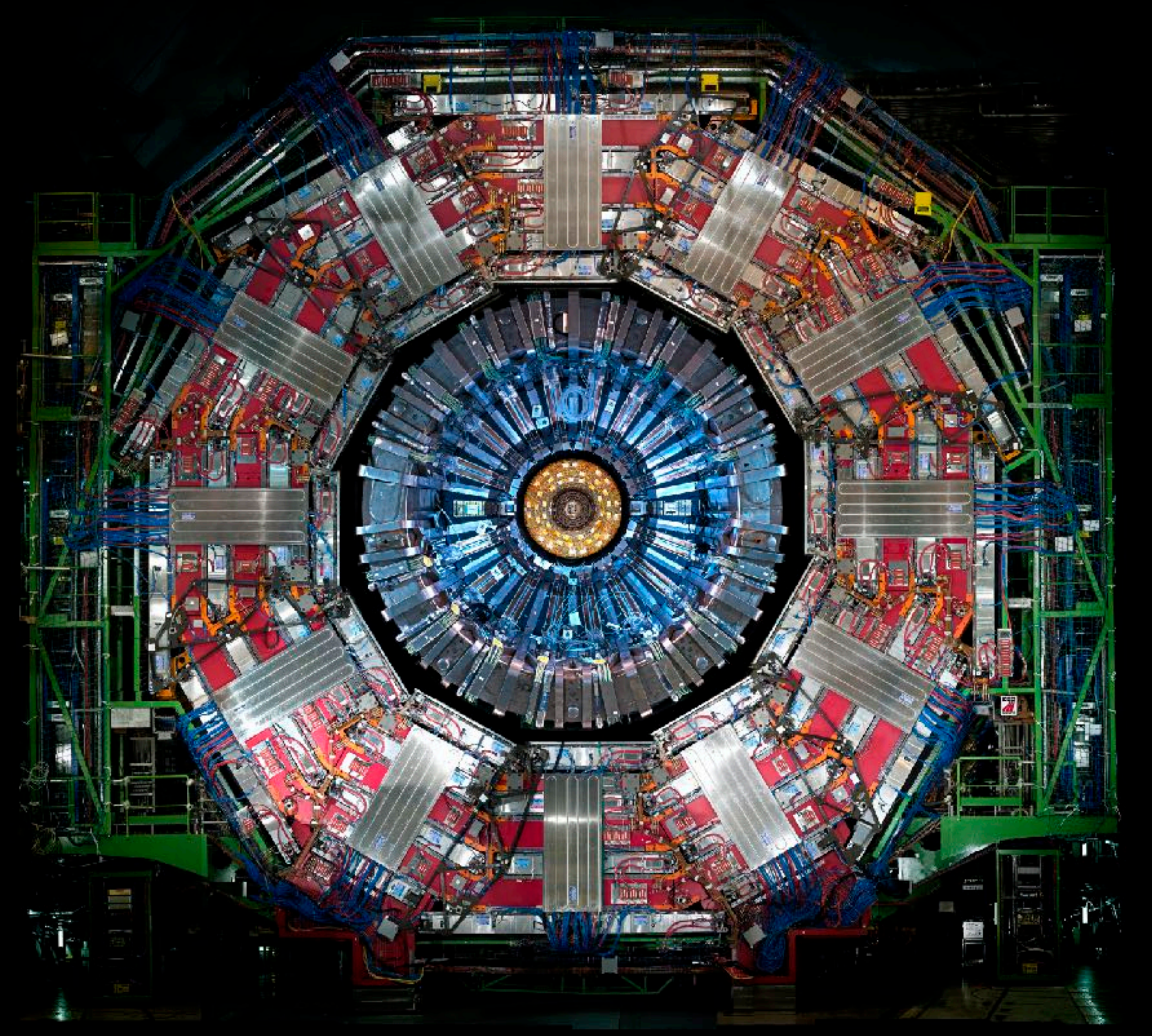
Timing Detector R&D For The CMS Experiment At KU

Andrés Abreu
University Of Kansas

KU PALOOZA 2022
Saturday, March 26 2022

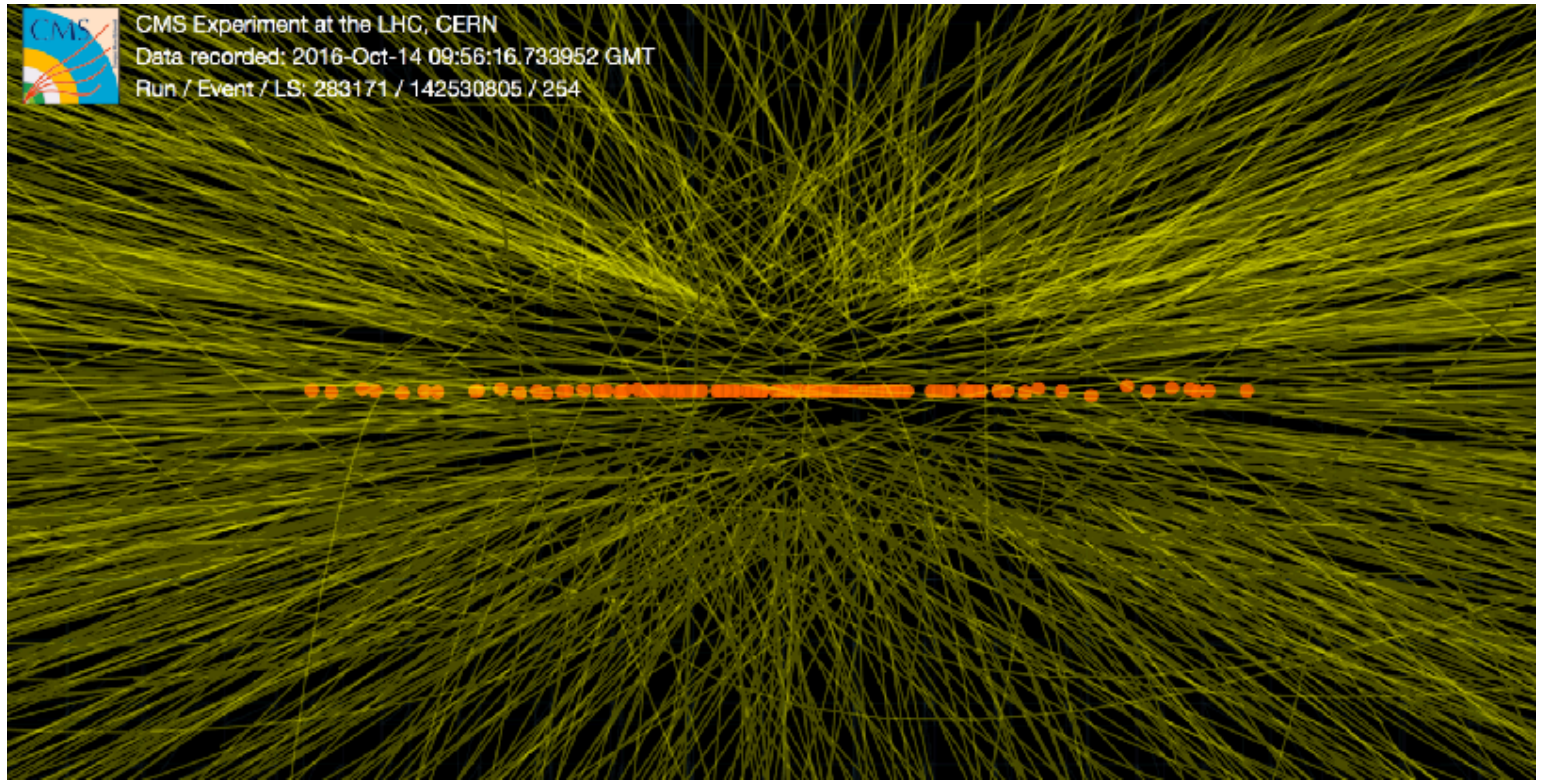
The CMS detector

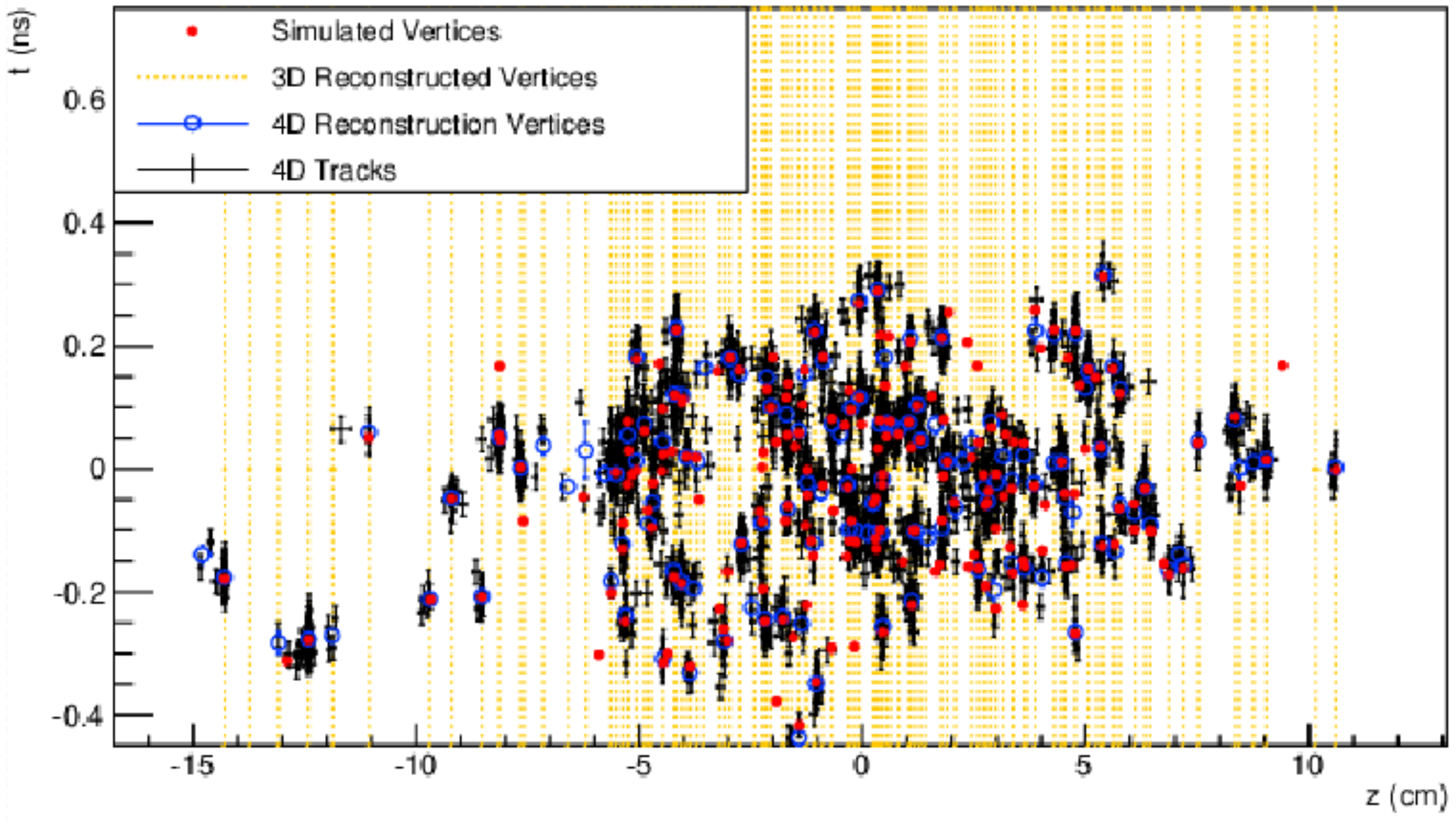


- One of the four main particle detectors at the Large Hadron Collider in Geneva, Switzerland.
- Consists of multiple subdetectors arranged in layers like an onion.
- Higgs boson discovery achieved in 2012.
- Complete overhaul coming soon with addition of new technologies:
  - New timing detector

