Unified Embedded System Development: From 3D-Printed PCB Prototyping to Raspberry Pi and Sorting and Search Algorithms

Nour Aboul, Midwood High School

Mentors: Avakash Mishra, Shorn Grant, Professor Kofi Bosompim, Professor Raul Armendariz

August 13, 2028

ABSTRACT:

The Queensborough Community College (QCC), along with other colleges, have used Arduino-Based systems for Cosmic Ray Data Acquisition (DAQ). 20 detectors have been devised with each a plastic scintillator sheet, Photomultiplier Tube (PMT), and front-end DAQ box. However, perf boards yield lots of issues like assembly issues and short circuits which leads to unstandardized perfboards where no two boards are exactly the same. The solution was to 3D print. Using CAD software, circuits were designed and 3D printed on an FDM printer where traces were extruded to place copper tape. To process the resulting multi-detector data, a algorithm was implemented which groups events that arrive within a configured time window (W) whilst disregarding background noise that fail to meet the minimum hit threshold (N).

BACKGROUND INFORMATION

Cosmic rays are continuously bombarding the Earth's atmosphere. These are high-energy particles, dispelled from astrophysical phenomena, such as solar flares, stellar black holes, and explosive supernovae. These subatomic particles are primarily protons (89%) (CERN, 25). These particles collide with atmospheric atoms, producing a nuclear cascade that initially produces unstable pions ($\pi^{\pm}$) which rapidly decay into muon particles ($\mu^{\pm}$) (Samson, 2020).

To capture these muons, a front-end Data Acquisition (DAQ) hardware is required. On of these components includes, perfboards. These low-cost boards are composed of insulated and conductive materials like copper and fiber glass and are meant for prototyping circuits. However, they come with certain drawbacks. Assembly errors are very likely to occur. Placing components on a blank perfboard grid is entirely reliant on manual pin counting and schematics (especially hand drawn) tend to be confusing as well. This leads to no prototypes like another and no standardized model. Additionally, when soldering components short circuits may occur when soldering paths overlap.

On the other hand, 3D printing seamlessly resolves this issue. Through Autodesk Fusion 360, schematics can be drawn and its traces can be elevated using the FDM printer. Then copper tape may be wrapped onto the extruded traces and the components' places may even be labeled as well.

To simulate this, a simple circuit was created where the potentiometer forms a parallel branch from a series resistor and LED diode. Both of which share the same Arduino voltage supply and ground node (See Figure 2 and 3). The following section details the formulas used for circuit analysis:

Equation 1 expresses Ohm's Law, the mathematical relationship between voltage, current, and resistance in a circuit.

Where: $$V=IR \quad (1)$$

V = Voltage measured in volts (V).

I = Current measured in ampere (A).

R = Resistance measured in ohms (Ω).

Equation 2 expresses Kirchhoff's Voltage Law (KVL), the mathematical equation in regards to the net sum of voltage in a closed circuit.

Where: $$\Sigma V=0 \quad (2)$$

ΣV = The algebraic sum of all voltage drops in a closed circuit must equal to 0.

Equation 3 expresses Kirchhoff's Current Law (KCL), the mathematical equation in regards to the net sum of voltage at a node.

Where: $$\Sigma I=0 \quad (3)$$

ΣI = The algebraic sum of all of the current must equal to 0 at a node.

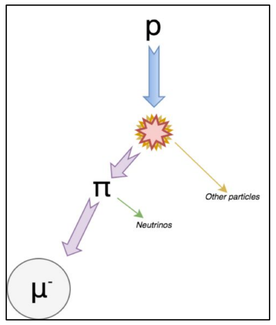
Following hardware implementation, output logs from the Arduino and Raspberry Pi DAQ units are processed using algorithm to discern genuine air-shower from background noise. Because individual background noise and genuine shores are physically indistinguishable for a singular detector, the algorithm will be tasked to sort the exported hit timestamps from N detectors in chronological order. Then, it will evaluate the data and group detectors that occurred during the time window (W=2.0s) and exceeding the minimum detector threshold (N=2). These detectors will be flagged as a cosmic shower while filtering out the background noise.

DISCUSSION and CONCLUSION:

This ongoing study demonstrates a possible alternative to traditional perforated board prototyping by combining 3D-printing circuit with an automated algorithm that filters fir cosmic ray detection. While the hardware development remain in progress, the initial fabrication revealed important constraints. This includes soldering as it leaves no room for error due to the heat from the soldering iron far exceeding the melting point of the plastic PCB. Extra CAD fine tuning is necessary for proper printability. The FDM printer requires a smaller nozzle for precise printing. Lastly, the age of copper directly affects the adherence. In regards to the software side, the coincidence window algorithm (W = 2.0s, N ≥ 2) is projected and only expected to accurately rejected isolated background events along with other tasks. Future work will be necessary in regards to printer optimization and deploying the algorithm with real data.

