEN_DGTZ_ReadRegister(gDGTZ[i], 0x8000, &enable_egttt);
  enable_egttt = enable_egttt | 0x00100000;  // bit 20
  CAEN_DGTZ_WriteRegister(gDGTZ[i], 0x8004, enable_egttt);
}
```

# EGTTT in DGH0 bank

old `bank.data` :

```
Event # 0 of type ID 1 contains banks TIME, CAM0, TSP0, FID0, DIG0, DGH0
Received event with timestamp 212 containing banks TIME, CAM0, TSP0, FID0, DIG0, DGH0
1970-01-01 00:03:32, banks TIME, CAM0, TSP0, FID0, DIG0, DGH0
[       1     1742     1024       32       27     4096     1333    13107
    13107    13107    13107    13107    16384    26214    32768    32768
    32768    32768    32768    32768    32768    32768    32768    32768
    32768    32768    32768    32768    32768    32768    32768    32768
    32768    32768    32768    32768    32768    32768    32768   106518
 57176852 58178086 59179314 60180544 61181774 62182998 63184236 64185460
 65186690 66187922 67189146 68190382 69191608 70192846 71194072 72195300
 73196534 74197762 75198994 76200218 77201454 78202678 79203908 80205142
 81206372 82207608      134      735      440      109      812      491
      134      889      530      211      928      568      288      966
      696      352       19      748      415      109      774      491
      134      838      542      224      966]
```

LSB of N events

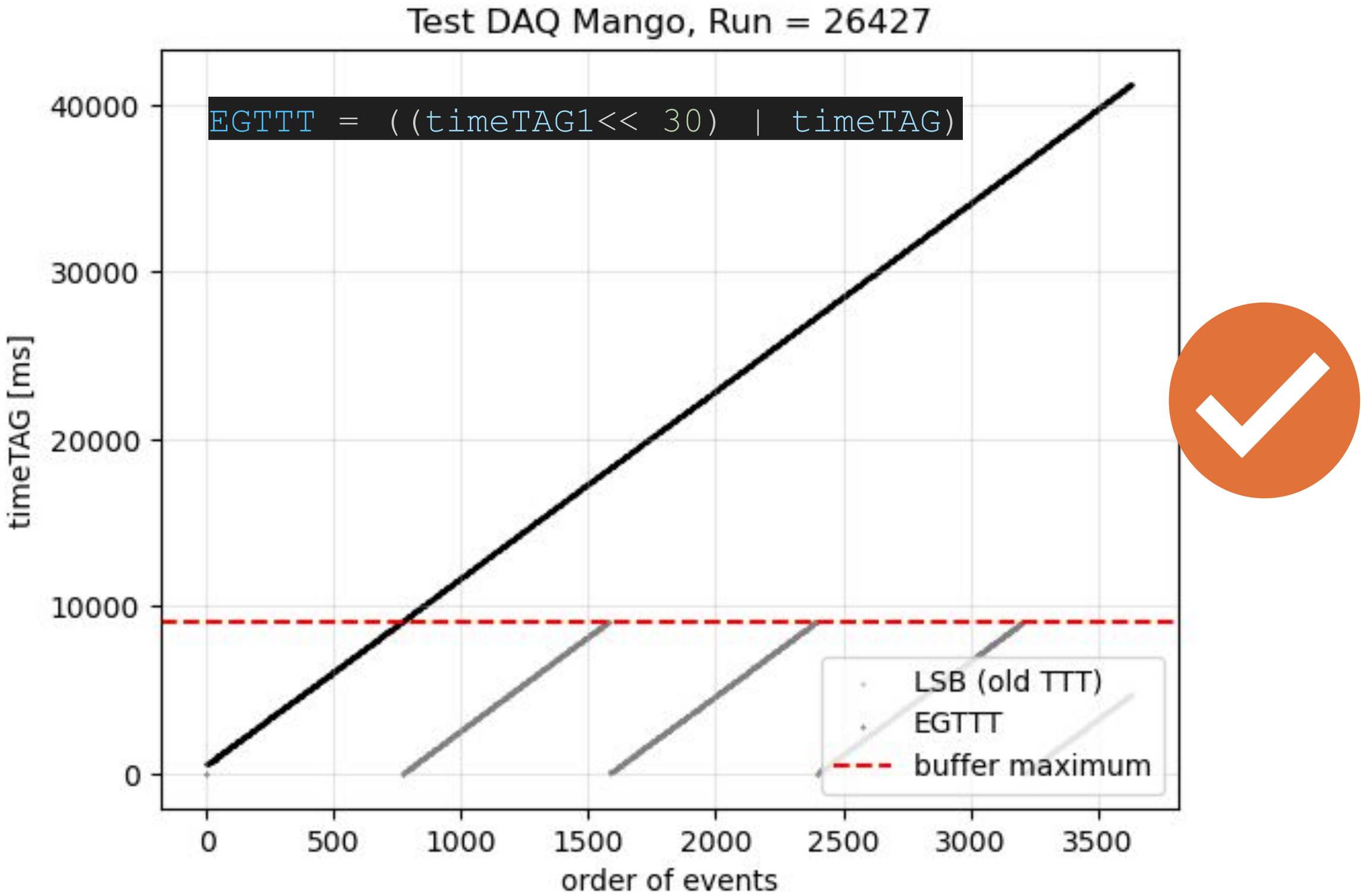
# EGTTT in DGH0 bank

new `bank.data`:

```
Event # 0 of type ID 7 contains banks TCAM
Received event with timestamp 1746522140 containing banks TCAM
2025-05-06 09:02:20, banks TCAM
Event # 1 of type ID 1 contains banks CAM0, TSP0, FID0, DIG0, DGH0
Received event with timestamp 906 containing banks CAM0, TSP0, FID0, DIG
1970-01-01 00:15:06, banks CAM0, TSP0, FID0, DIG0, DGH0
[        1      1742      1024        32        22      4096      1333
     13107     13107     13107     13107     13107     16384     26214
     32768     32768     32768     32768     32768     32768     32768
     32768     32768     32768     32768     32768     32768     32768
     32768     32768     32768     32768     32768     32768     32768
     32768     32768     32768     32768 119068308 120068208 121068106
 122068008 123067898 124067804 125067698 126067602 127067502 128067394
 129067300 130067194 131067096 132066990 133066888 134066788 135066682
 136066590 137066486 138066390 139066288 140066184         0         0
         0         0         0         0         0         0         0
         0         0         0         0         0         0         0
         0         0         0         0         0         0       427
       812       160       555       876       275       619         6
       389       722       121       465       863       185       555
       940       262       696        32       440       812       147]
```

added MSB of N events

# EGTTT:


Test DAQ Mango, Run = 26427
EGTTT = ((timeTAG1<< 30) | timeTAG)
timeTAG [ms]
40000
30000
20000
10000
0
0
500
1000
1500
2000
2500
3000
3500
order of events
LSB (old TTT)
EGTTT
buffer maximum

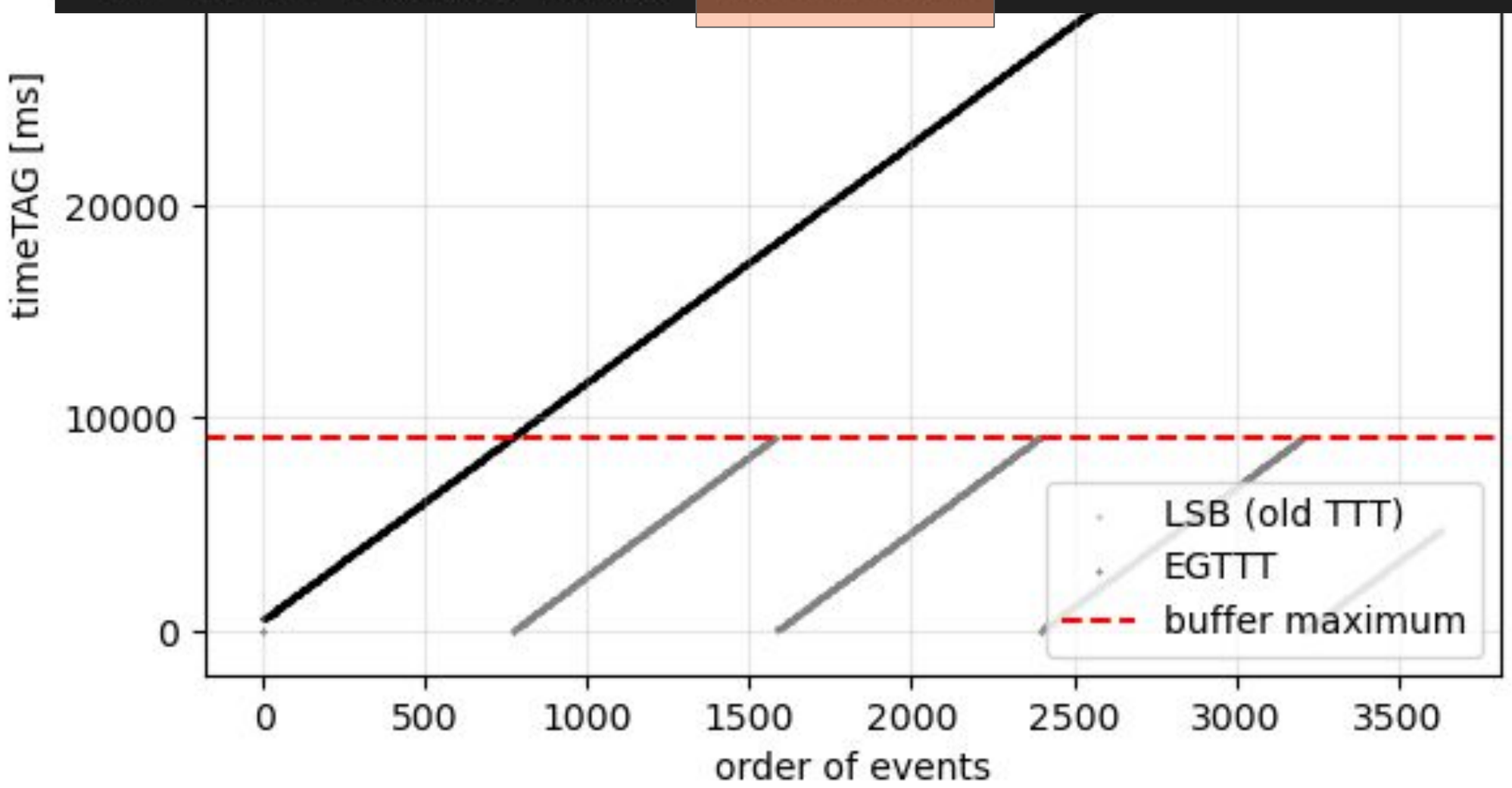
# EGTTT:

```
sec_to_year =  60*60*24*365
buffer_max_60 = (2**60 -1 )* 8.5 * 1e-9 / sec_to_year
print('Now the new max at 60 bit is: ', np.round(buffer_max_60,3), 'years')
```

✓ 0.0s

```
Now the new max at 60 bit is:  310.751 years
```

enough



# **CygnoLib:** a fix for EGTTT

```
if cam_flag:
    for bank_name, bank in event.banks.items():
        if bank_name=='DGH0': # PMTs wavform
            flag = True

            fullhead = np.array(bank.data)
            DgtzHeader = cy.daq_dgz_full2header(bank)
            #print(fullhead)
```

how get it

```
help(fullhead)
```

✓ 0.3s

```
Help on dgtz_header in module cygno object:

class dgtz_header(builtins.object)
 |  dgtz_header(a)
 |
 |  Methods defined here:
 |
 |  __getitem__(self, index)
 |
 |  __init__(self, a)
 |      Initialize self.  See help(type(self)) for accurate signature.
 |
 |  ----------------------------------------------------------------------
 |  Data descriptors defined here:
 |
 |  __dict__
 |      dictionary for instance variables
 |
 |  __weakref__
 |      list of weak references to the object
```

dgtz_header class

# cy.daq_dgz_full2header(bank)

```
DgtzHeader.__dict__
✓ 0.0s
```

```
{'ntriggers': array([26]),
 'nchannels': array([32]),
 'nsamples': array([1024]),
 'vertical_resulution': array([4096]),
 'sampling_rate': array([1333]),
 'offsets': [array([13107, 13107, 13107, 13107, 13107, 16384, 26214, 32768, 32768,
        32768, 32768, 32768, 32768, 32768, 32768, 32768, 32768, 32768,
        32768, 32768, 32768, 32768, 32768, 32768, 32768, 32768, 32768,
        32768, 32768, 32768, 32768, 32768])],
 'TTT': [array([531648030, 532647932, 533647830, 534647734, 535647636, 536647532,
        537647438, 538647332, 539647240, 540647144, 541647042, 542646944,
        543646840, 544646746, 545646636, 546646542, 547646442, 548646346,
        549646248, 550646146, 551646052, 552645952, 553645860, 554645760,
        555645660, 556645560])],
 'SIC': [array([4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4,
        4, 4, 4, 4])],
 'nBoards': 1,
 'boardNames': array([1742]),
 'itemDict': {'0': array([26]),
  '1': array([32]),
  '2': array([1024]),
  '3': array([4096]),
  '4': array([1333]),
  '5': [array([13107, 13107, 13107, 13107, 13107, 16384, 26214, 32768, 32768,
         32768, 32768, 32768, 32768, 32768, 32768, 32768, 32768, 32768,
...
         555645660, 556645560])],
  '7': [array([4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4,
         4, 4, 4, 4])],
  '8': 1,
  '9': array([1742])}}
```

MSB goes wrongly into SIC

# bank.data

```
fullhead
✓ 0.0s  Open 'fullhead' in Data Wrangler
```

```
array([       1,     1742,     1024,       32,       26,     4096,
        1333,    13107,    13107,    13107,    13107,    13107,
       16384,    26214,    32768,    32768,    32768,    32768,
       32768,    32768,    32768,    32768,    32768,    32768,
       32768,    32768,    32768,    32768,    32768,    32768,
       32768,    32768,    32768,    32768,    32768,    32768,
       32768,    32768,    32768, 531648030, 532647932, 533647830,
   534647734, 535647636, 536647532, 537647438, 538647332, 539647240,
   540647144, 541647042, 542646944, 543646840, 544646746, 545646636,
   546646542, 547646442, 548646346, 549646248, 550646146, 551646052,
   552645952, 553645860, 554645760, 555645660, 556645560,         4,
           4,         4,         4,         4,         4,         4,
           4,         4,         4,         4,         4,         4,
           4,         4,         4,         4,         4,         4,
           4,         4,         4,         4,         4,         4,
           4,         4,         4,         4,         4,         4,
           4,       288,       683,        32,       440,       838,
         173,       594,       940,       352,       760,       109,
         504,       863,       262,       581,      1005,       364,
         773,       147,       517,       940,       301,       735,
          96,       478,       863])
```

we need to modify dgtz_header class in cygno library

# modified `dgtz_header` class:

```
class dgtz_header:      # very simple class for the dgtz header
    def __init__(self, a):


        self.ntriggers           = a[0]
        self.nchannels           = a[1]
        self.nsamples            = a[2]
        self.vertical_resulution = a[3]
        self.sampling_rate       = a[4]
        self.offsets             = a[5]
        self.TTT                 = a[6]
        self.SIC                 = a[7]
        if len(a)>8:
            self.nBoards          = a[8]
            self.boardNames       = a[9]
        else:
            self.nBoards         = 1
            self.boardNames      = [1742]
            print('WARNING: You are using an older version of the data

        if (len(a) > 10):
            self.TTT1 = a[10]
            #print('TTT1 set OK')
        else:
            self.TTT1 = None
            #print('TTT1 set to NONE')
```

new data member: `self.TTT1` get `Group[1]`

# modified `cy.daq_dgz_full2header(bank)` :

```
def daq_dgz_full2header(bank, verbose=False):
    # v0.1 full PMT recostruction
    import numpy as np
    nboard              = bank.data[0]
    full_buffer_size    = len(bank.data)
    name_board          = np.empty([nboard], dtype=int)
    number_samples      = np.empty([nboard], dtype=int)
    number_channels     = np.empty([nboard], dtype=int)
    number_events       = np.empty([nboard], dtype=int)
    vertical_resulution = np.empty([nboard], dtype=int)
    sampling_rate       = np.empty([nboard], dtype=int)
    channels_offset     = []
    channels_ttt        = []
    channels_ttt1       = []
    channels_SIC        = []

    # from buffer length
    # bank.data[4] events
    # bank.data[3] channels

    limit_length = ( bank.data[4] * 2 + bank.data[3] + 7 ) * nboard
    if (full_buffer_size > limit_length + bank.data[4] - 1):
        EGTTT = True
        if verbose:
            print('EGTTT is True')
    else:
        EGTTT = False
        if verbose:
            print('EGTTT is False')
```

from buffer size decides if EGTTT is enabled

modified `cy.daq_dgz_full2header(bank)` :

then read sequence