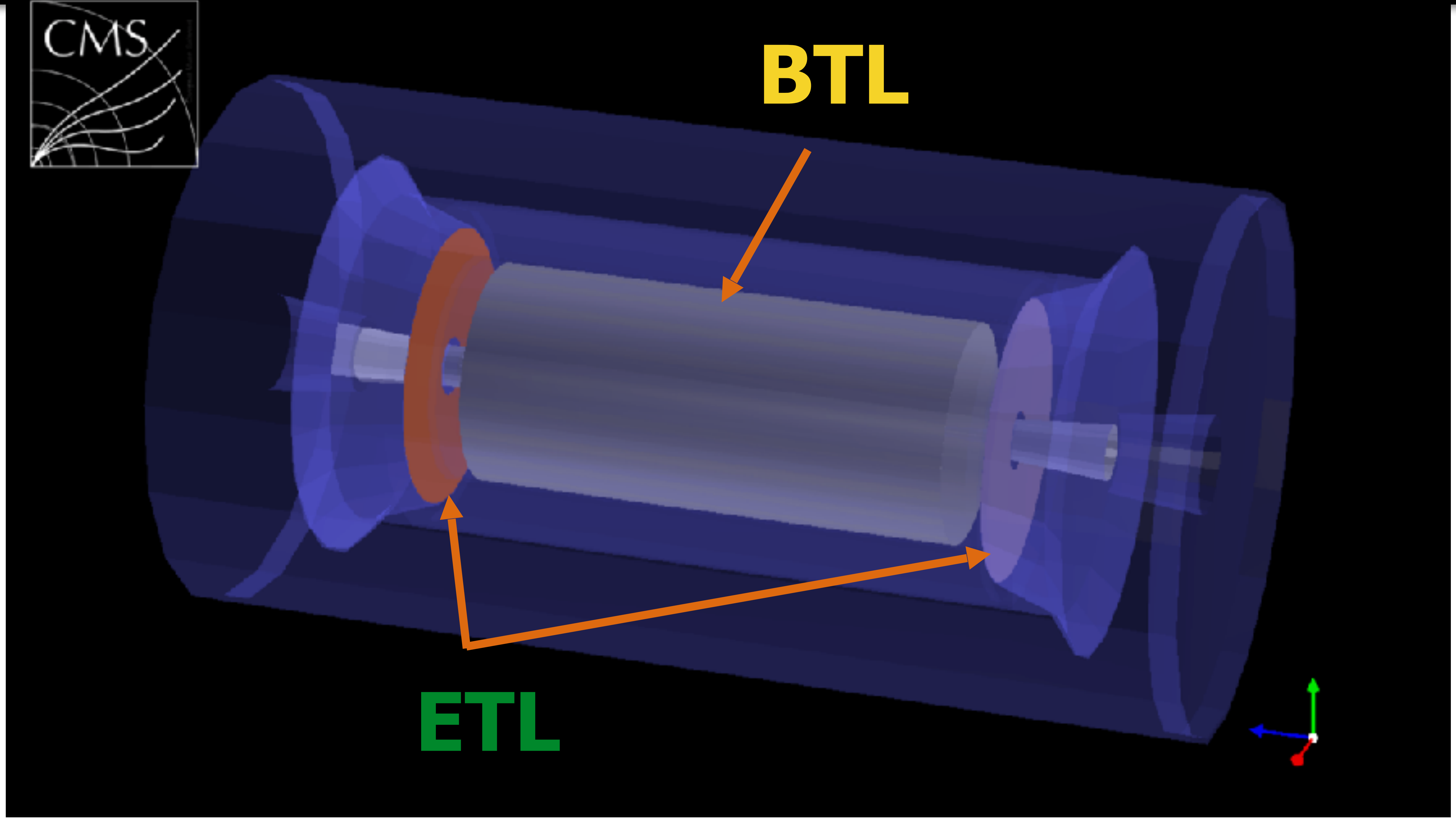
Why Timing?

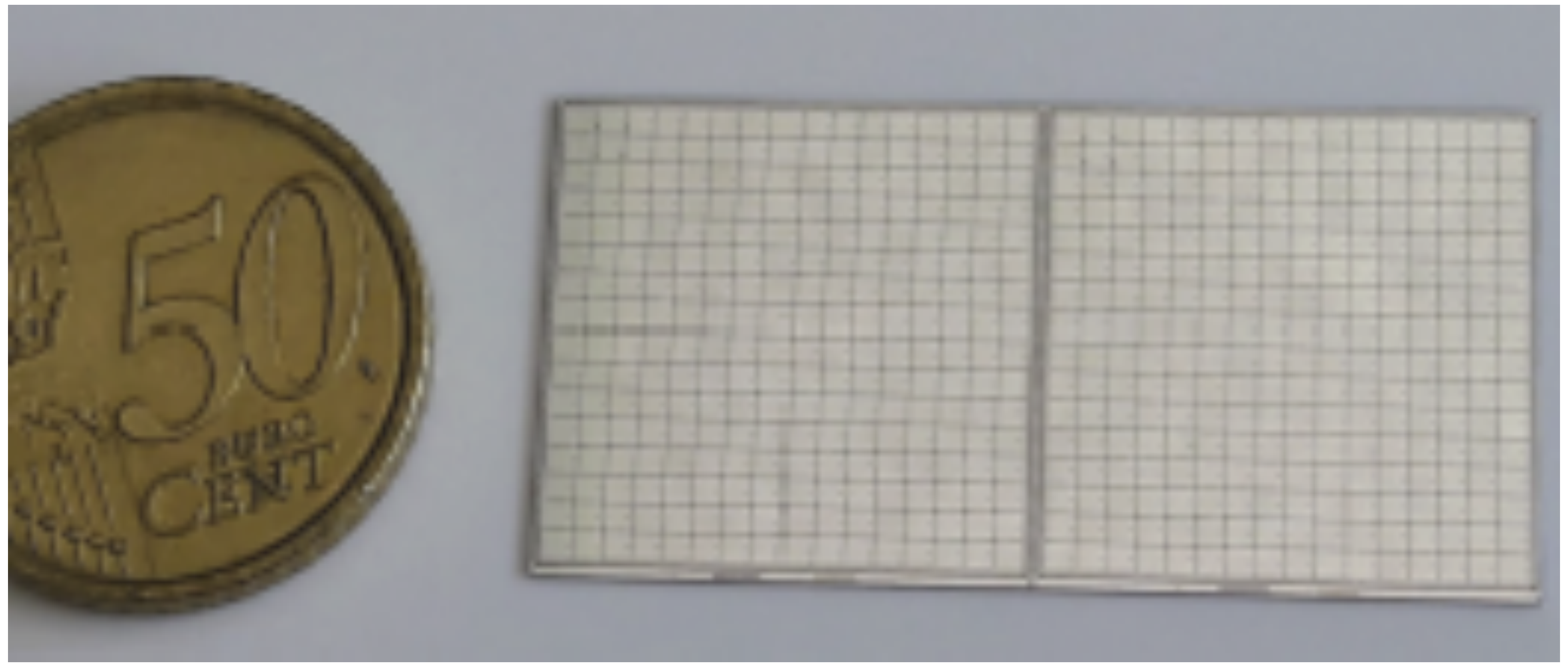
- No current dedicated component for timing at the CMS detector.
- Essential for handling the conditions of the new and improved LHC (HL-LHC).
- Increased potential for finding new Physics.





MIP Timing Detector Overview





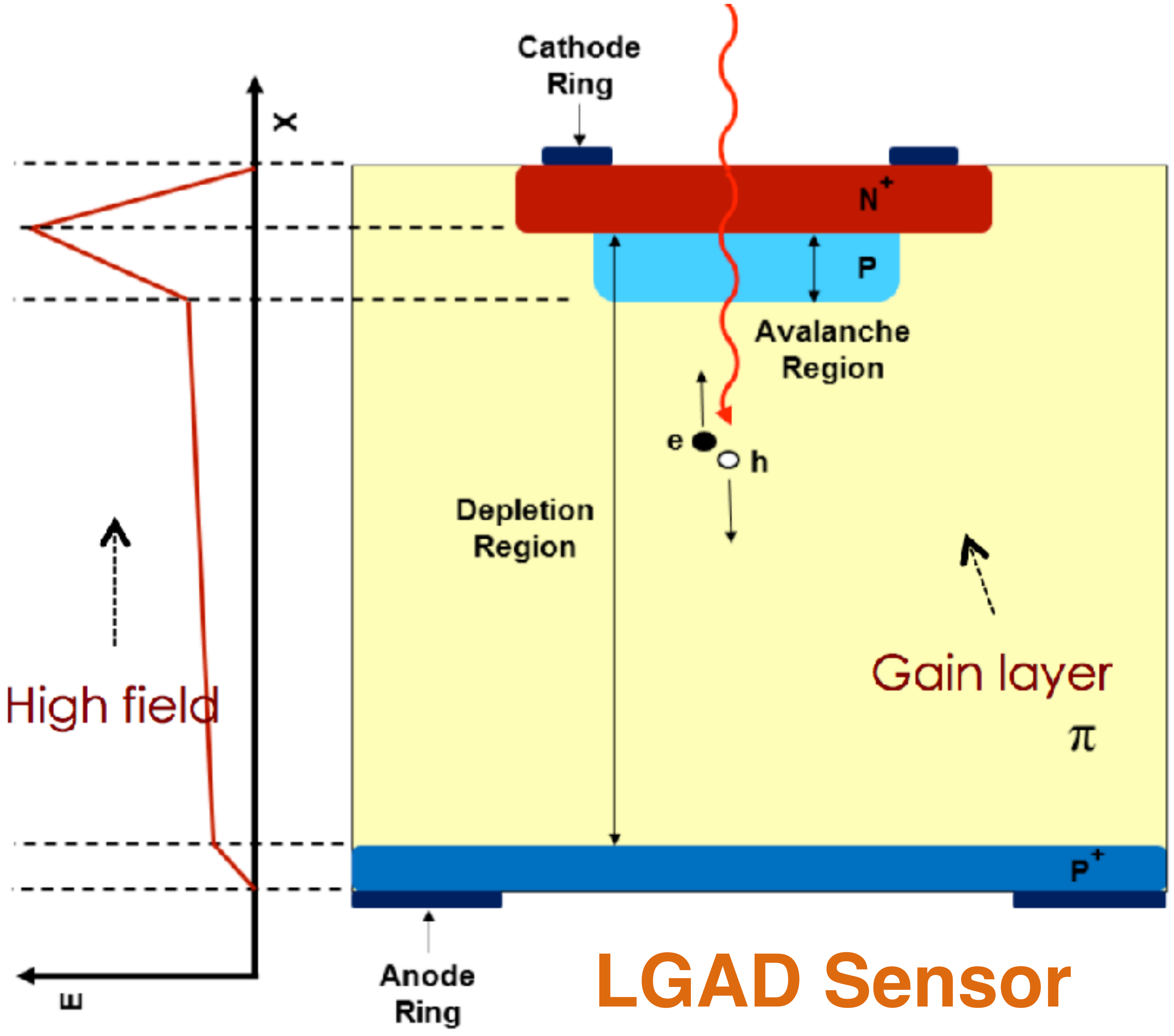
LGAD

- New detector in the CMS Phase-2 Upgrade (HL-LHC era).
- Maintain same performance despite increased amount of collisions.
- Two main components:
  - **Endcap Timing Layer (ETL) - Low Gain Avalanche Detectors (LGADs).**
  - Barrel Timing Layer (BTL) - LYSO crystals and Silicon Photomultipliers (SiPMs).

