

## EXPERT ROOT - Developing #64

### BeamDet диджитализация

07/11/2017 07:33 AM - Vitaliy Schetinin

<b>Status:</b>	Закрота	<b>Start date:</b>	07/11/2017
<b>Priority:</b>	Низкий	<b>Due date:</b>	
<b>Assignee:</b>	Mikhail Kozlov	<b>% Done:</b>	0%
<b>Category:</b>	BeamDet	<b>Estimated time:</b>	0.00 hour
<b>Target version:</b>	v-0.4		

#### Description

Необходимо разработать класс ERRBeamDetDigitizer диджитализации для детектора BeamDet.

Моделирование отклика или диджитализация учитывает гранулярность электроники, неэффективности, шумы, тёмные счёты и продуцирует такую структуру данных сигнала, как если бы он был получен от реального детектора.

В первом приближении все перечисленные выше эффекты учитываются параметризованными распределениями ошибок.

Класс должен быть наследован от FairTask. Можно посмотреть структуру в ERRTelescopeHitFinder. Фактически task перегоняет данные из одного представления (к примеру ERBeamDetTofPoint и ERBeamDetMWPC) в другое представление (назовем эти класс ERBeamDetTofDigi и ERBeamDetMwpcDigi). В каждом событии у нас есть коллекция поинтов на входе и должна быть коллекция Digi на выходе. Digi это в перспективе структура данных, которая будет приходить с реального детектора. В реальном детекторе мы не можем определить в какое именно место twpc проволоочки попала частица мы можем сказать ее номер, также не можем определить простым способом сколько частиц одновременно родилось/навелось на проволочку и сколько каждая из них оставила на ней энергии, мы можем сказать только суммарную энергию (которая потом будет в отсчетах электроники, а пока в энергетических единицах), также мы не можем говорить в рамках сенсора какая именно частица в какой момент в событии в него попала, мы зафиксируем время от первой, как время появления сигнала.

В Tof digi соответствует пластинки с которой снимается сигнал. В MWPC номеру проволоочки(Тут нужно тащить весь адрес: номер сдвоенной станции, первая или вторая в сдвоенной, номер проволоочки).

Классы данных нужно наследовать от FairMultiLinkedData(позволяет отслеживать на базе каких других объектов образован) и расположить в папке data. Он должен содержать (полный адрес в геометрии - (все номера объемов)), суммарную энергию поинтов в секторе, время (как время самого первого поинта в сенсоре)). С помощью метода digi->AddLink(FairLink("BeamDetTofPoint",iPoint)); должен быть слинковам со всеми поинтами, которые участвовали в его создании.

Энергию и время в обоих случаях нужно разыграть по гауссу с сигмой, переданной по интерфейсу из макроса (SetTofESigma(sigma), SetTofTSigma(sigma)).

Также нужно ввести порог на энергию для записи digi в файл (SetTofEThreshold(threshold))

Для теста: Tof: sigmaT = 100пс, sigmaE - уточнить у СГ, thresholdE - определить по спектру - 90% должно быть больше порога;MWPC:Надо посмотреть на спектр энерговыделений и на множественность vs порог. Множественность - количество загоревшихся проволочек на станции.

#### History

##### #1 - 07/11/2017 12:49 PM - Mikhail Kozlov

Как я понял, нужно еще создать класс ERBeamDetSetup.

Какие у него должны быть атрибуты?

## #2 - 07/11/2017 03:16 PM - Vitaliy Schetinin

He. Он нужен для реконструкции дальнейшей.

А макрос то добавь, которым ты все это тестировал. Можешь посмотреть на макрос NeuRad\_digi в macro/NeuRad

## #3 - 07/11/2017 03:28 PM - Vitaliy Schetinin

Да, в диджитизации Setup класс нужен, когда мы используем файлы параметров. Здесь пока их нет. А вот в макросе нойрада есть. Когда будешь копировать не забудь удалить логику которая включает загрузку дополнительных текстовых файлов параметров:

```
TString NeuRadDetDigiFile = gSystem->Getenv("VMCWORKDIR");  
NeuRadDetDigiFile += "/parameters/NeuRad.digi.v4.par";  
parInput1->open(NeuRadDetDigiFile.Data(),"in");
```

## #4 - 07/11/2017 04:30 PM - Mikhail Kozlov

Хорошо, это учту когда доберусь до макроса.

Я не понял, как правильно использовать `digi->AddLink(FairLink("BeamDetTofPoint",iPoint))`. Должно же быть несколько поинтов, из которых формируется один digi?

Внутри Ehex() диджитизатора проходим по коллекции поинтов как в ERRTelescopeHitFinder, сравниваем энергию с пороговой:

```
Float_t eloss = gRandom->Gaus(point->GetEnergyLoss(), fElossDispersion);  
if (eloss < fElossThreshold) continue;
```

и получается, что просто не записываем в digi все, что меньше порога. В каком месте делать ликовку поинтов?

## #5 - 07/12/2017 07:41 AM - Vitaliy Schetinin

0) Ну да в ERRTelescopeHitFinder пока ошибка, но там не страшно это из физики. Хотя я поправлю в ближайшее время. И он должен называться дигитайзер тоже.

1) У нас к одному считывающему объему подключен один проводок. На нем появится суммарный сигнал от всех энерговыделений в этом чувствительном объеме. Поэтому да, нужно отсортировать поинты по проволочкам (для MWPC), и их энергию просуммировать. А время взять как время самого первого.

2) Как видно из пункта 1, на один digi может приходиться несколько поинтов. Информация об этом нам нужна при анализе. К примеру если мы захотим узнать какие треки сформировали этот digi. Для этого есть механизм линков. Первый аргумент AddLink это название ветки в root файлах которое задается в Register. В этих ветках мы же храним TClonesArray, то есть массив объектов. Вот второй аргумент это номер объекта в этом TClonesArray. Тот по которому ты с помощью At(index) достаешь оттуда его. Это механизм кстати позволяет вешать на линки веса и сворачивать многоуровневые линки. К примеру дальше мы будем выстраивать такую цепь points

## #6 - 07/12/2017 11:45 AM - Vitaliy Schetinin

Закончил примерно тоже самое для QTelescope. Можно посмотреть:  
<https://github.com/ExpertRootGroup/er/blob/dev/telescope/ERQTelescopeDigitizer.cxx>

**#7 - 07/14/2017 11:22 AM - Mikhail Kozlov**

1) Как и можно ли вообще сделать так, чтобы в digi был номер события?

**#8 - 07/15/2017 11:30 AM - Vitaliy Schetinin**

Можно, но не нужно.

1) Номер события можно достать из io менеджера таким образом `ioman->GetEntryNr()`. (а он сам синглтон - `FairRootManager* ioman = FairRootManager::Instance();`)

2) В выходном файле всегда есть атрибут `fEventID` в `MCEventHeader`, если используется `FairRunSim` или `fMCEntNo` в `EventHeader`, если используется `FairRunAna`

**#9 - 07/31/2017 03:31 PM - Mikhail Kozlov**

Для анализа трекинга MWPC с помощью TreeViewer нужно чтобы в качестве entry было событие, а не диджи. Должно быть 4 ветки, в каждой из которых лежат только диджи для одной из плоскостей.

При этом вроде бы теряется смысл коллекции диджи. Можешь как-то это прокомментировать?

**#10 - 07/31/2017 03:33 PM - Vitaliy Schetinin**

В далеком и светлом будущем у нас будут методы работы с MWPC со множественностью больше 1.

**#11 - 08/09/2017 08:08 AM - Vitaliy Schetinin**

- *Status changed from Открыта to Закрыта*