```
#### VitoMonno: reading TTT1 (ttt MSB/group[1]) for EGTT
if EGTTT:
    channels_ttt1_tmp = np.empty(number_events[iboard], dtype=int)
    for ttt in range(number_events[iboard]):
        ich+=1
        channels_ttt1_tmp[ttt] = bank.data[ich]
    if verbose:
        print ("channels_ttt: ", channels_ttt1)
    channels_ttt1.append(channels_ttt1_tmp)
```

and call the constructor

```
if EGTTT:
    full_header = dgtz_header([number_events, number_channels, number_samples, vertical_resulution,
                               sampling_rate, channels_offset, channels_ttt, channels_SIC, nboard, name_board, channels_ttt1])
else:
    full_header = dgtz_header([number_events, number_channels, number_samples, vertical_resulution,
                               sampling_rate, channels_offset, channels_ttt, channels_SIC, nboard, name_board])

return full_header
```

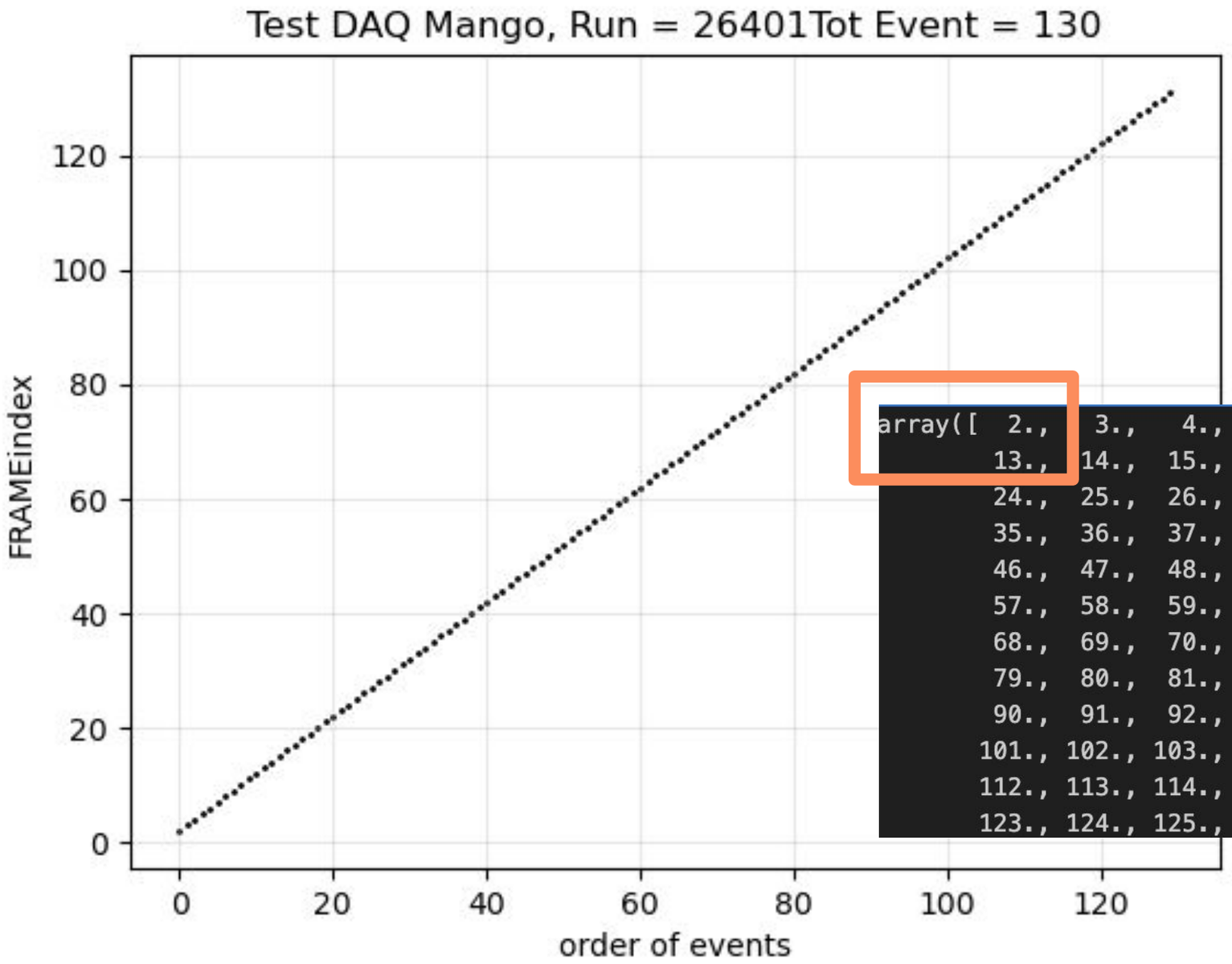
# Result (retrocompatible):

```
DgtzHeader.__dict__
✓ 0.0s
{'ntriggers': array([26]),
 'nchannels': array([32]),
 'nsamples': array([1024]),
 'vertical_resulution': array([4096]),
 'sampling_rate': array([1333]),
 'offsets': [array([13107, 13107, 13107, 13107, 13107, 16384, 26214, 32768, 32768,
        32768, 32768, 32768, 32768, 32768, 32768, 32768, 32768, 32768,
        32768, 32768, 32768, 32768, 32768, 32768, 32768, 32768, 32768,
        32768, 32768, 32768, 32768, 32768])],
 'TTT': [array([531648030, 532647932, 533647830, 534647734, 535647636, 536647532,
        537647438, 538647332, 539647240, 540647144, 541647042, 542646944,
        543646840, 544646746, 545646636, 546646542, 547646442, 548646346,
        549646248, 550646146, 551646052, 552645952, 553645860, 554645760,
        555645660, 556645560])],
 'TTT1': [array([4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4,
        4, 4, 4, 4])],
 'SIC': [array([ 288,  683,   32,  440,  838,  173,  594,  940,  352,  760,  109,
         504,  863,  262,  581, 1005,  364,  773,  147,  517,  940,  301,
         735,   96,  478,  863])],
 'nBoards': 1,
 'boardNames': array([1742]),
 'itemDict': {'0': array([26]),
  '1': array([32]),
  '2': array([1024]),
  '3': array([4096]),
...
  '7': [array([ 288,  683,   32,  440,  838,  173,  594,  940,  352,  760,  109,
          504,  863,  262,  581, 1005,  364,  773,  147,  517,  940,  301,
          735,   96,  478,  863])],
```

now seems fine :)

# Other improvements

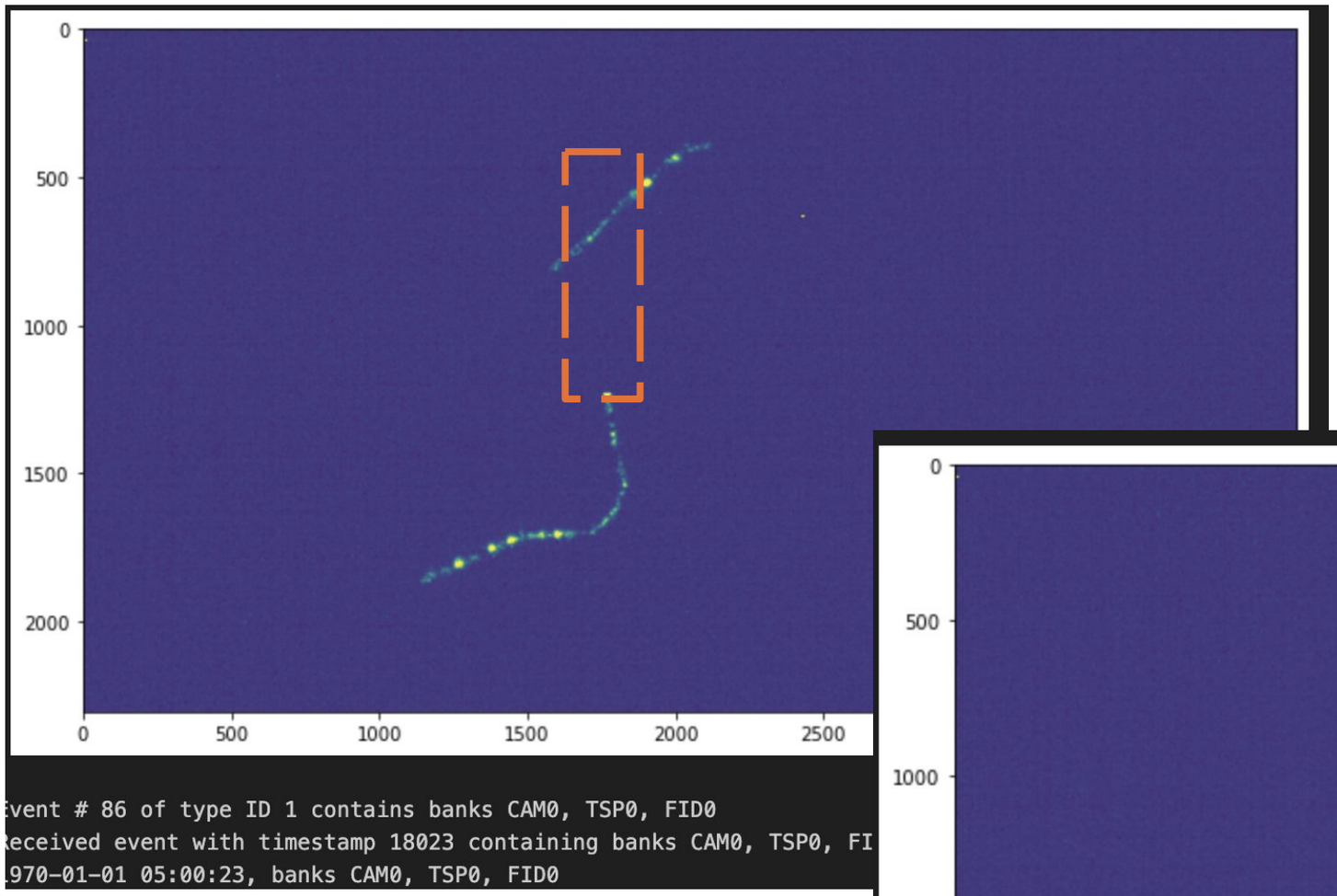
# **NEW**: added FID0 bank to have frame index