ETL Sensor Requirements

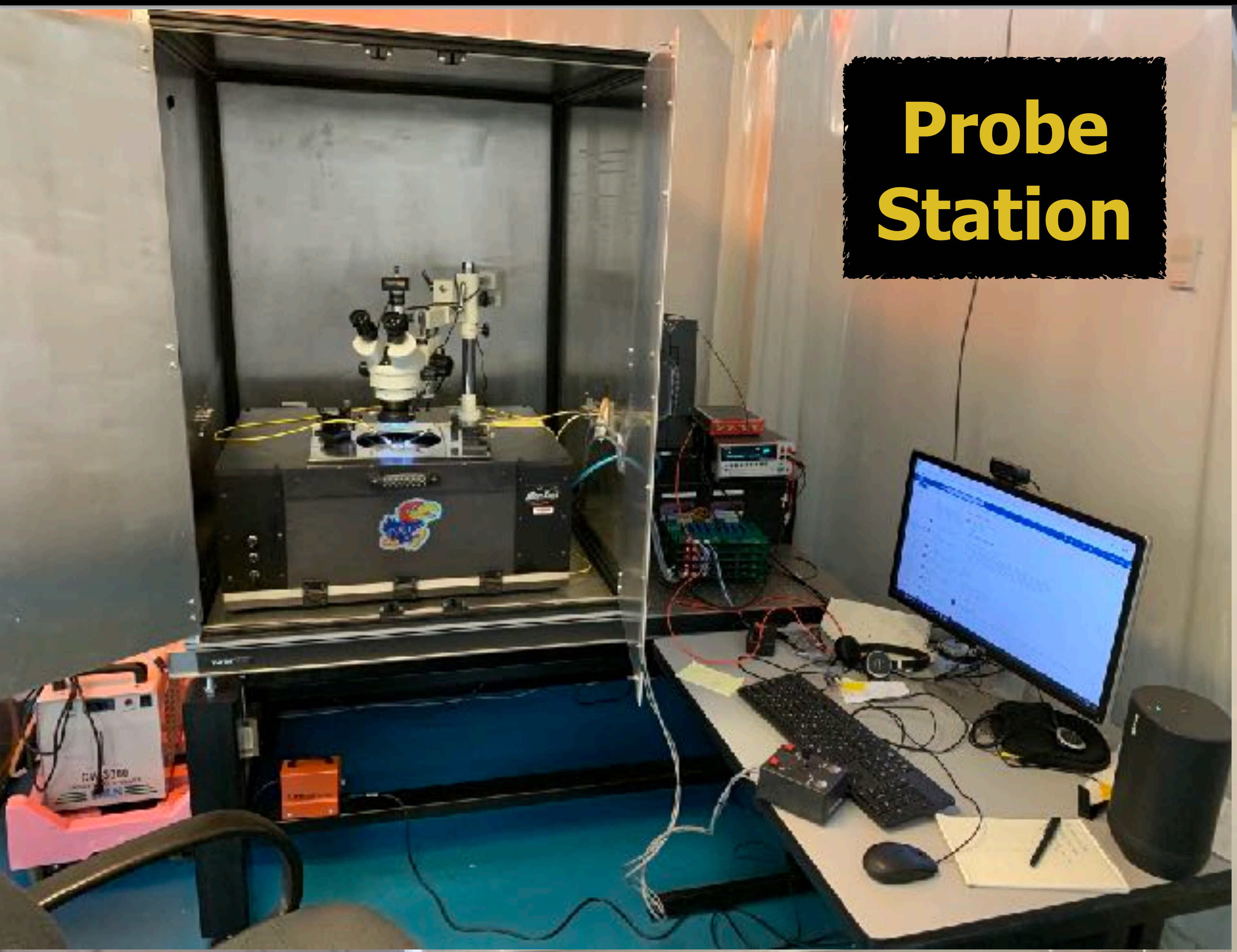
- **LGAD Requirements**:

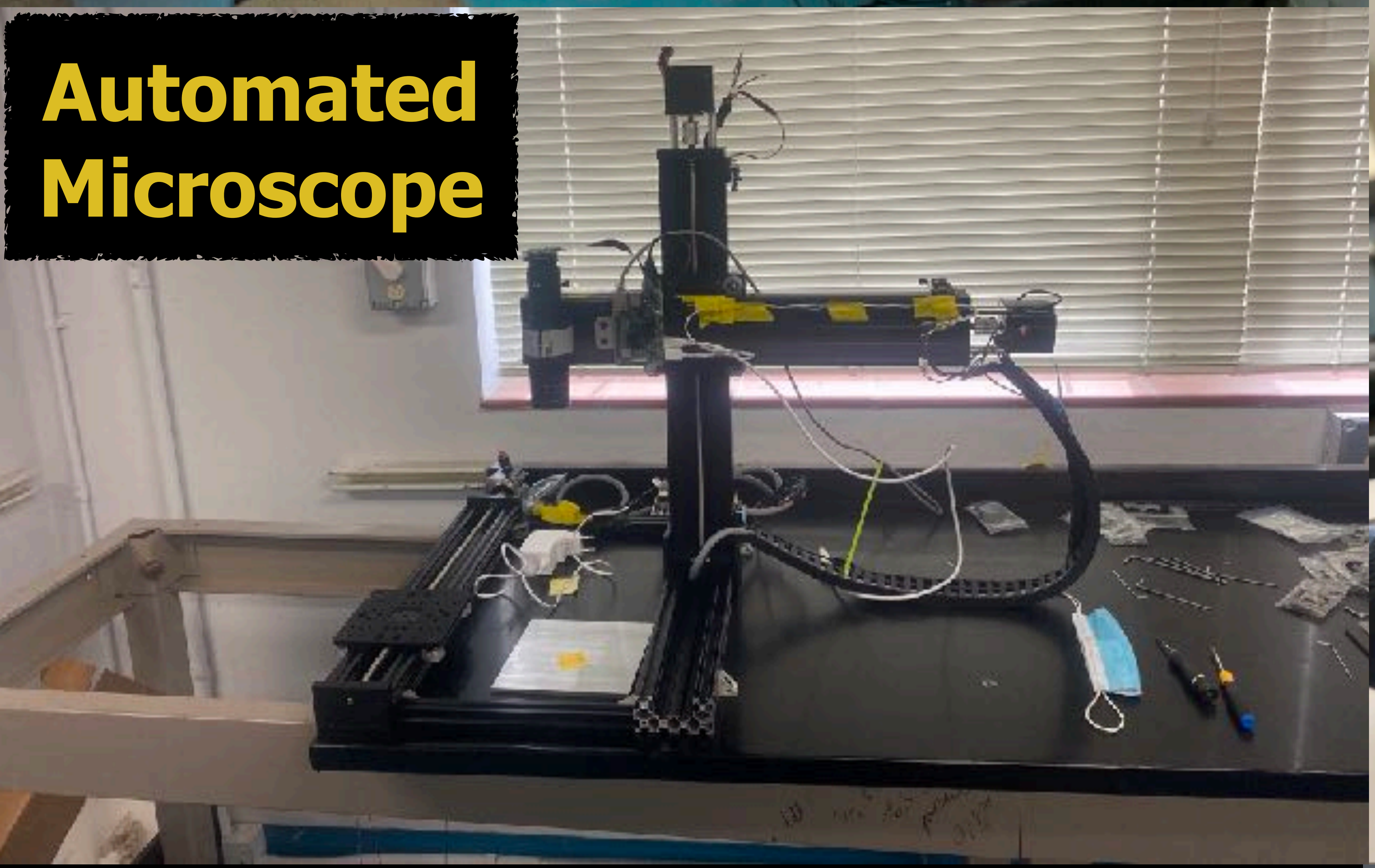
  - ~100% triggering **efficiency** across all sensitive elements.
  - Sufficient **radiation tolerance** to ensure consistent performance.
  - Maintain highly granular **timing resolution** throughout its lifetime (~30 ps).

