

# SHUTTLE prototype for PHOS

B.Polichtchouk

IHEP, Protvino

ALICE Off-Line Week

4 Oct 2006

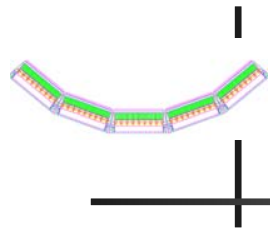


# PHOS calibration strategy

---



- At the end of each run an array of mean amplitudes in channels will be obtained and can be stored as root **file1** on the file exchange server (FES)
- At the end of each run the special procedure ("**DeadChannelsFinder**") will start finding channels with the average amplitude below the threshold. These channels are declared as dead. Dead channels map will be updated and stored as root **file2** on the FES.
- Then calibration procedures will start, using the combined information from **file1** and **file2**



# PHOS calibration procedures

---



Two procedures are foreseen:

- Adjustment of mean amplitudes in channels

*Adjustment coefficient for  $i$ -th crystal  $C[i]=\langle A[i] \rangle / A_0$ ,*

*where  $A_0$  – mean amplitude in **arbitrary chosen “reference” cell.***

- Minimizing the deviation of two-photon invariant mass from the  $\pi^0$  mass



# AliPHOSCalibHistoProducer class



```
class AliPHOSCalibHistoProducer : public TObject {  
public:
```

```
    AliPHOSCalibHistoProducer(AliRawReader* rawReader);
```

```
    void Run();
```

```
    void UpdateHistoFile();
```

```
    void SetUpdatingRate(Int_t rate) { fUpdatingRate = rate;}
```

```
protected:
```

```
    TH1F* fMeanAmp[5][56][64]; // amplitudes in [module][column][row].
```

```
    AliRawReader* fRawReader; // raw data reader.
```

```
    TFile* fHistoFile;
```

```
    Int_t fUpdatingRate;
```

# AliPHOSCalibHistoproducer: functionality



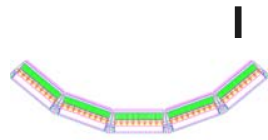
- Produces root file with calibration histograms on the online machine (GDC/LDC,HLT)
- Input raw stream is provided by **AliRawReader** class
- Updates histograms every Nth event
- If histogram file already exists in the working directory, updates it
- At the moment the only kind of histograms is "amplitudes per channel"



# AliPHOSCalibHistoProducer: how to use



```
void run(const char* file="2006run2211.root")
{
  AliRawReaderRoot* rf = new AliRawReaderRoot(file);
  AliPHOSCalibHistoProducer hp(rf);
  hp.Run();
}
```



# AliPHOSDeadChannelsFinder



**...to be implemented**



# AliPHOSPreprocessor class



- Uses root files with calibration histograms and dead channels map produced by **AliPHOSCalibHistoProducer** and **AliPHOSDeadChannelsFinder**
- Implements calibration algorithms
- At the moment only one precalibration algorithm is implemented: **adjustment of mean amplitudes in channels**

*Adjustment coefficient for  $i$ -th crystal  $C[i]=\langle A[i] \rangle / A_0$ ,*

*where  $A_0$  – mean amplitude in **arbitrary chosen “reference” cell.***

- More elaborated calibration algorithms can be added easily



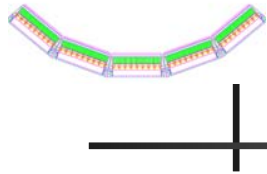


# Summary

---



- Prototype of PHOS preprocessor and procedure for running on the online machine are implemented
- More elaborate calibration algorithms can be readily added into the PHOS preprocessor



# TODO

---



- Using of dead channels map in the PHOS preprocessor
- Special online procedure for pedestals calculation
- “Pi0 mass” calibration algorithm in the PHOS preprocessor. If AliRoot is not allowed on the online machine, lightweight replacement of AliPHOSClusterizer have to be developed.