```
array([  2.,   3.,   4.,   5.,   6.,   7.,   8.,   9.,  10.,  11.,  12.,
        13.,  14.,  15.,  16.,  17.,  18.,  19.,  20.,  21.,  22.,  23.,
        24.,  25.,  26.,  27.,  28.,  29.,  30.,  31.,  32.,  33.,  34.,
        35.,  36.,  37.,  38.,  39.,  40.,  41.,  42.,  43.,  44.,  45.,
        46.,  47.,  48.,  49.,  50.,  51.,  52.,  53.,  54.,  55.,  56.,
        57.,  58.,  59.,  60.,  61.,  62.,  63.,  64.,  65.,  66.,  67.,
        68.,  69.,  70.,  71.,  72.,  73.,  74.,  75.,  76.,  77.,  78.,
        79.,  80.,  81.,  82.,  83.,  84.,  85.,  86.,  87.,  88.,  89.,
        90.,  91.,  92.,  93.,  94.,  95.,  96.,  97.,  98.,  99., 100.,
       101., 102., 103., 104., 105., 106., 107., 108., 109., 110., 111.,
       112., 113., 114., 115., 116., 117., 118., 119., 120., 121., 122.,
       123., 124., 125., 126., 127., 128., 129., 130., 131.])
```

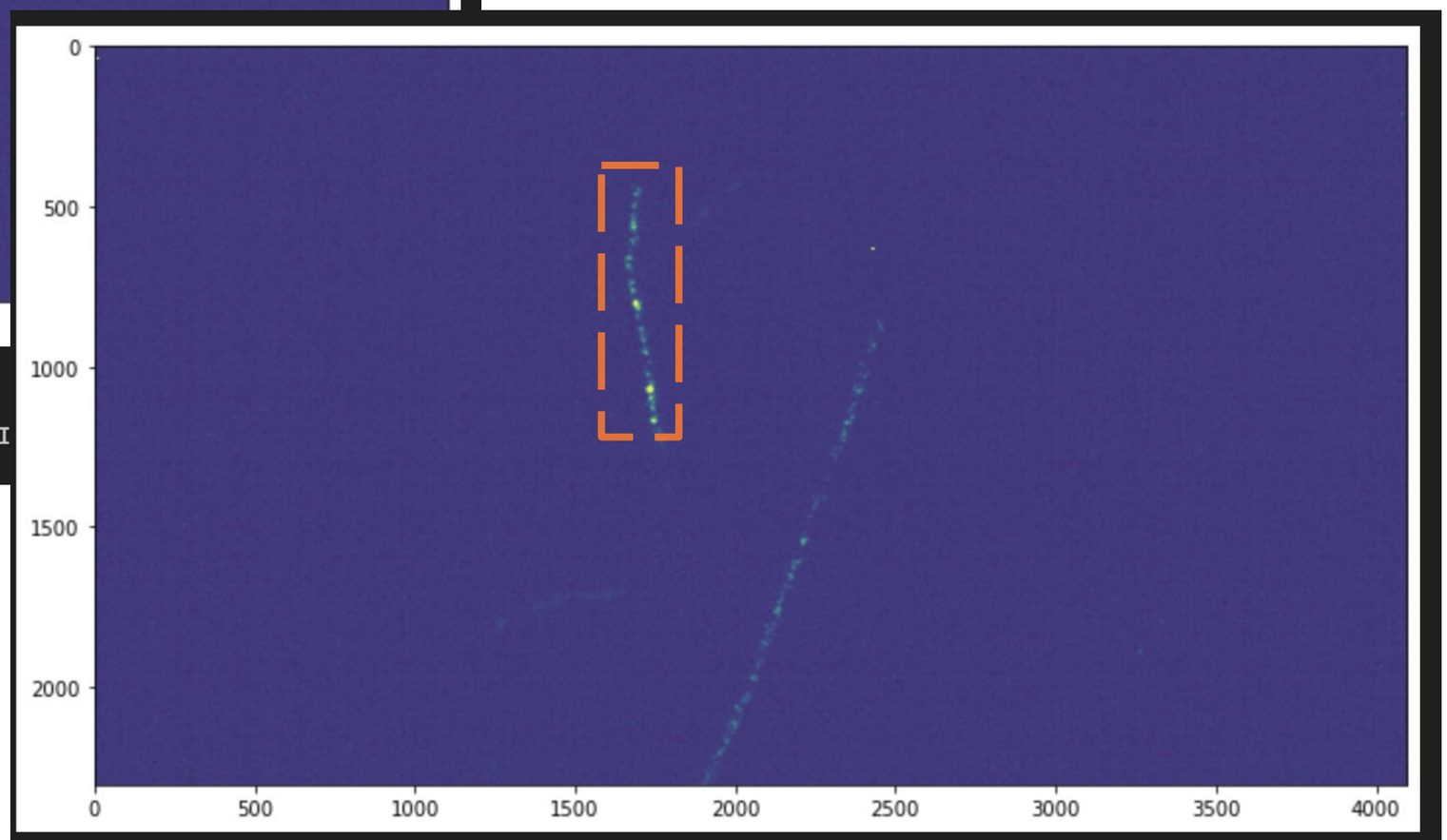
start correctly from 2, we skip willingly first 2 frames

# **NEW**: Mango ON, some examples of cut tracks



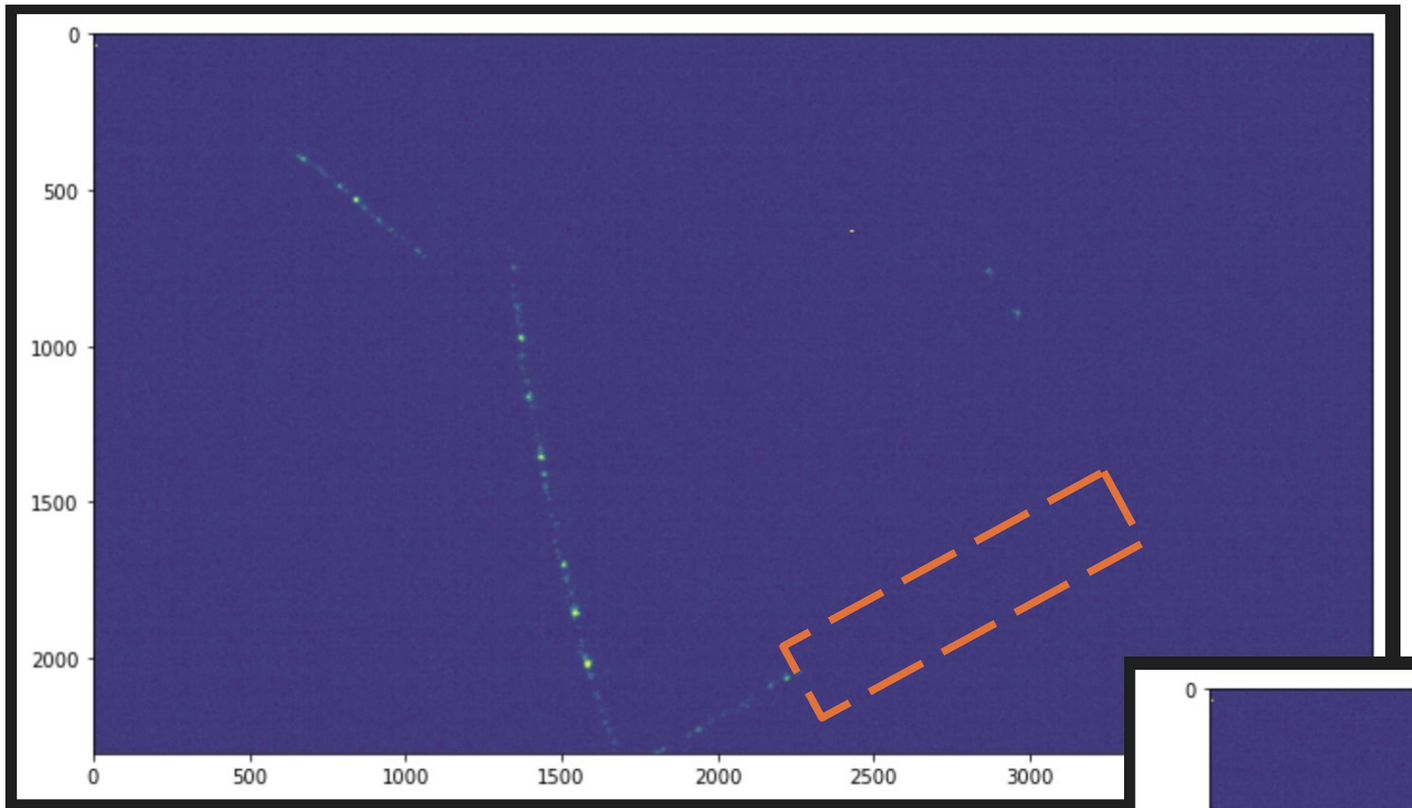