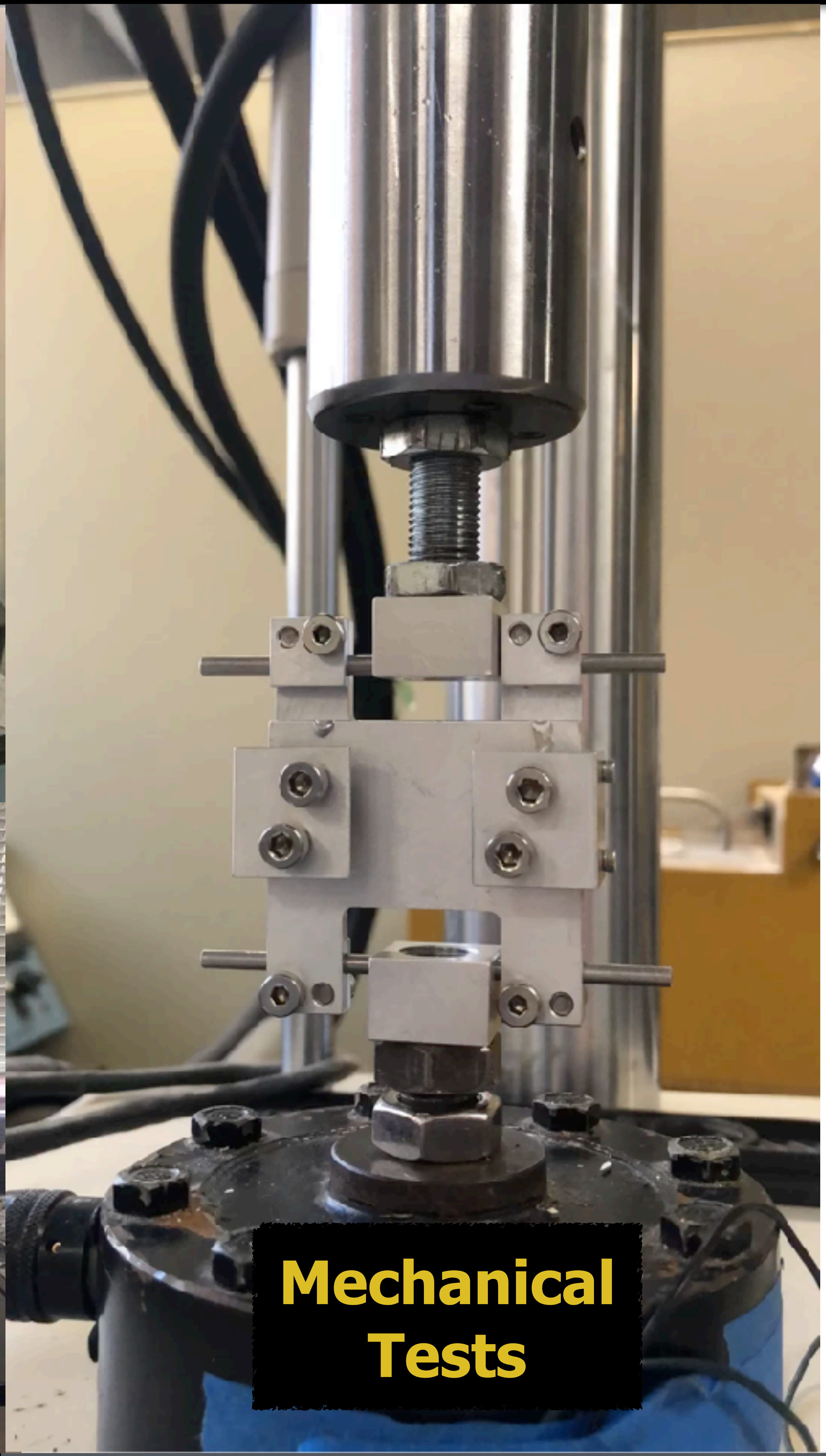


LGAD Sensor Concept

LGAD Testing at KU

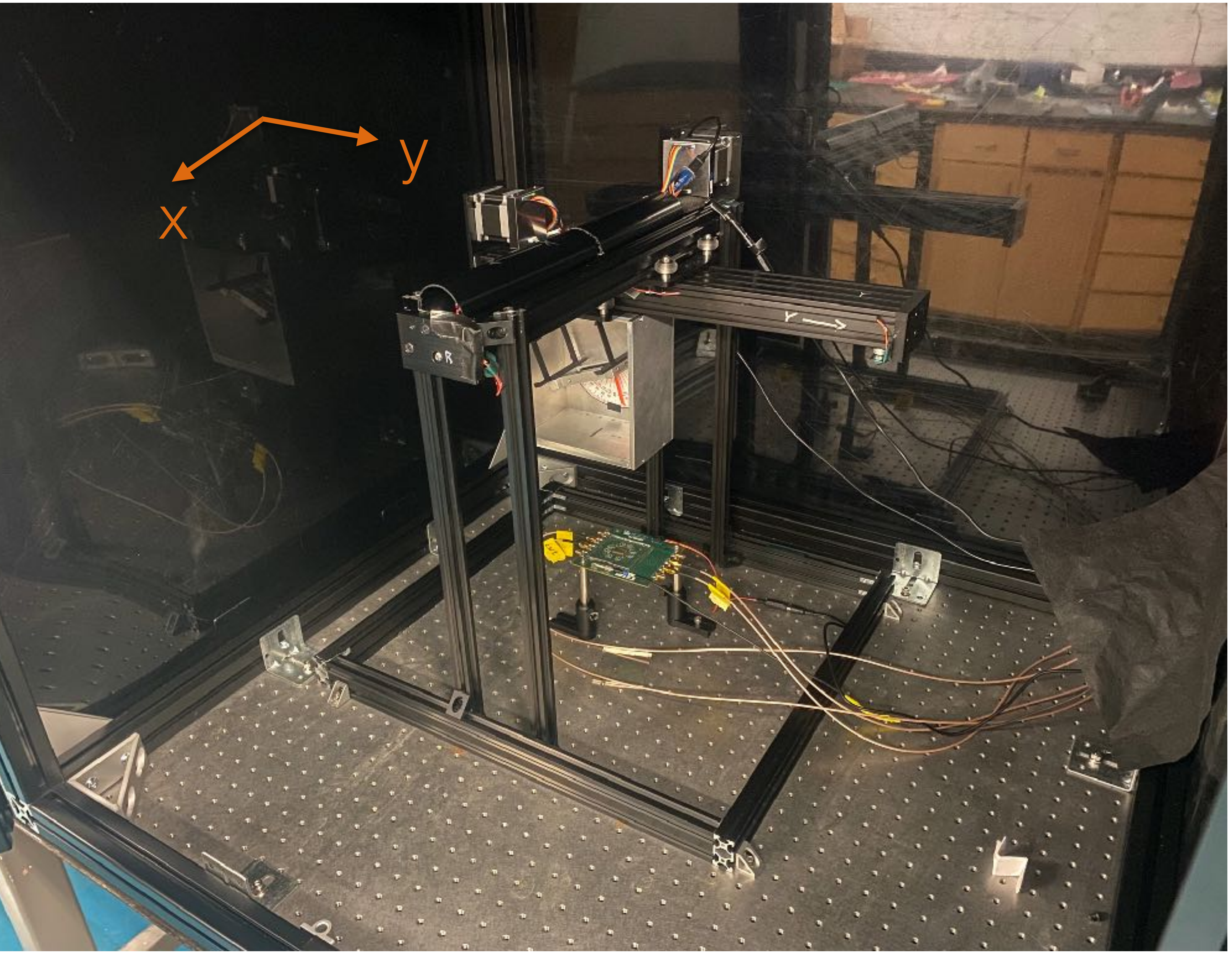






- **Several ongoing LGAD test projects at KU**:
  - Probe station (IV measurements, capacitance, etc.)
  - Mechanical tests (sheer and compression)
  - Automated microscope for high-resolution images
  - **Test Platform** system (time resolution measurement)

Test Platform System

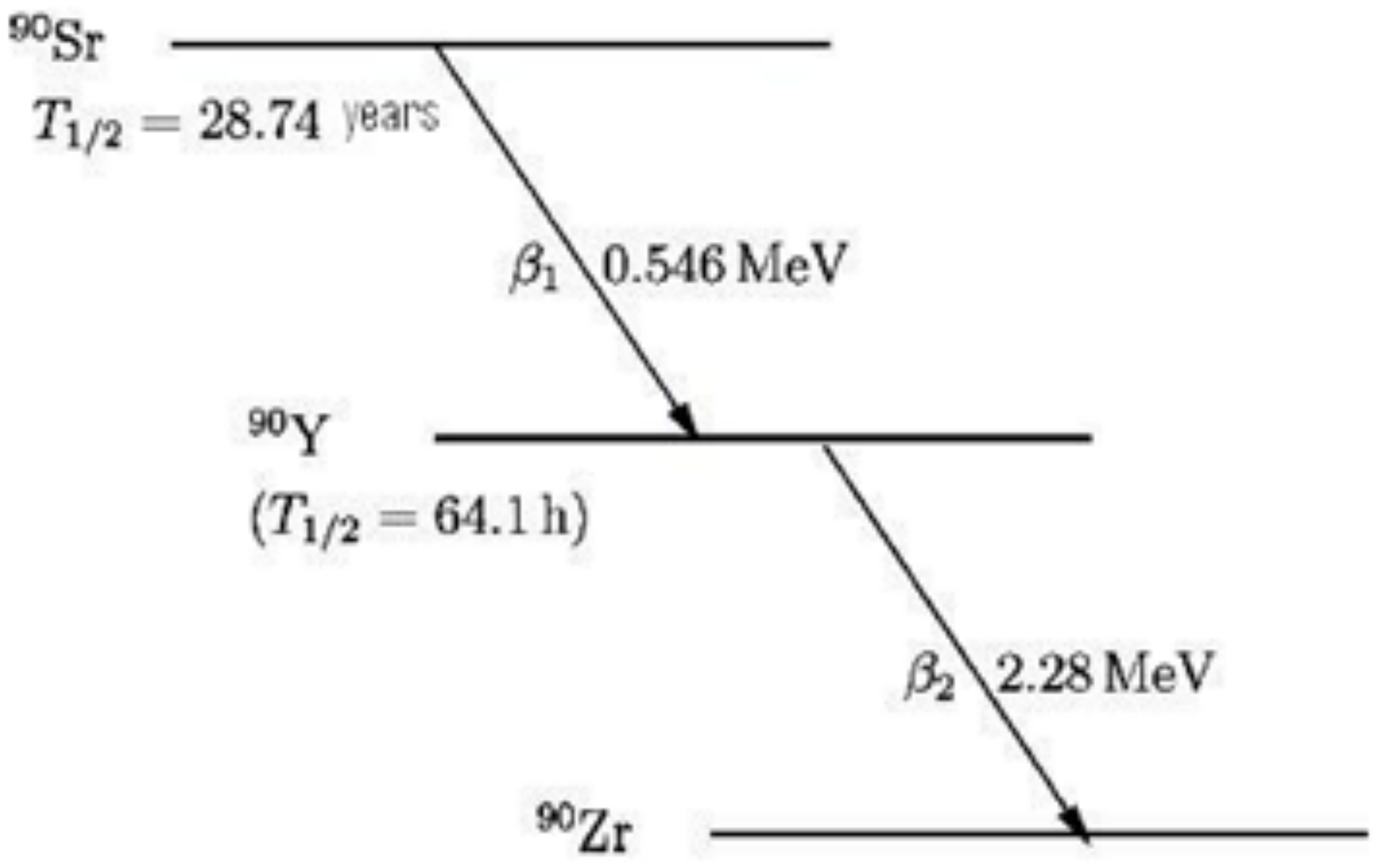


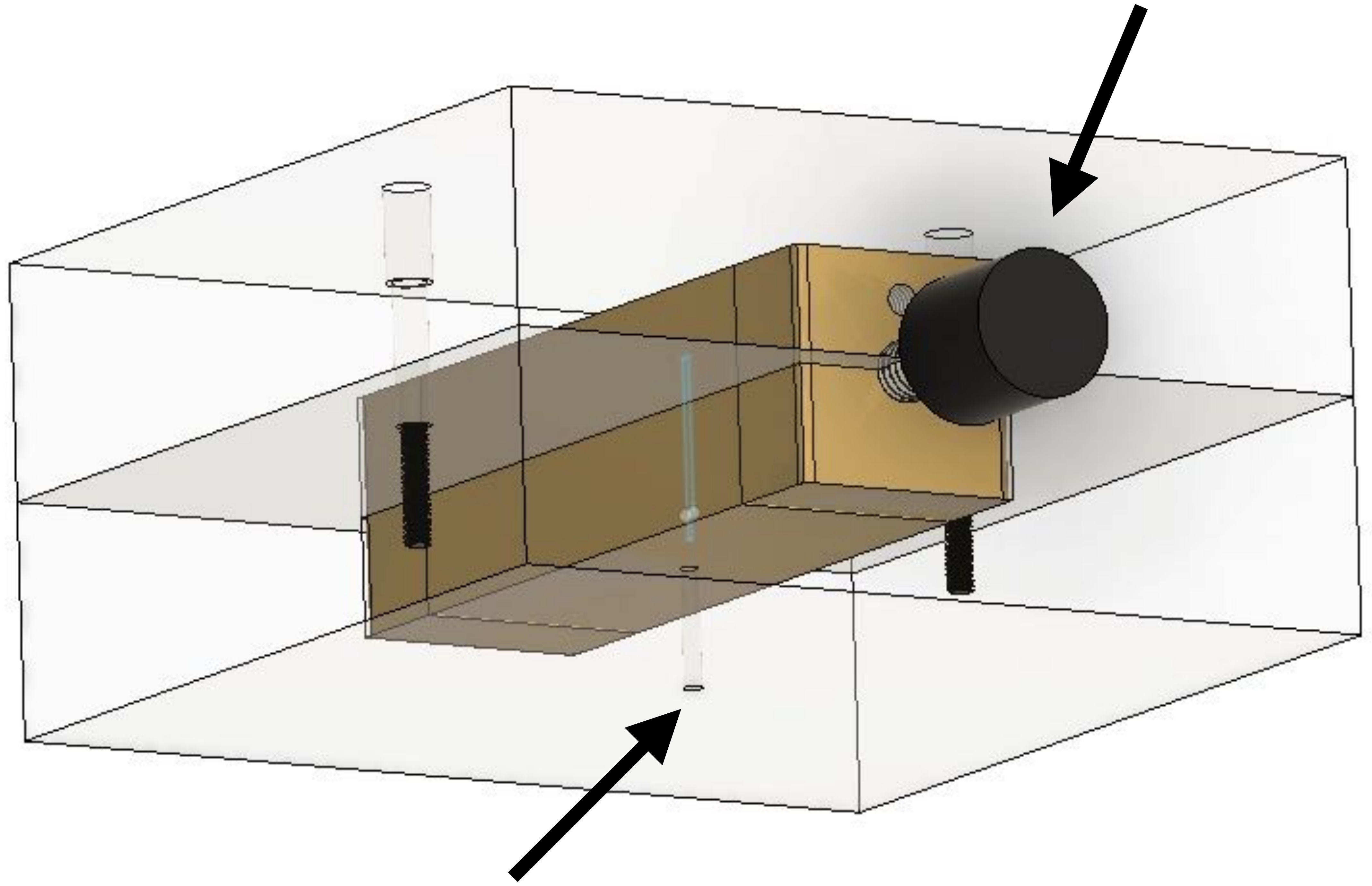
Test Platform System

- LGAD testing system with **movable radioactive Sr_{90} source**.
- Platform designed with **magnetic spectrometer** for selecting electrons with energy > 1MeV.
- **Mobility over an x-y grid** using electric motors and Arduino.
- **Reference sensor** to obtain time resolution.
- Readout data with **8-ch oscilloscope**.
- Plans to fully automate the testing procedure in the future!

