The Instruction Manual
CosmicWatch
The Desktop Muon Detector (v3X)

Contacts:
Prof. Spencer N. Axani (saxani@udel.edu)
Masooma Sarfraz (masoomas@udel.edu)
Miles Garcia (milesg@udel.edu)
Katarzyna Frankiewicz (katarzyna.frankiewicz@gmail.com)
Prof. Janet M. Conrad (conradg@mit.edu)

August 20, 2025

Contents

Chapter 1

Document Overview

The CosmicWatch Detector v3X is a compact, self-contained, inexpensive (100$), low-power (0.5 W), particle detector that uses a plastic scintillator and silicon photomultiplier to explore particle, nuclear, astro-, and geophysics. Through these detectors we can explore a wide range of phenomena, including muon energy loss in materials, special relativity, radioactive decay, cosmic ray shower composition, and much more. The detectors support standalone data logging to microSD or live USB streaming, offers coincidence mode for background suppression, and logs rich event metadata (e.g. timestamp, ADC value, temperature, pressure, acceleration, and angular velocity). The v3X will be useful for education, outreach, citizen science, and more advanced cosmic-ray studies.

This instruction manual provides a comprehensive exploration of the CosmicWatch Desktop Muon Detectors v3X. It will discuss the details specific to the detectors, including the technology employed and provide instructions on how to build a detector yourself.

All material for v3X can be found in the GitHub repository located here:

`https://github.com/spenceraxani/CosmicWatch-Desktop-Muon-Detector-v3X`

Version v3X represents a significant advancement over the previous v2 detector. In addition to the significant improvements in the analog electronics, high-frequency PCB design, and component optimization, v3X delivers orders-of-magnitude enhancements across several performance metrics and also introduces new functionality, as shown in Table 1.1.

Components of this instructions manual has been lightly adapted from MIT's Junior Lab Manual, which was based off of Prof. Spencer Axani's Masters Thesis, accessible directly at `https://arxiv.org/abs/1908.00146`. However, this is a new detector, specifically designed to increase the CosmicWatch's physics, education, and industrial reach, as well as providing the backbone for a future citizen science project.

CosmicWatch v2	**Specification**	**CosmicWatch v3X**
ATmega328P	Processor	**Cortex-M0+**
Single Core	Number of cores	**Dual Core**
16 MHz	Clock frequency	**133 MHz**
32 KB	Flash memory	**2 MB**
2 KB	RAM	**264 KB**
0.27 W	Power consumption	0.5 W
<8-bit	ADC resolution	**12-bit**
15 Hz	Maximum event rate	**700 Hz**
0.5 mV RMS	Noise level	**0.1 mV RMS**
Yes	microSD Card	**Yes**
Yes (w/o SD card)	OLED readout	**Yes**
Yes	Temperature sensor	Yes
No	Pressure sensor	**Yes**
No	Accelerometer & gyroscope	**Yes**
No	Buzzer	**Yes**
25 cm^2	Effective area	25 cm^2
50 ms	dead time per event	**400 μs**
106 g	Dry weight	110 g
36 mm^2	Photocathode area	36 mm^2
No	Citizen science support	**Yes**
12 mV	Min. trigger threshold	**4 mV**
4.5×10^{-4} Hz	Accidental coincidence rate	**6.0×10^{-6} Hz**
$100	Approximate cost	$100

Table 1.1: Specification comparison between CosmicWatch v2 and v3X detectors. Bold entries indicate the specification with superior performance.

The original CosmicWatch project was significantly developed with co-collaborator Dr. Katarzyna Frankiewicz and Prof. Janet Conrad. The original project has had funding from the MIT Physics Department, the Wisconsin Particle Astrophysics Center (WiPAC), the National Science Foundation (NSF), and is now funded through the University of Delaware (UD).

We are excited to have the CosmicWatch detector incorporated into the Experimental Methods in Physics courses at UD. Developing this detector has been a passion, not just a hobby. We hope you enjoy it, and even consider building your own.