```
vent # 86 of type ID 1 contains banks CAM0, TSP0, FID0
eceived event with timestamp 18023 containing banks CAM0, TSP0, FI
970-01-01 05:00:23, banks CAM0, TSP0, FID0
```

Runs:
26411-26414

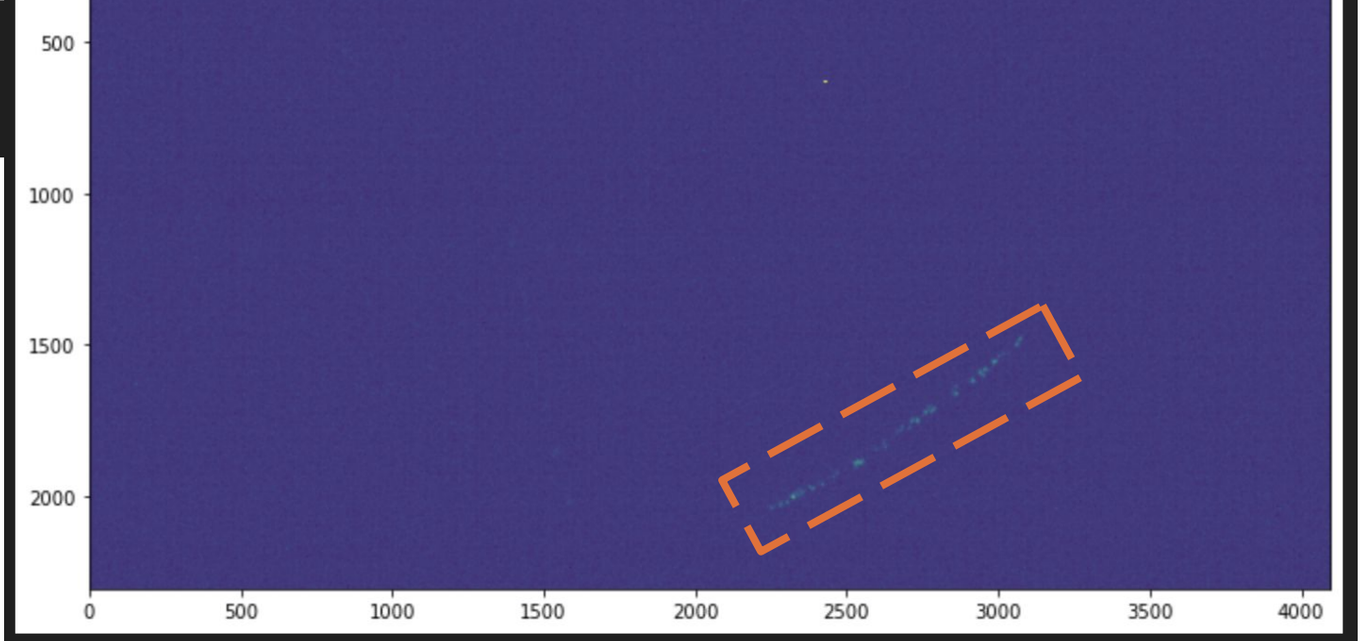