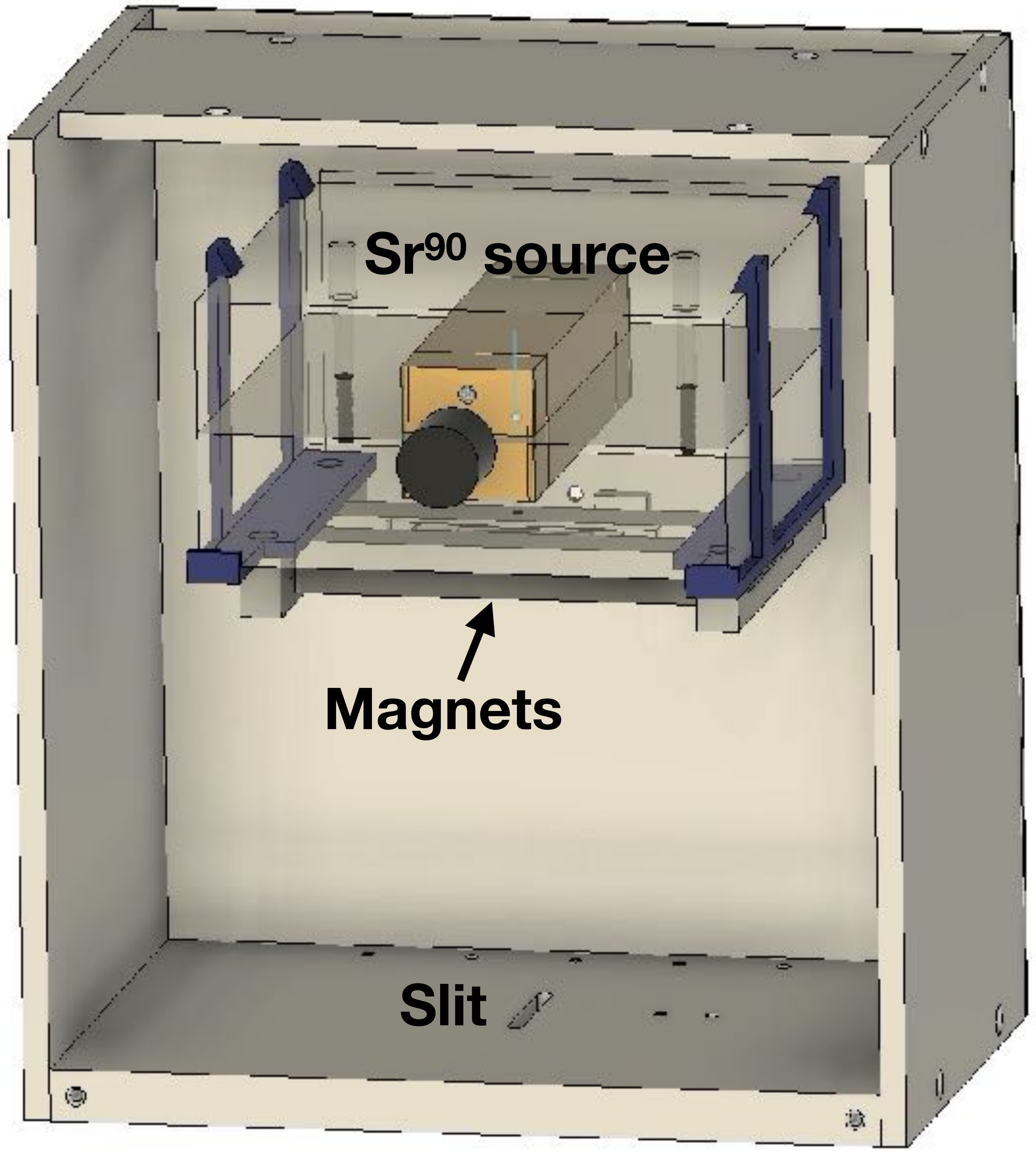
Sr_{90} Radiation Source

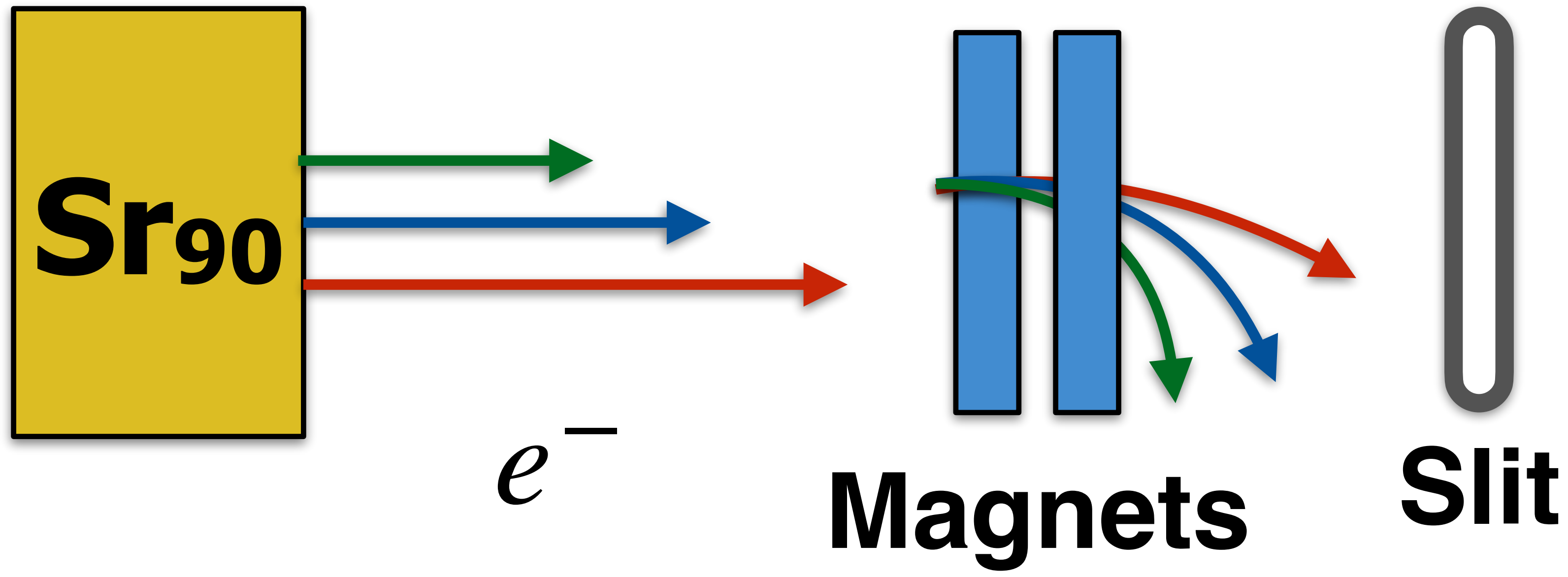
- Test sensors with β radiation
- Inside enclosure with pinhole
- Can be opened/closed with knob





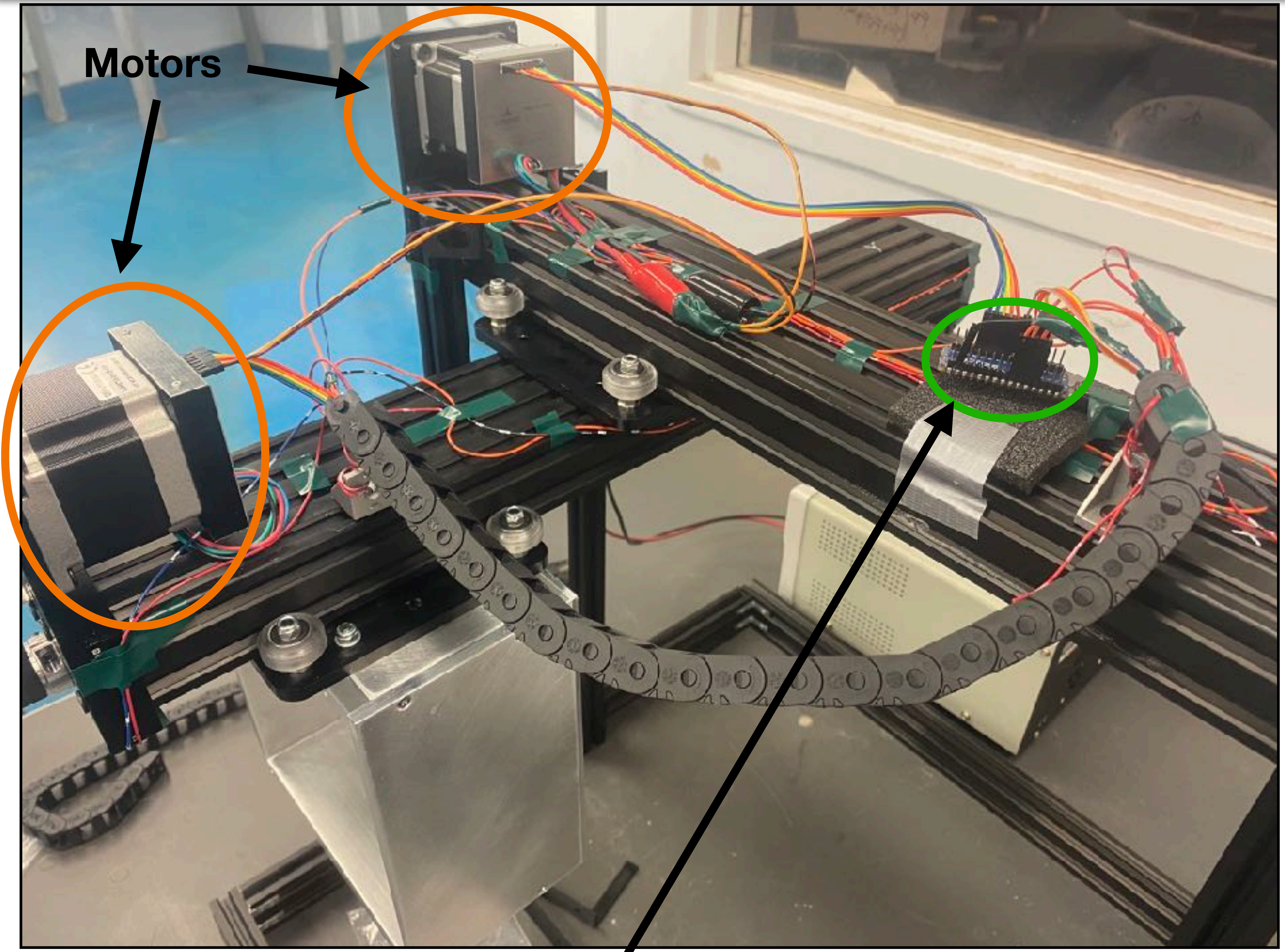
Magnetic Spectrometer

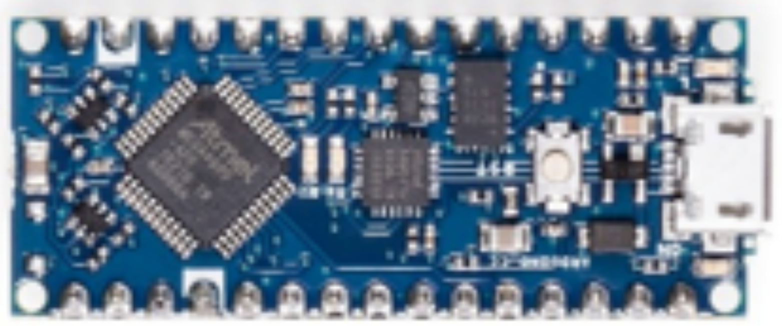




- Immediately underneath source pinhole opening.
- Used for selecting electrons with higher energies.
- Can be tuned depending on needs:
  - Magnetic field strength
  - Separation between magnets
  - Angle of source

