

How to debug conversion problems between midas and Eva data files

The time of flight data gathered from the MPET MCP is collected by MIDAS and placed in a file with a .mid extension. In order to Eva with this data it must be converted into a file format that Eva understands. This is the purpose of the MidasToEva.py script which is located on titan01 under the rr/midastoeva directory. Typically it is invoked using the m2e.sh shell script located in the same directory and aliased somewhere in the path. The invocation script is self explanatory and should be consulted if there are questions.

MidasToEva converts the .mid file into an ascii format that it can understand before rewriting the data in Eva format. It does this by executing the script

```
mdump -b MPET -x <filename>
```

You can try this yourself from the command line and pipe the output into a file. The data that we're interested in is located in the MPET bank. In each bank you'll see strings of hex numbers. Each event has two sequential hex numbers associated with it.

```
Event X: 0x83ab0000 0x00000002
```

The above sample event illustrates how the data is arranged. The first digit of the first hex number, in this case an 8, indicates the beginning of the TDC gate. The first digit specifies the event type. According to what I've been told there are only 4 valid event types defined

- 1 - end of TDC gate
- 2 - an in-gate ion event
- 4 - an out of gate ion event
- 8 - start of TDC gate

In the first hex number the three digits following the event type is a cycle counter. Often it happens that this counter rolls over as it can only count $16^3 = 4096$ cycles. The remaining four digits of the first hex number are unused. The second hex number is the time, so in the case of an ion event, it is the time of flight of the ion to the detector.

In the mdump data, MidasToEva expects to see an end of gate event at some point after the corresponding start of gate event such as:

```
0x8abc0000 0x00000002 .... 0x1abc0000 0x0004e21
```

For each start of gate a corresponding end of gate (with the same cycle number) needs to be present or the cycle isn't counted properly and it will result in a missed scan in the resulting Eva file and typically shift events with respect to the applied frequency (a bad thing).

A problem that popped up randomly was that every once in a while when converting a midas file, we would observe that the eva file contained one fewer scan than expected. Upon inspecting the mdump data, it was found that some start of gate events were labeled with an “a” instead of 8. This was corrected in MidasToEva5.py by relabeling the event type in the reorganizeMdmpData method (see below)

```
for i in range(0,arraylen,2):
    firsthex=self.mdmpdata[i]
    sechex=self.mdmpdata[i+1]
    evttype=firsthex[2]
    if(evttype=='a'): <---relabel 'a' events
        evttype=8
    cyclenum=int(firsthex[3:6],16)
    tof=float(int(sechex[2:],16))*1e-8
    self.temparray.append(int(evttype))
    self.temparray.append(cyclenum)
    self.temparray.append(tof)
    temp=(int(evttype),cyclenum,tof)
    self.mdmparray.append(temp)
```

That fixed the problem for quite some time. Now another problem has popped up where end of gate events are labeled with a 6. This was fixed in a similar fashion in MidasToEva6. This appears to be a random problem with stuck bits in the hardware. It probably makes sense to avoid this relabeling of event types just to see when these problems arise, and to make sure they're being handled correctly by inspecting the mdump data to verify what the problem is.

A method has been added in MidasToEva6 called writeMdmpData which writes each individual event which can then be parsed with other software (I use Mathematica) to locate the problem. The file is written to the same directory as the Eva file (triumfcs\trshare\titan\MPET\Data) with the extension .dump. It can be enabled by uncommenting the line in the m2e.sh file, but I suggest using the m2e_test.sh to avoid making changes that will affect other users.