```
vent # 87 of type ID 1 contains banks CAM0, TSP0, FID0
eceived event with timestamp 18224 containing banks CAM0, TSP0, FID0
970-01-01 05:03:44, banks CAM0, TSP0, FID0
```

# some examples

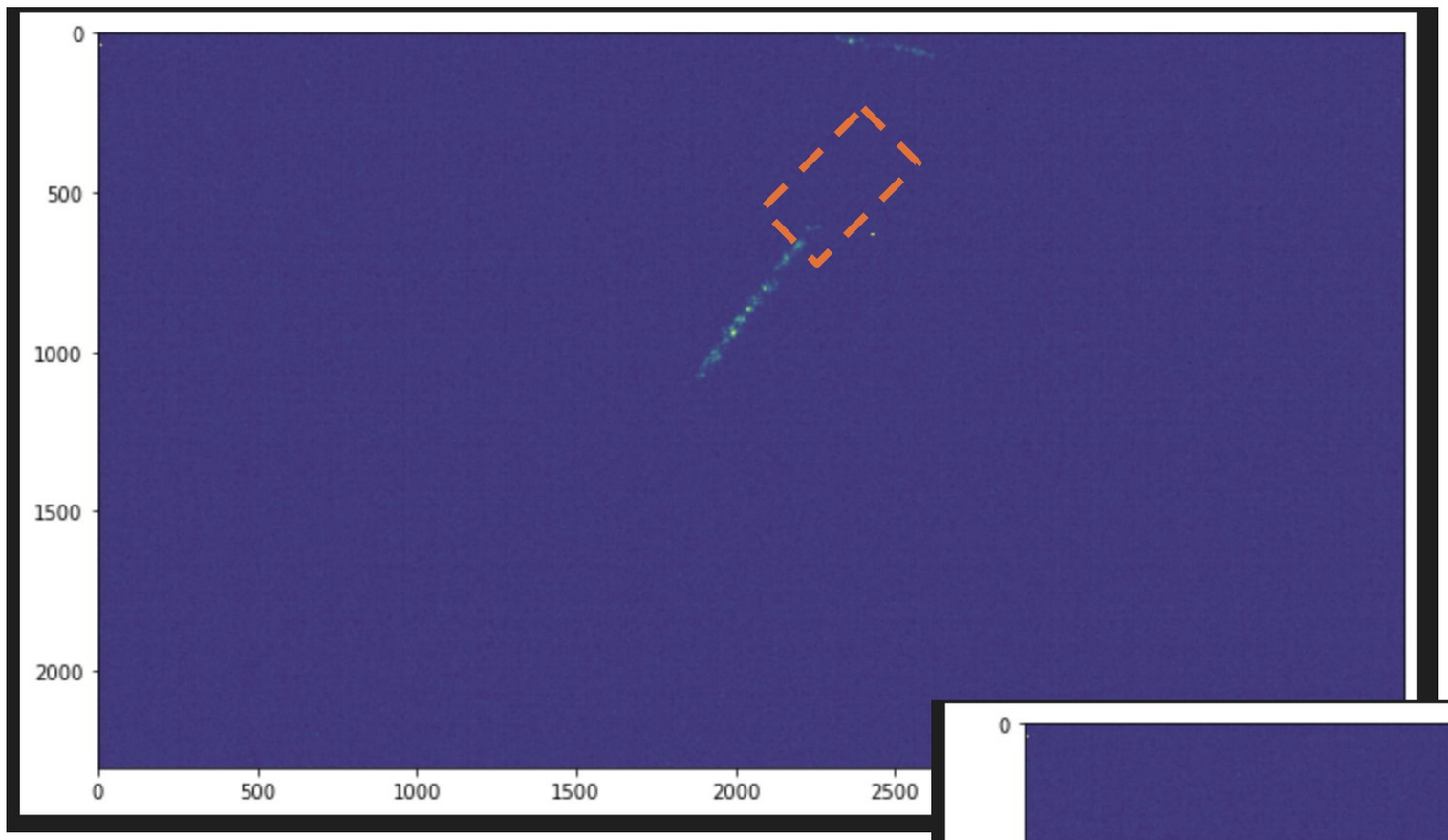


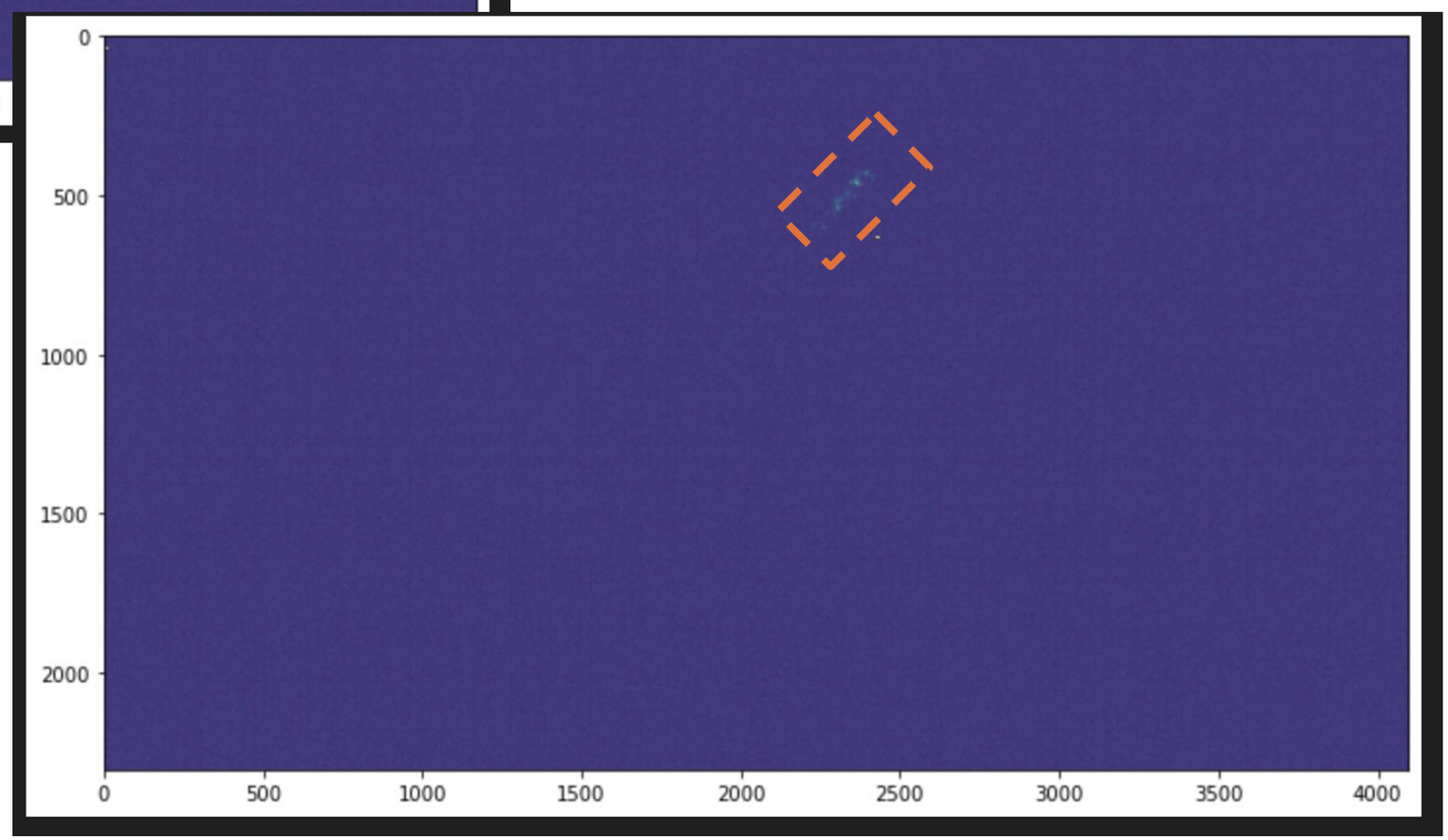

```
Event # 83 of type ID 1 contains banks CAM0, TSP0, FID0
Received event with timestamp 17421 containing banks CAM0, TSP0, FID0
1970-01-01 04:50:21, banks CAM0, TSP0, FID0
```



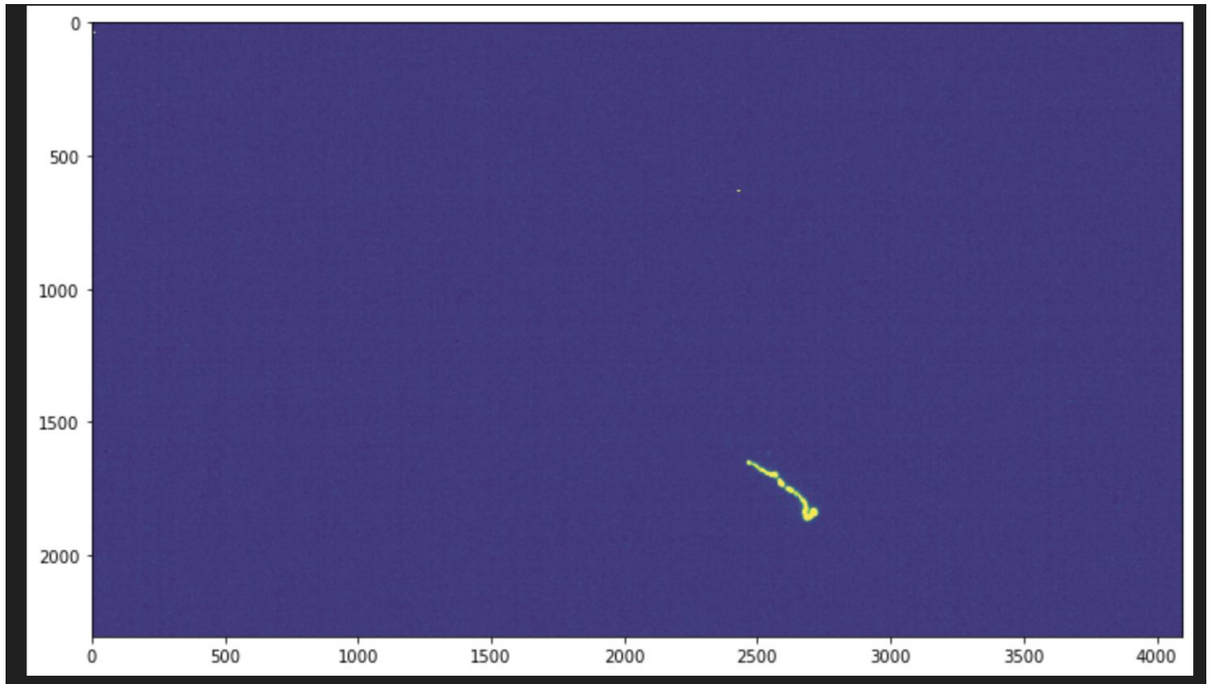