Mobility Over X-Y Grid

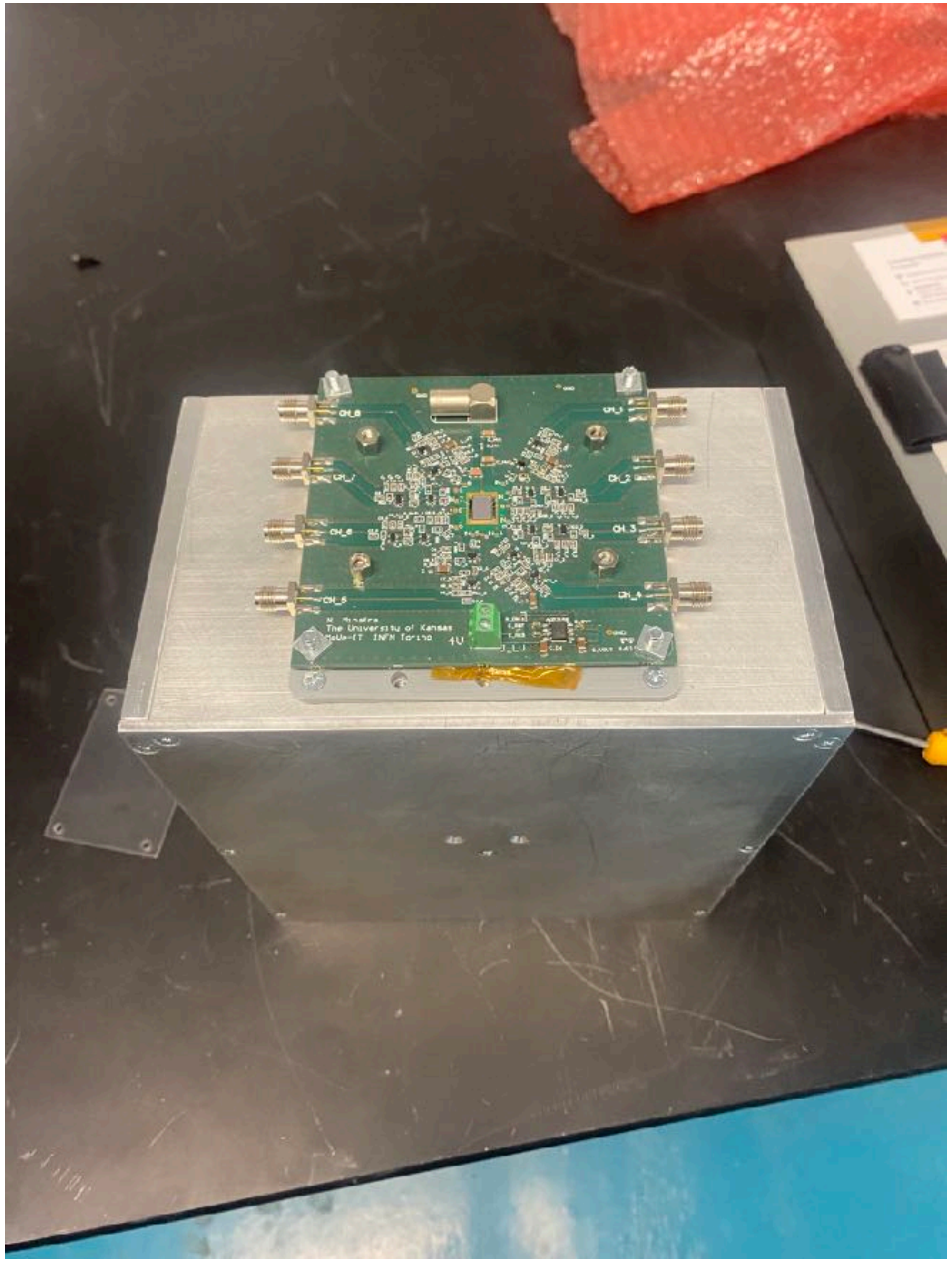




Arduino Nano Every

- Spectrometer with source can be moved over device under test (DUT) with 2 **stepper motors** (X-Y plane).
- Controlled using serial commands with **Arduino nano**.
- Can be used to test multiple LGADs in a relatively short amount of time.
- Plans for having testing procedure fully automated.

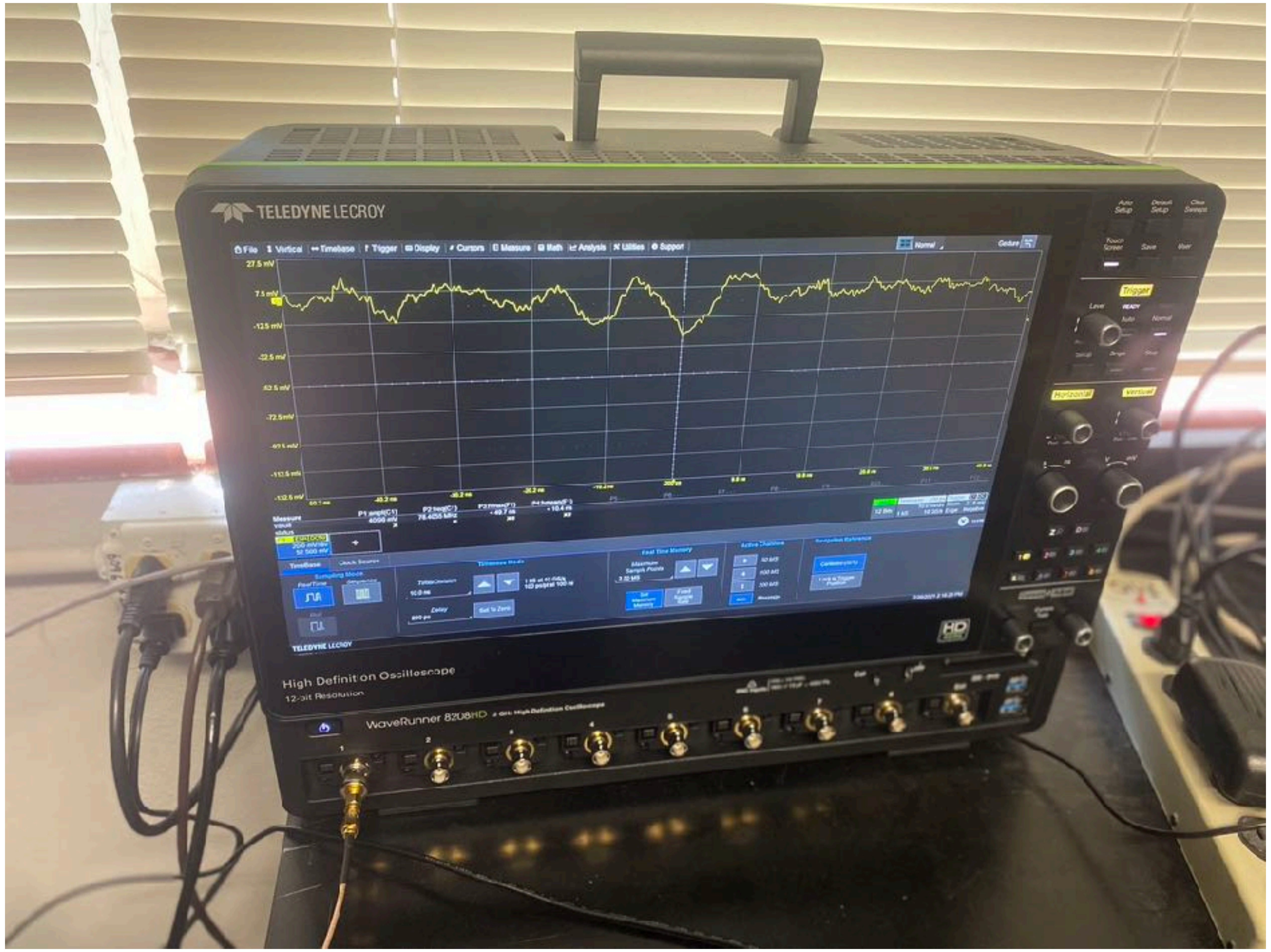
Reference Sensor



- Total of **8 working sensors** connected to readout channels.
- The read-out board has a cutout under the sensor to reduce material and prevent stopping electrons.
- Attached to bottom of source holder.
- Necessary for time resolution measurements.
- Currently testing which of the 8 sensors gives the best performance.

Data Acquisition

- **8-channel Oscilloscope**:
  - Used to control test platform motors.
  - Acquires and saves data for up to 8 sensors simultaneously.
  - Controlled with Python and stores data in HDF5 format.
  - Data analyzed using our own framework.