Chapter 2

The CosmicWatch Desktop Muon Detector (v3X)

2.1 Introduction to the Detector

The CosmicWatch Desktop Muon Detector (v3X) comprises a 5×5×1 cm^3 slab of plastic scintillator interfaced with a single photon sensor, known as a silicon photomultiplier (SiPM). When a charged particle traverses the scintillator and deposits some amount of its energy, a portion of that energy is re-emitted isotropically along the particle track in the form of photons. For a typical scintillator, for every 100 eV units of energy the particle deposits, one photon is created. Photons reaching the photosensitive area of the SiPM can induce a Geiger discharge in the SiPM microcells. The microcells, when discharged, generate a small but measurable current. A single photon can trigger a single microcell (ignoring second-order effects), while multiple photons may trigger multiple cells. Through a measurement of the amount of current (or voltage, in our case) of the signal, we can infer the amount of energy deposited.

The SiPM signal (a pulse of current) is directed through a custom-designed printed circuit board (PCB), which amplifies and shapes the signal to make it suitable for measurement by an inexpensive micro-controller (MCU), a Raspberry Pi Pico (RPi). For each event, the RPi logs the event number, event time in seconds relative to the detector start time, whether or not the event was flagged as coincident with another detector (binary flag 1 or 0 respectively), the measured 12-bit ADC value of peak detector waveform, calculated SiPM pulse height in millivolts, the cumulative detector dead time in seconds, temperature in Celsius, pressure in Pascals, the x-y-z linear acceleration, and the x-y-z angular velocity. If data is recorded via the provided software, three additional entries per event are also provided, including the detector name (modifiable), the Network Time Protocol (NTP) time stamp date stamp from the host machine.

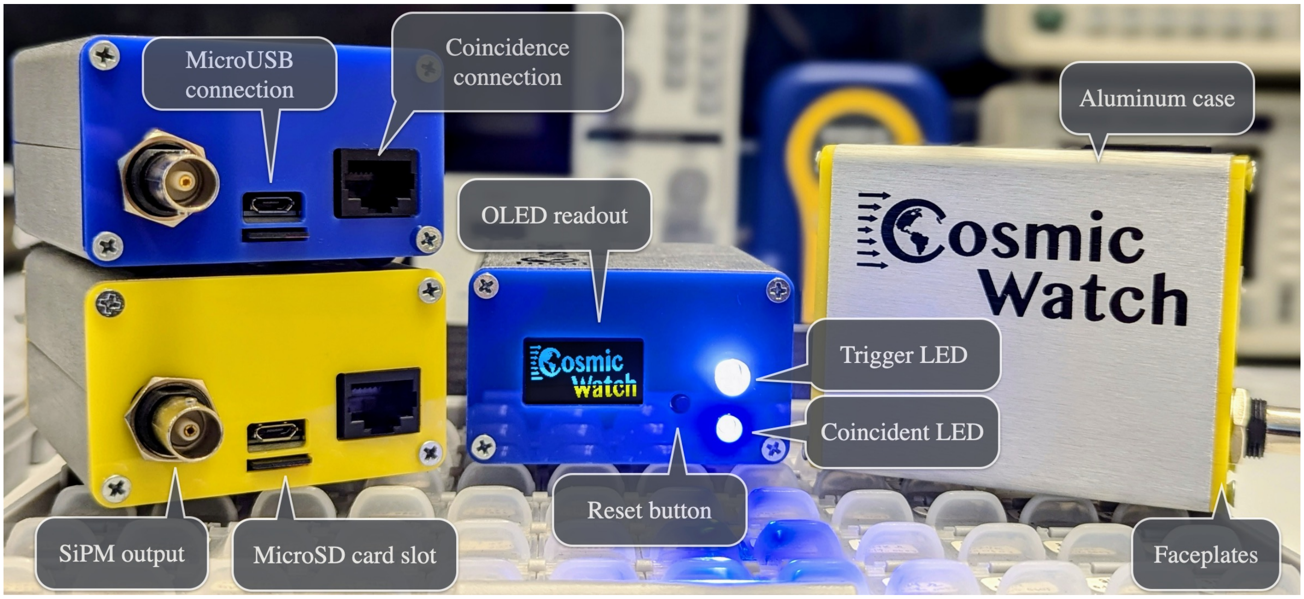


Figure 2.1: An array of the Desktop Muon Detectors.