```
Event # 84 of type ID 1 contains banks CAM0, TSP0, FID0
Received event with timestamp 17621 containing banks CAM0, TSP0, FID0
1970-01-01 04:53:41, banks CAM0, TSP0, FID0
```

# some examples

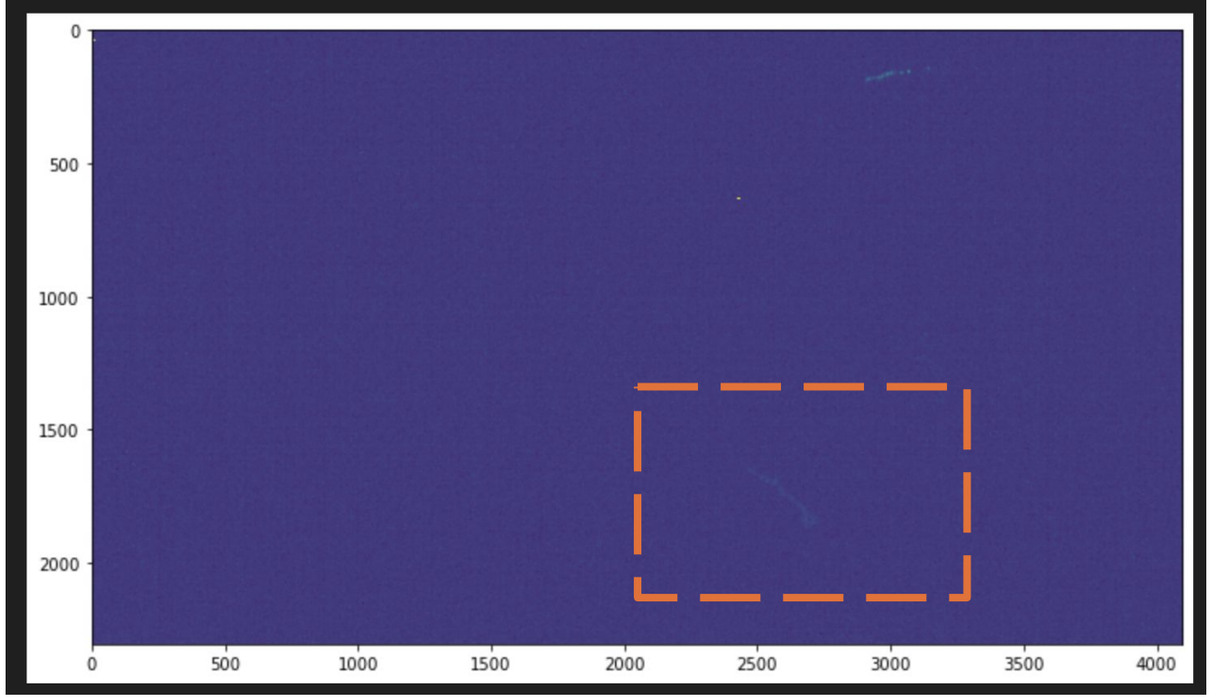


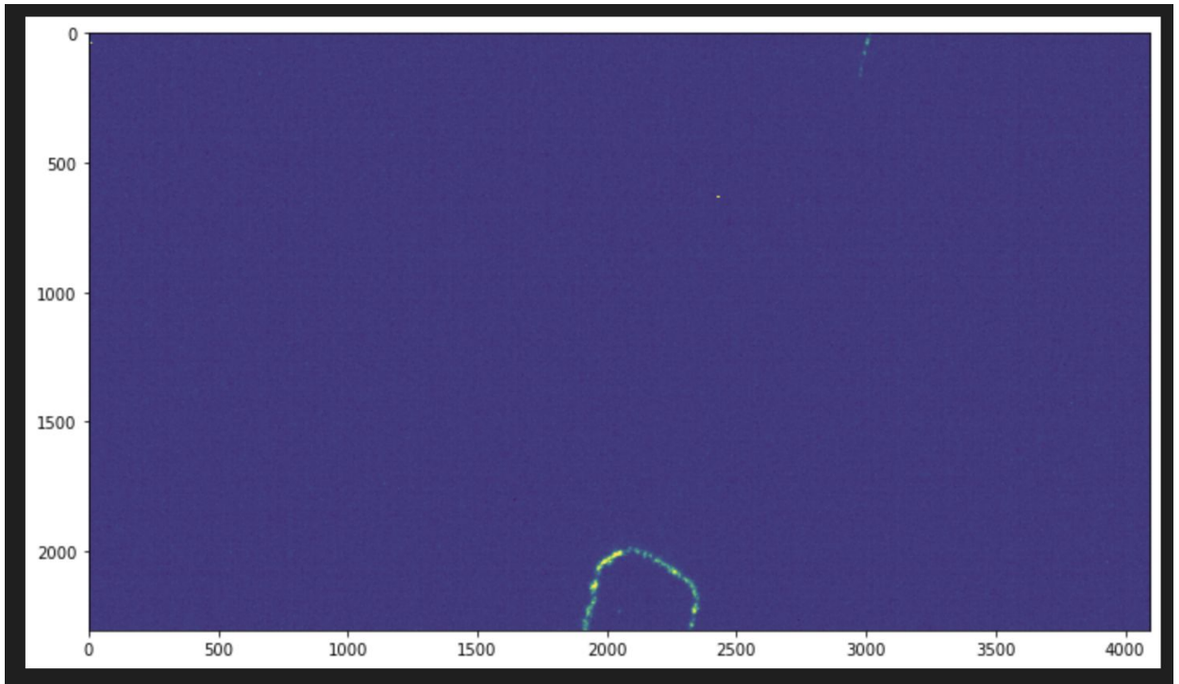



# some residual effects

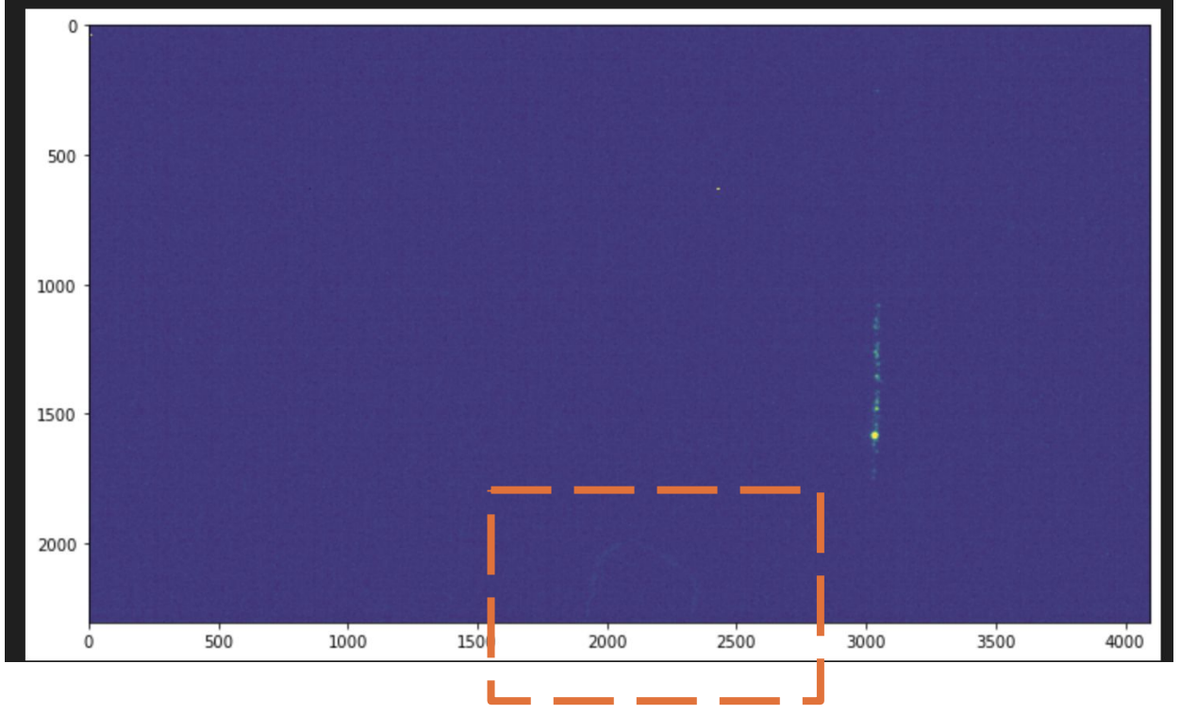