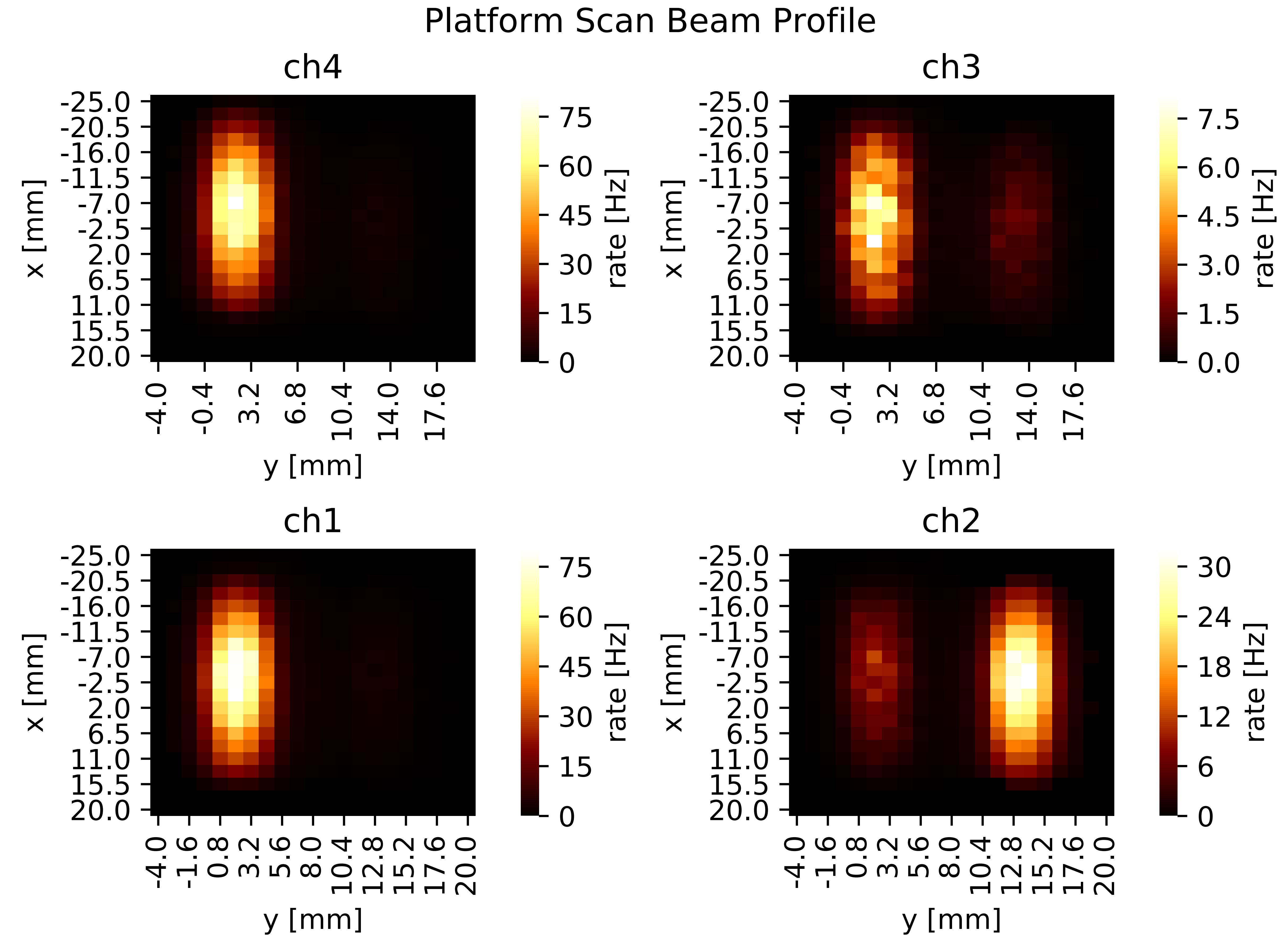


Analysis Framework

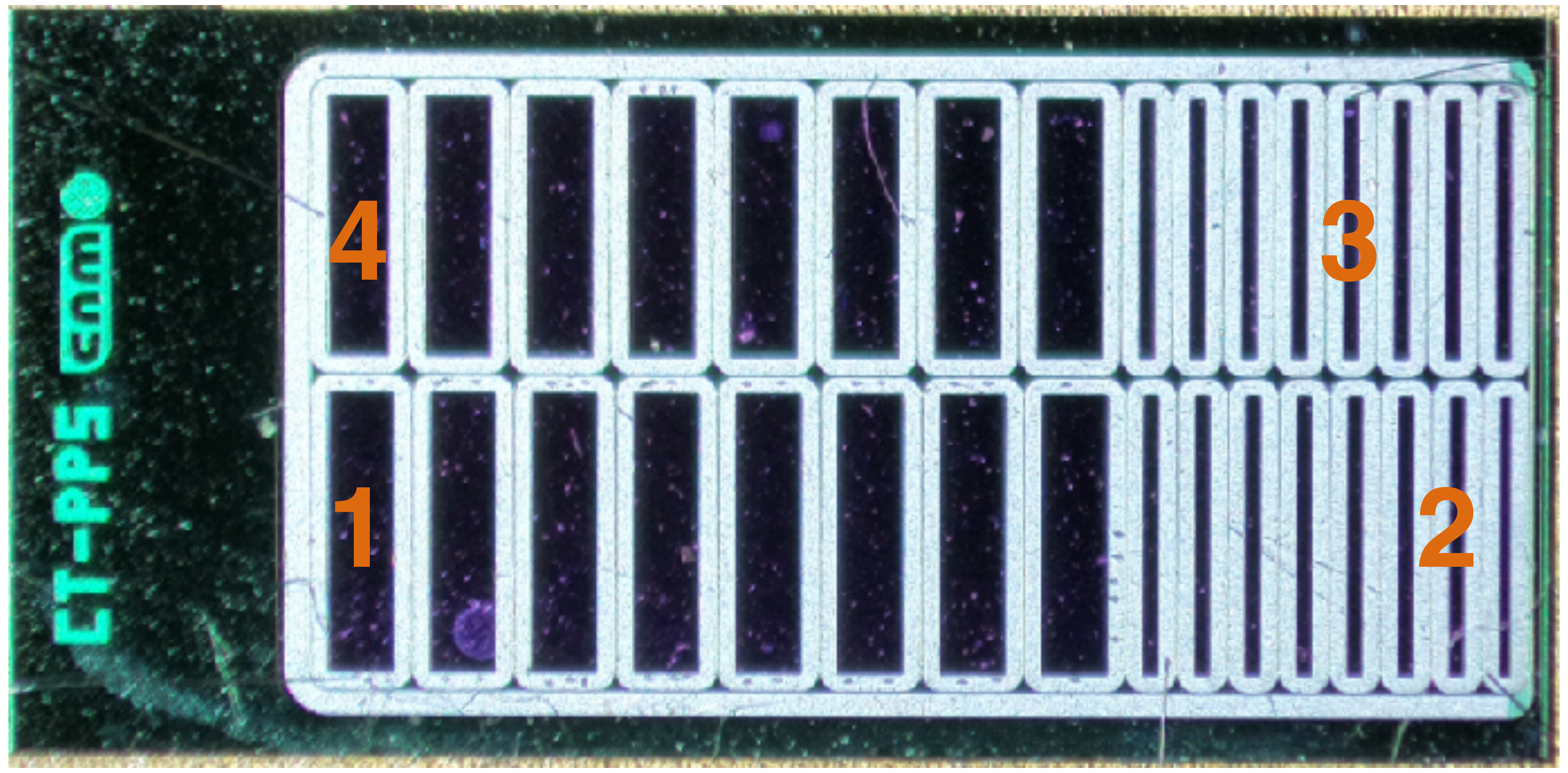
```
#include "PulseVariation.h"

//Constructors
PulseVariation::PulseVariation(const type){}
PulseVariation::PulseVariation(const PulseList& pulseList, const type){

  collectionSize_ = pulseList.size();


}

PulseVariation::PulseVariation(const std::vector<TH1D> histCollection, const std::vector<double> timeAxis){

  collectionSize_ = timeAxis.size();
  timeAxis_ = timeAxis;
  histCollection_ = histCollection;
  collectionMean_ = GetMeanPulse();
}

//Destructor
PulseVariation::~PulseVariation(){}

//public methods
std::vector<double> PulseVariation::GetMeanPulse() const{

  int size = histCollection_.size();
  std::vector<double> meanPulse;

  for(int h = 0; h < size; h++)
    meanPulse.push_back(histCollection_[h].GetMean());

  return meanPulse;
}
```

- Written mostly on C++ with some python scripts.
- Designed for analysis on single pulses and groups of pulses.
- Freely available on Github.
- Work-in-progress.

