

PEBS E-cal
Work Plan and Status
PEBS meeting 27-28 January 2009

Tatsuya Nakada
LPHE/EPFL
(Swiss Federal Institute of
Technology Lausanne)

PEBS E-Cal

- Identification of e/hadron by E/p and shower development (up to few 100 GeV/c)
- Tracking of minimum ionising particles
- Space, weight and power constraints
- Keep commonality with other detectors as much as possible

Proposed solution

- W-scintillator sampling calorimeter with lateral segmentation:
 - Sampling part: scintillator bar with embedded WLS-fibre
 - Photon readout with MPPC (SiPMT)
 - Electronics very similar to the fibre-tracker (same FE chips and functionality but adjusted to different geometrical constraints)

Detector Parameters

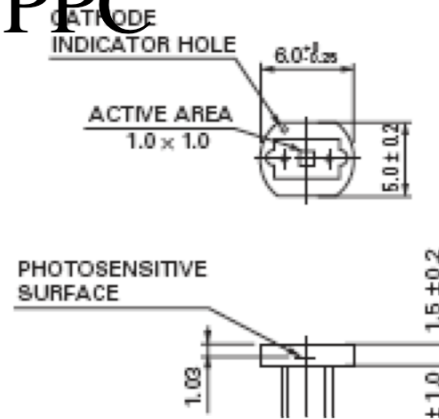
- Size of the active area
- Thickness of the W converter and number of layers
- Thickness and lateral size of the scintillator bar
- Dimension of the WLS-fibre
⇒ first optimization was done by Aachen-ITEP
2mm W and 2mm scintillator: $4 \times 5 = 20$ layers
($x \times y$ coordinate), $11.4 X_0$ ($0.2 \times 20 / 0.35$)



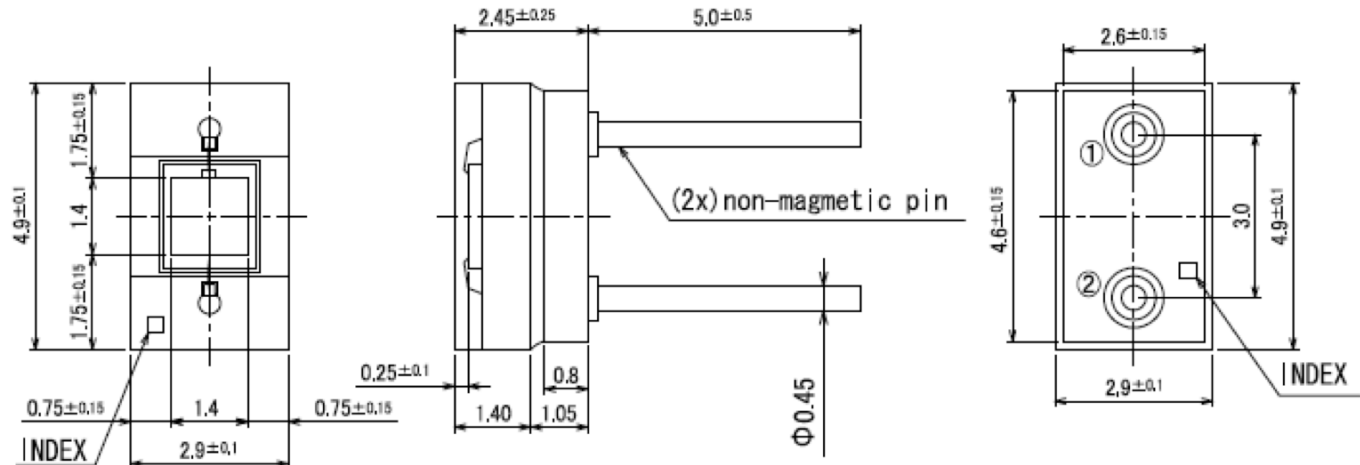


- WLS-fibre $\phi = 1\text{mm}$
- MPPC both sides, one with a filter to reduce n_γ
- Modified Hamamatsu MPPC