```
Event # 8 of type ID 1 contains banks CAM0, TSP0, FID0
Received event with timestamp 2413 containing banks CAM0, TSP0, FID0
1970-01-01 00:40:13, banks CAM0, TSP0, FID0
```



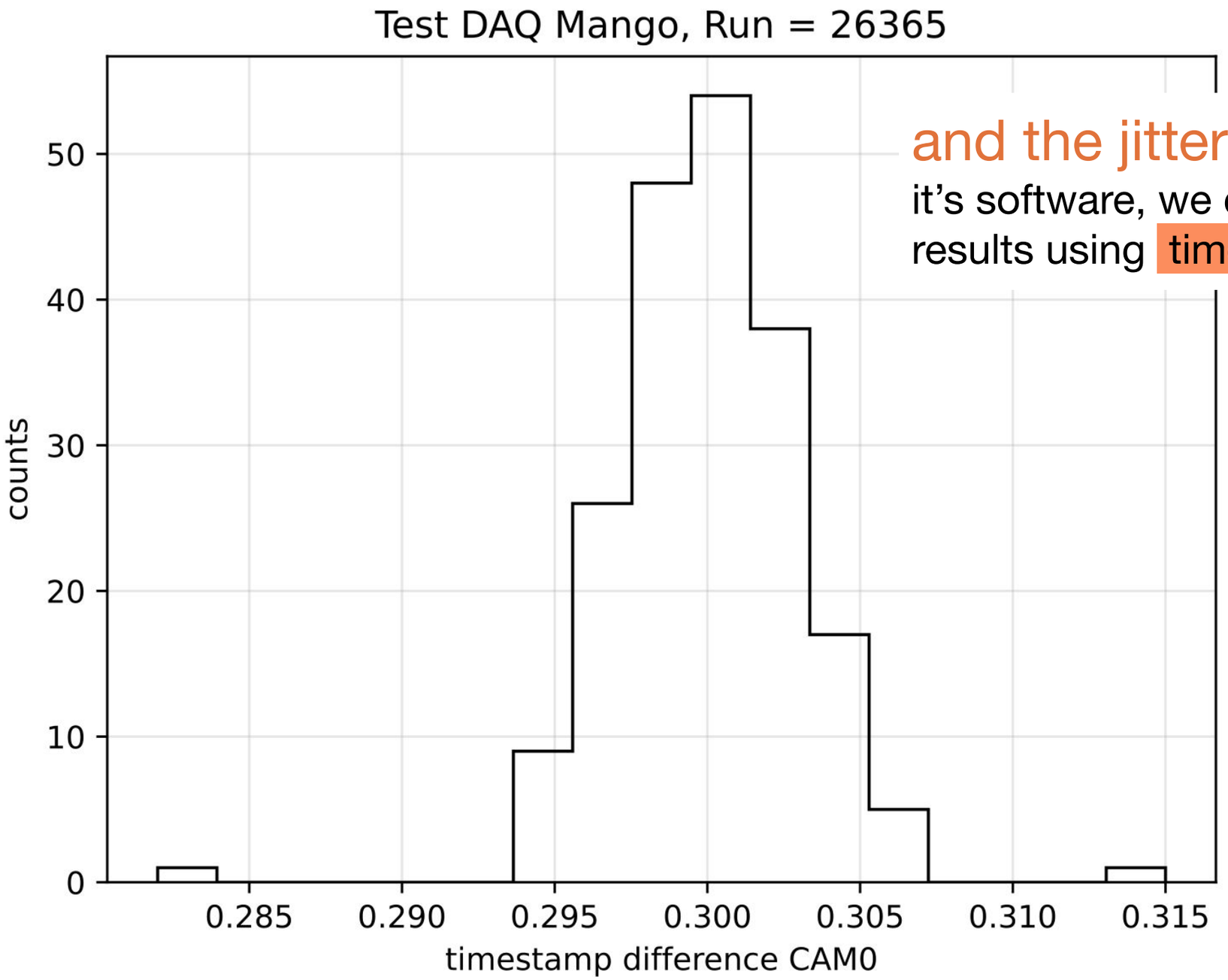




```
Event # 26 of type ID 1 contains banks CAM0, TSP0, FID0
Received event with timestamp 6017 containing banks CAM0, TSP0, FID0
1970-01-01 01:40:17, banks CAM0, TSP0, FID0
```



# **Update:** timestamp

previous version:

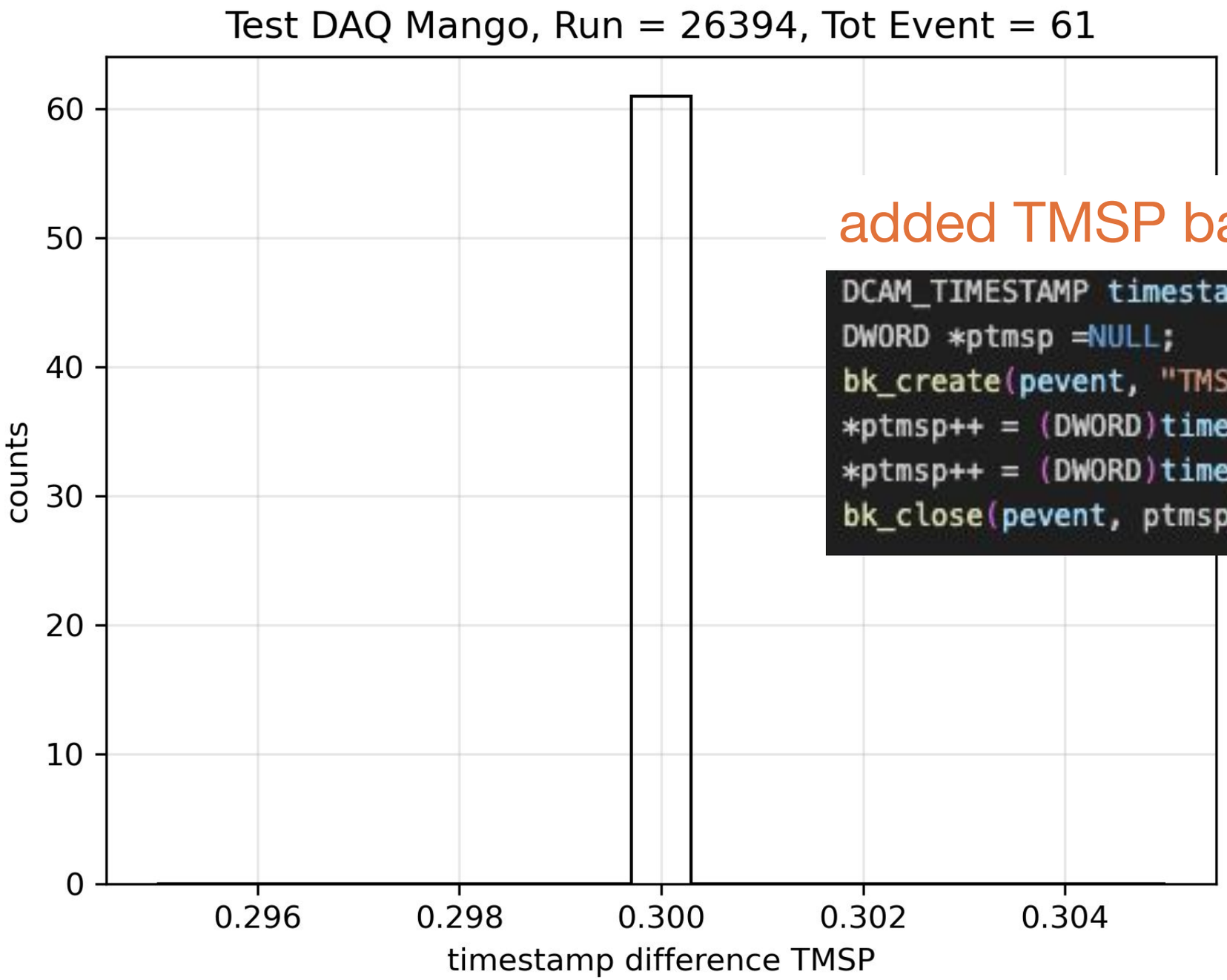



and the jitter?

it's software, we could achieve better results using timestamp from cam

# **Update:** timestamp

now: from camera




added TMSP bank:

```
DCAM_TIMESTAMP timestamp = bufframe.timestamp;
DWORD *ptmsp =NULL;
bk_create(pevent, "TMSP", TID_DWORD, &ptmsp);
*ptmsp++ = (DWORD)timestamp.sec;
*ptmsp++ = (DWORD)timestamp.microsec;
bk_close(pevent, ptmsp);
```

changed name in TSP0 now

# **Update:** timestamp

now: from camera

```
TIMESTAMP_TOT = TMSTAMP_sec+TMSTAMP_microsec*1e-6
```

✓ 0.0s

```
TMSPdiff = TIMESTAMP_TOT[1:] - TIMESTAMP_TOT[:-1]
TMSPdiff
```

✓ 0.0s Open 'TMSPdiff' in Data Wrangler

```
array([0.300154, 0.300153, 0.300154, 0.300154, 0.300153, 0.300154,
       0.300153, 0.300154, 0.300154, 0.300153, 0.300154, 0.300153,
       0.300154, 0.300154, 0.300153, 0.300154, 0.300153, 0.300154,
       0.300154, 0.300153, 0.300154, 0.300153, 0.300154, 0.300154,
       0.300153, 0.300154, 0.300153, 0.300154, 0.300154, 0.300153,
       0.300154, 0.300153, 0.300154, 0.300154, 0.300153, 0.300154,
       0.300153, 0.300154, 0.300154, 0.300153, 0.300154, 0.300153,
       0.300154, 0.300154, 0.300153, 0.300154, 0.300153, 0.300154,
       0.300154, 0.300153, 0.300154, 0.300153, 0.300154, 0.300154,
       0.300153, 0.300154, 0.300153, 0.300154, 0.300154, 0.300153,
       0.300154])
```

actually it’s just the microsec digit 3 or 4

# thanks for your attention :)

Vito Monno