METHODS:

First a new schematic design was created on Autocad Fusion 360 by sourcing the components with their correct physical footprint. Then electrical connections were established using NET wiring tool (see Figure 3a). The design was switched to a 2D PCB to adjust the layout. The traces were defined in bottom of the pcb with a with of 2.0 mm. Next, the design was converted to 3D PCB where "package","1-copper-1", "64-soldermask" were hidden through the display settings (see Figure 3b). Then, the file was exported as a STEP file and the traces were extruded to a height of 1.5mm. Lastly, in Bambu Studio, "whole compensation" was set to 0.02 mm and "thin wall detection" was activated before it was 3D printed.



Source: Journal of Emerging Investigators

Figure 1. Simplified diagram of Cosmic Ray Shower: protons (p), producing pions, producing muons, along with other secondary particles.

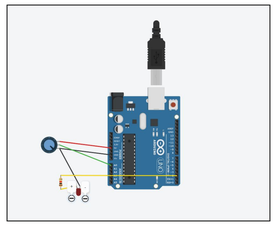


Figure 2a. Digital Arduino Setup of PCB.

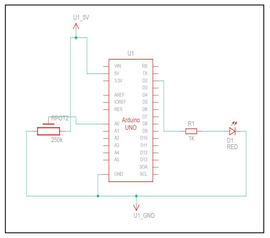


Figure 2b. Schematics of PCB.

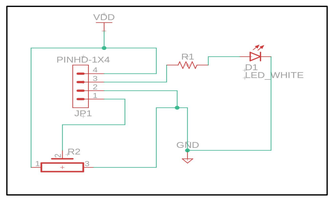


Figure 3a. Schematics of 3D printed PCB.

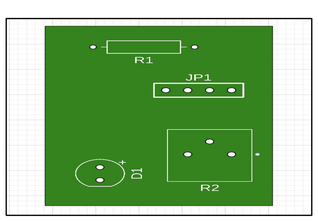


Figure 3b. 3D version of schematics.

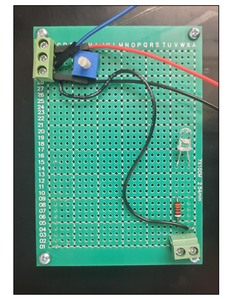


Figure 4. Physical perfboard.

Figure 5. Printed perfboard with components and copper tape on extruded traces.

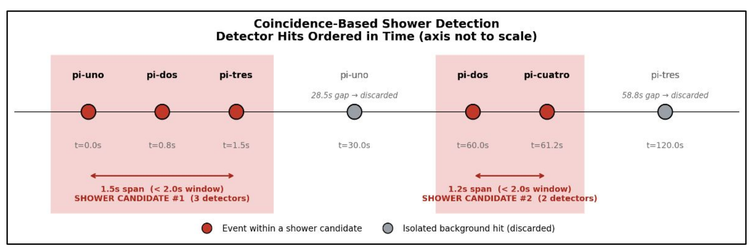


Figure 6a. Illustration of the coincidence-clustering method on a sample detector sequence.

(Refer to 6a) Detector hit timestamps are sorted chronologically and grouped a sliding coincidence window: events occurring within W seconds of one another are treated as a single shower candidate. Isolated hits below the minimum detector-count threshold are discarded as background.

(Refer to 6b) The snippet of code see in Figure 6b functions as the core logic of the algorithm. First events are ordered chronologically and the data is structured such that temporal relationships become strictly localized. Then, a single linear traversal then compares contiguous event pairs, aggregating proximate items into clusters based on the predefined time window (W=2.0s) all while filtering the background noise.

Figure 6b. Snippet of algorithm.

REFERENCES:

ACKNOWLEDGMENT:

I would like to kindly thank Professor Raul Armendariz and Professor Kofi Bosompim for the opportunity to conduct research in the lab. I would like to kindly thank my mentors, Shorn Grant and Avakash Mishra, for their patience and support. Additionally, I would like to thank Dr. Brian Aguirre and Professor Paul Marchese for the weekly workshops and meetings. Lastly, this work was supported by the CUNY STEM Research Academy through a grant from The Pinkerton Foundation.