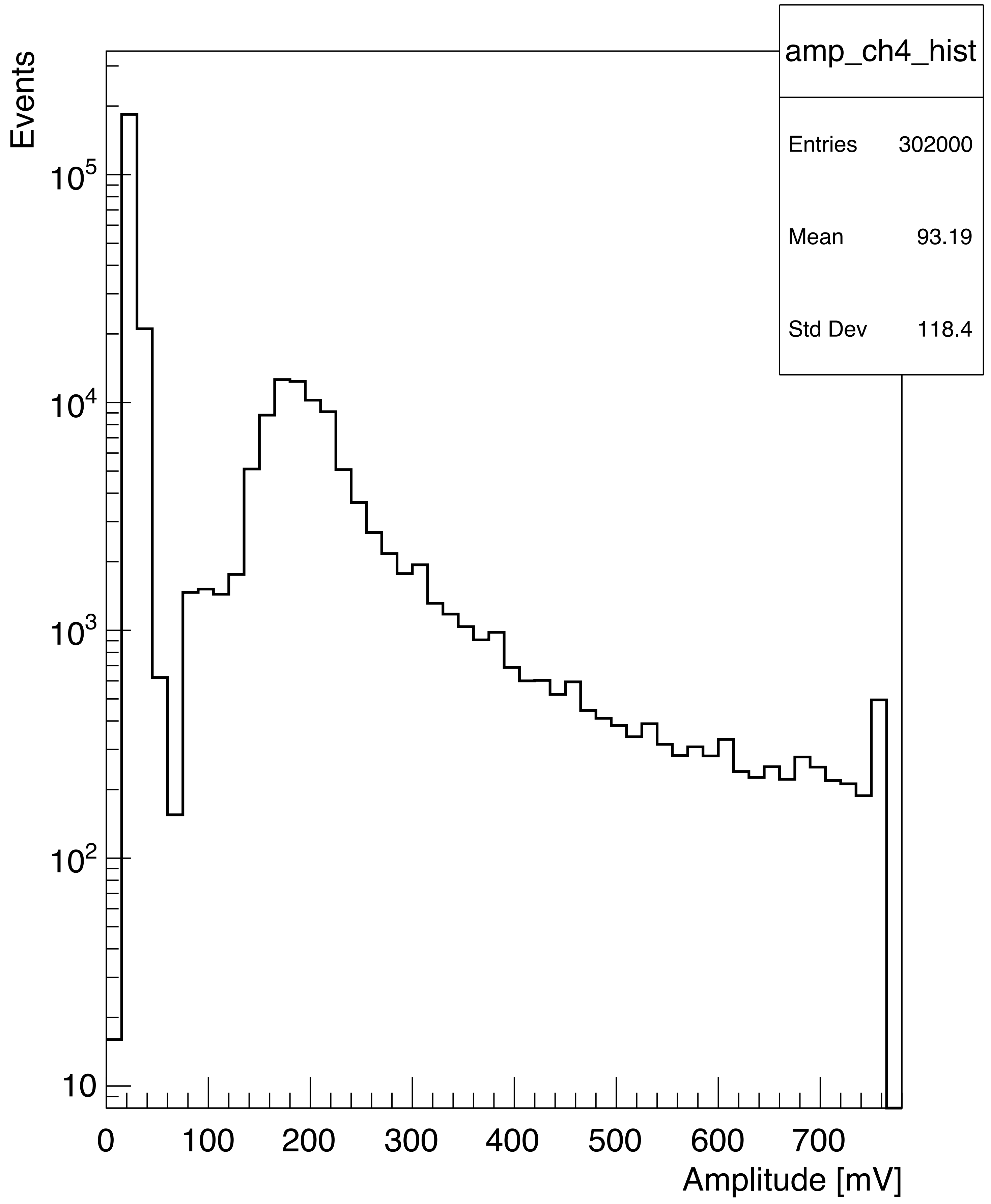
Trigger Rate

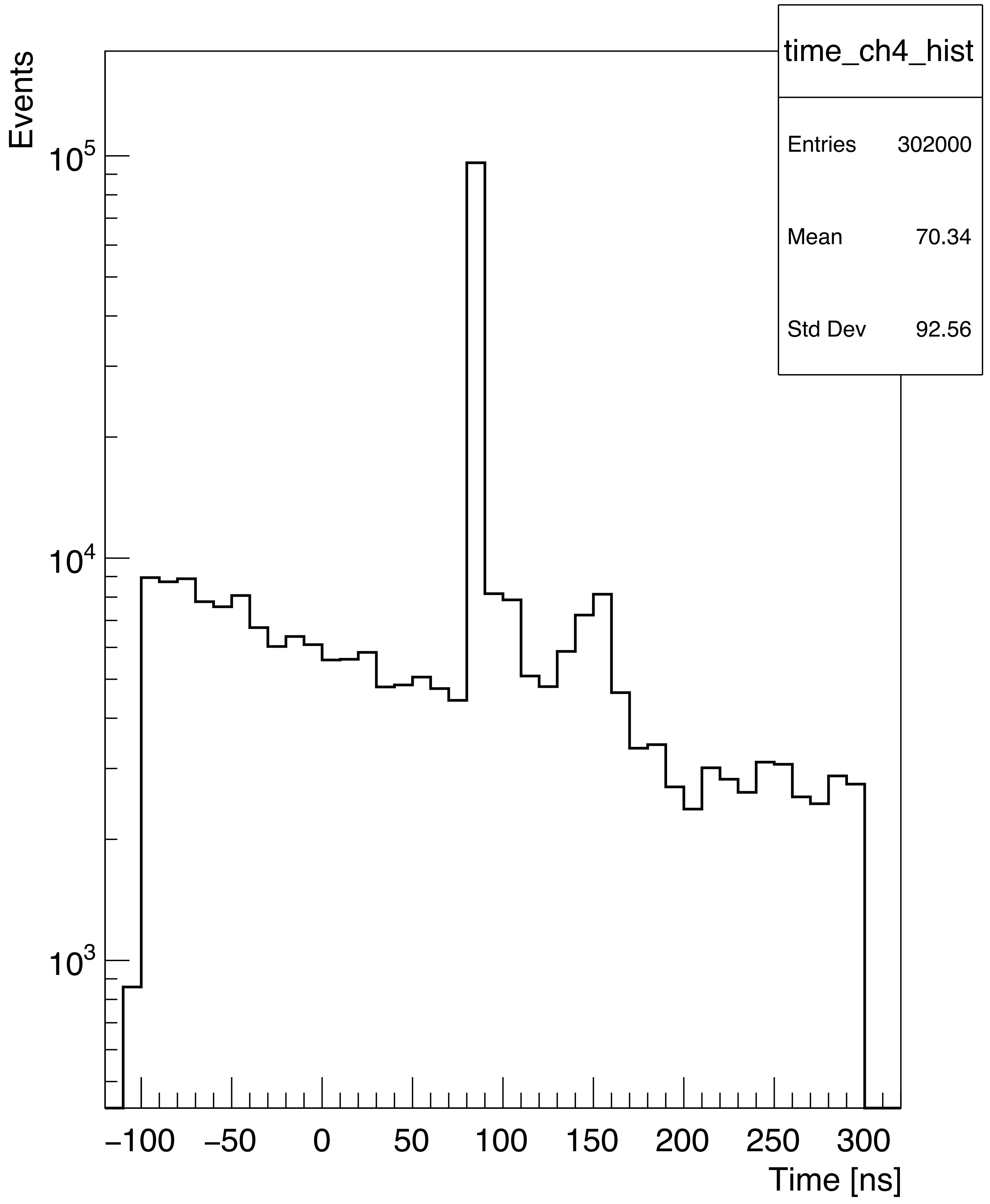


- First test with the source:
  - Scan over test sensor to find the highest rate in terms of platform position.
- Shape of profile depends on shape of opening.



Amplitude and Time Distribution

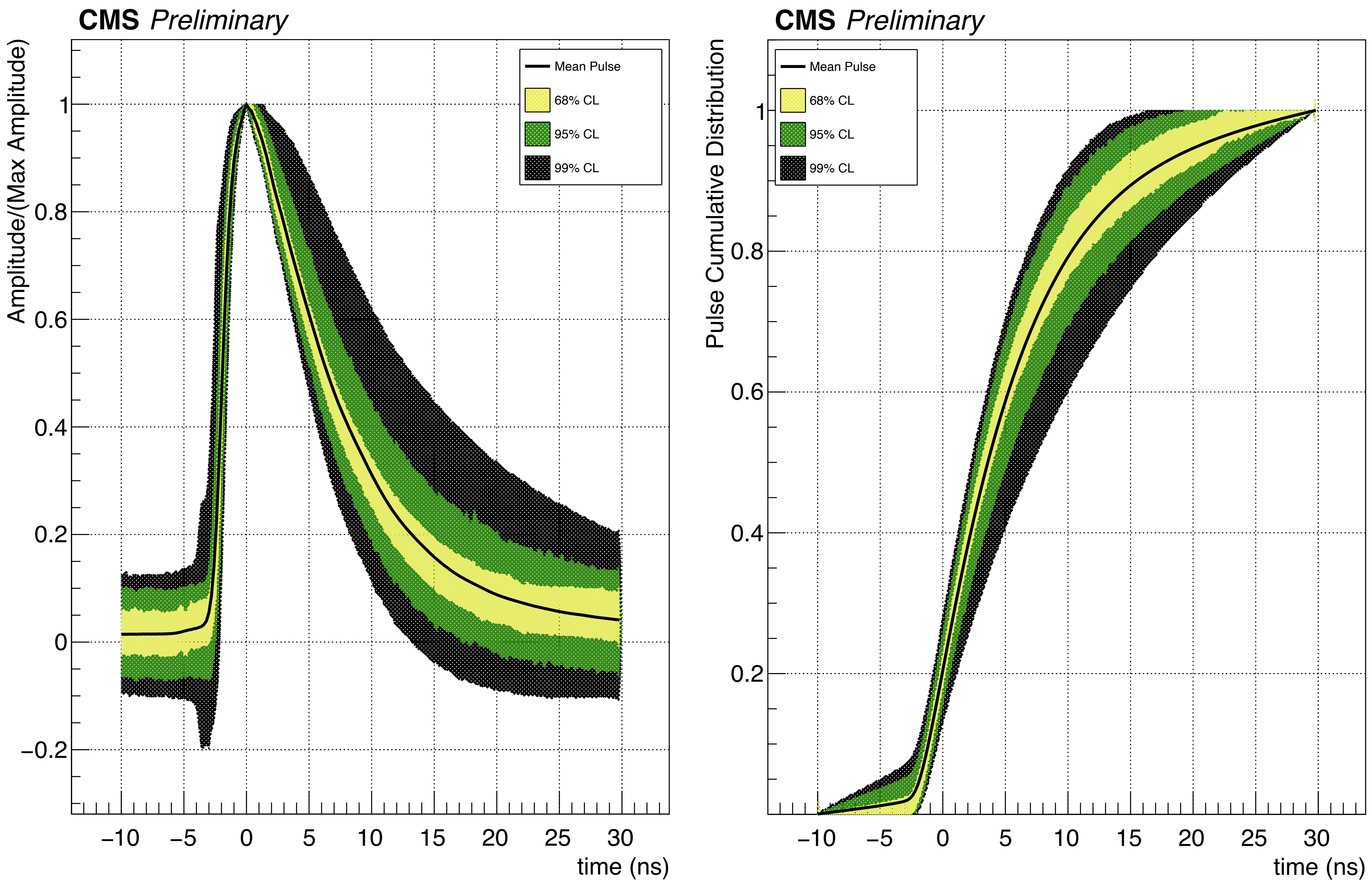




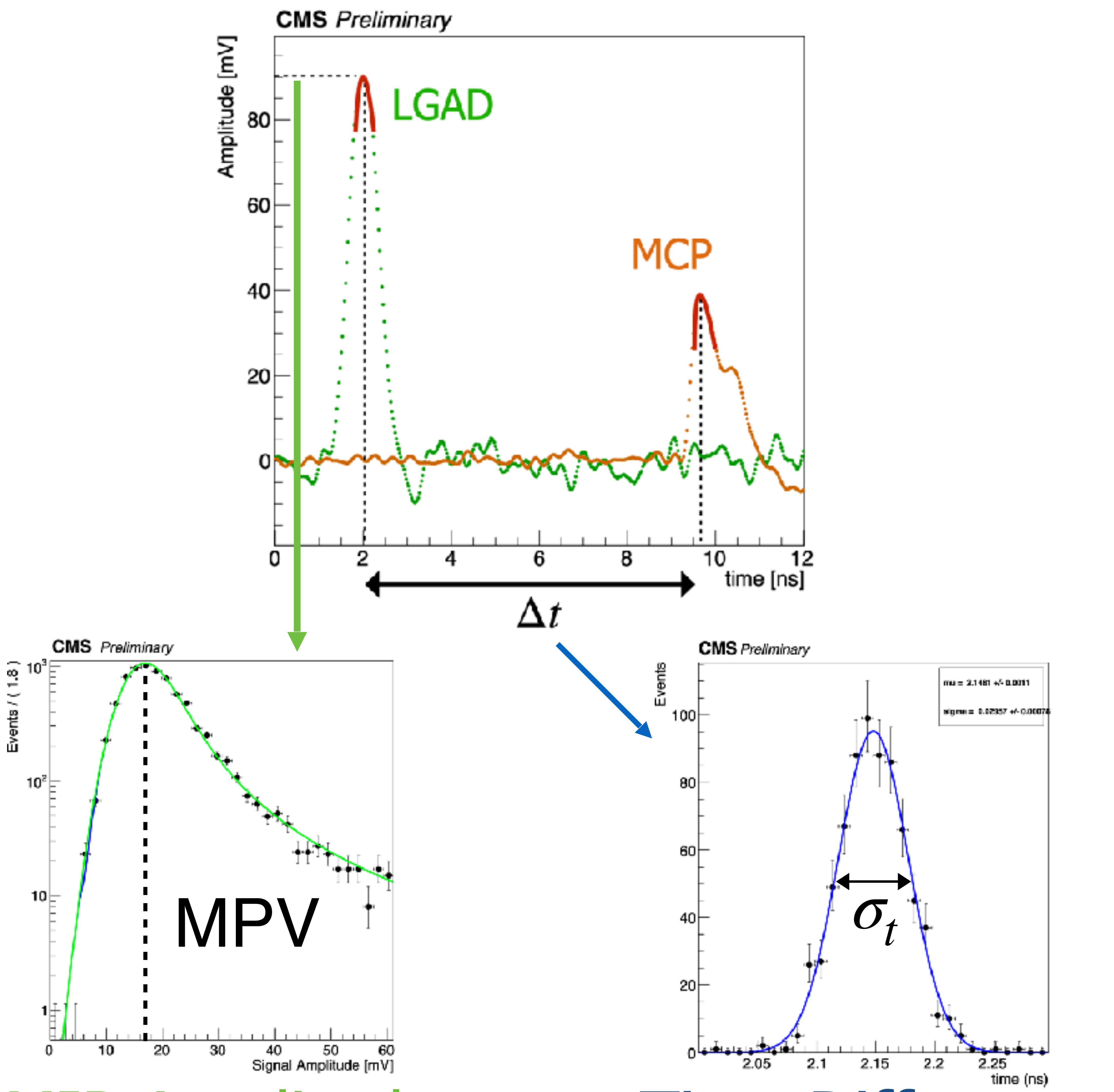
- Histograms made with the root library.
- Made to verify expected sensor behavior.
- Can be used to remove events that are just noise.
- Can be determined at each individual position.

Pulse Variation

- We can look at variations exhibited by pulses across many different events:
  - Certain parts of the pulse vary more than others.
  - Same is true for the cumulative distribution function (CDF) of the pulses.



Event Reconstruction and Timing Resolution



MIP Amplitude

Time Difference

- Obtain amplitude and Δt from each event:
  - Fit and interpolation used to find time value of signal max.
  - Difference between LGAD and reference peak mean.
- MIP amplitude fit to obtain peak (MPV).
- Δt distribution fit with Gaussian to obtain width (time resolution is σ_t).

Conclusion

- Test platform will allow us to take large amounts of data locally at KU (no need for huge particle accelerators).
- Can test many sensors in a short amount of time.
- Lots of progress but still a lot of work to be done:
  - Plans for fully automated system.
  - Installation of cooling system for testing irradiated sensors.
  - Evolving analysis framework.