Commercial one



Modified one



1.4x1.4mm² with 25x25μm² pixel (56² pixels)

No-magnetic pins

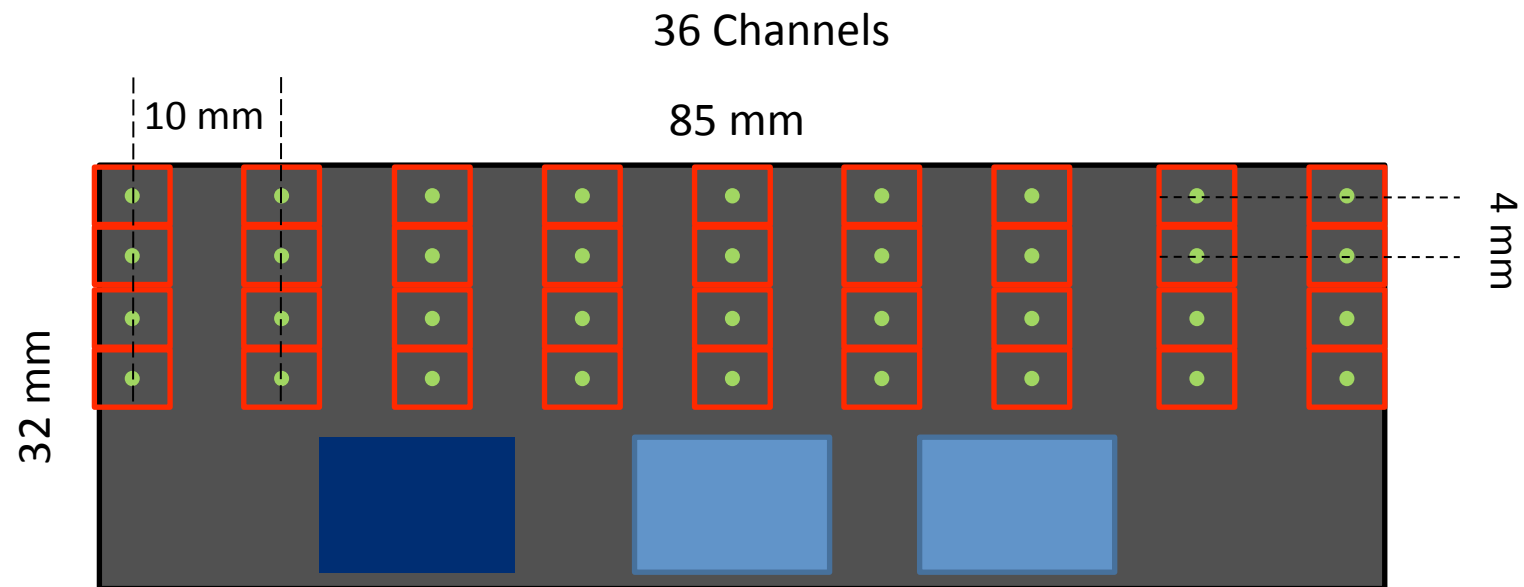
300 ordered and expected in W-12, i.e. mid March 09 → acceptance issue by JBM

general tolerance : ±0.2



Work Plan

- Construction of prototype E-cal



With SPIROC based electronics

Beam tent in Autumn 2009

- GEANT4 studies for the prototype and further optimisation of the PEBS E-cal, in view of new physics input^{*)} → LS
- Electronics and MPPC studies, e.g. gain stability etc.
- Mechanical design of PEBS E-cal

^{*)}Software framework in a modular structure would be welcome for developing subsystem software...

Group involved so far

- Aachen: mechanics, electronics
- ITEP: mechanical assembly
- EPFL: Hamamatsu MPPC, electronics, final assembly, simulation
- Tsinghua: electronics

EPBS-Ecal is not big → a better logistic of the detector construction would be more efficient?