An array of complete CosmicWatch v3X detectors is shown in Fig. 2.1. The front of the detector is equipped with a reset button, two LEDs, and a built-in 0.96" OLED screen displaying information such as the total run time, number of events, and count rates (accounting for detector dead time). The top 5 mm LED flashes on all events, while the bottom 3 mm LED triggers only on coincident events (events that simultaneously triggered connected detectors).

The backside of the detector features a micro USB port on the RPi for powering the detector, uploading new firmware, or transmitting data to a serial port on a computer. It also includes an Ethernet connection for linking multiple detectors to perform coincidence measurements (see Sec. 2.2). The Ethernet connection will later be capable of communicating with a *CosmicWatch Expansion Module*, which will enable further use of additional peripherals, such as GPS, magnetic field sensors, real time clocks, or anything that communicates over i2C, SPI, or UART. A microSD card socket under the USB enables a simply way to log the event-by-event data, with a new file being produced after each reset or power cycle. Additionally, a BNC connection connects directly connected to the SiPM output, and thus can be used to either inject pulses into the circuit or view the high fidelity SiPM waveform on an oscilloscope.

A single detector was measured to consume 0.5 W and can be powered through any USB connection supplying 1.8 V to 5.5 V. This could be a USB port on a computer, USB power bank, power outlet USB, or even a cell phone. The total mass of the detector, excluding the optional aluminum enclosure and end-plates, the detector has a mass of 83 g, whose measurements are illustrated in Fig. 2.2. Including the aluminum case, the mass is 110 g, and the case outer dimensions are 66.4 mm×70.0 mm×39.9 mm.

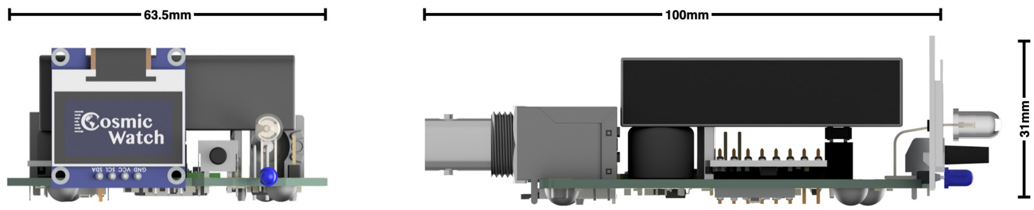


Figure 2.2: A rendering of the physical v3X detector.

2.2 Setting detectors in coincidence mode

Many measurements rely on the detectors operating in *coincidence mode*. This mode allows users to extra a rather pure sample of the cosmic-ray muon by rejecting events that likely came from radioactive backgrounds. Coincidence mode requires two detectors connected together using an Ethernet cable (we'll refer to it as the *coincidence cable*). Once connected, only one of the detectors requires power; the other will be powered through the cable.

To set the detectors into the coincident mode, simply reset both detectors within a second of each other while they are connected together. Alternatively, you can unplug the power cable (micro USB) and plug it back in. The detectors will acknowledge the presence of a coincidence detector by brightly illuminating both LEDs on the front panel for one second. An example of two detectors being set up in coincidence mode is shown in Fig. 2.3.

All triggered events are recorded. However, coincident events will be flagged in the data with a "1" in the coincident column (third column of the data), while singles events have a "0" in its place. In coincidence mode, the OLED display will display both the master and coincidence rate.

Below about 5 km, coincident events are overwhelmingly likely to be due to a cosmic-ray muon, particularly under $\geq$15 cm of concrete where the electromagnetic component of the shower is significantly attenuated. Radiogenic backgrounds and accidental coincidences are unlikely to trigger both detectors simultaneously. A summary of why the purity of the cosmic-ray muon signal increases in coincidence mode is provided below.

- Alpha particles will not penetrate a single detector (either the aluminum enclosure or even the black electrical tape) and therefore cannot trigger both detectors at the same time.
- Beta particles (electrons and positrons) can be significantly attenuated by the aluminum case, and have a significant chance of scattering, thus losing energy. It's unlikely that the beta particle will be able to deposit sufficient energy within the scintillator (making it paste the black tape and aluminum foil) in one of the detector, exit, then depositing sufficient energy in the second detector. However, there exists a rather substantial sub GeV electron/positron flux that develops in cosmic ray showers that has sufficient energy

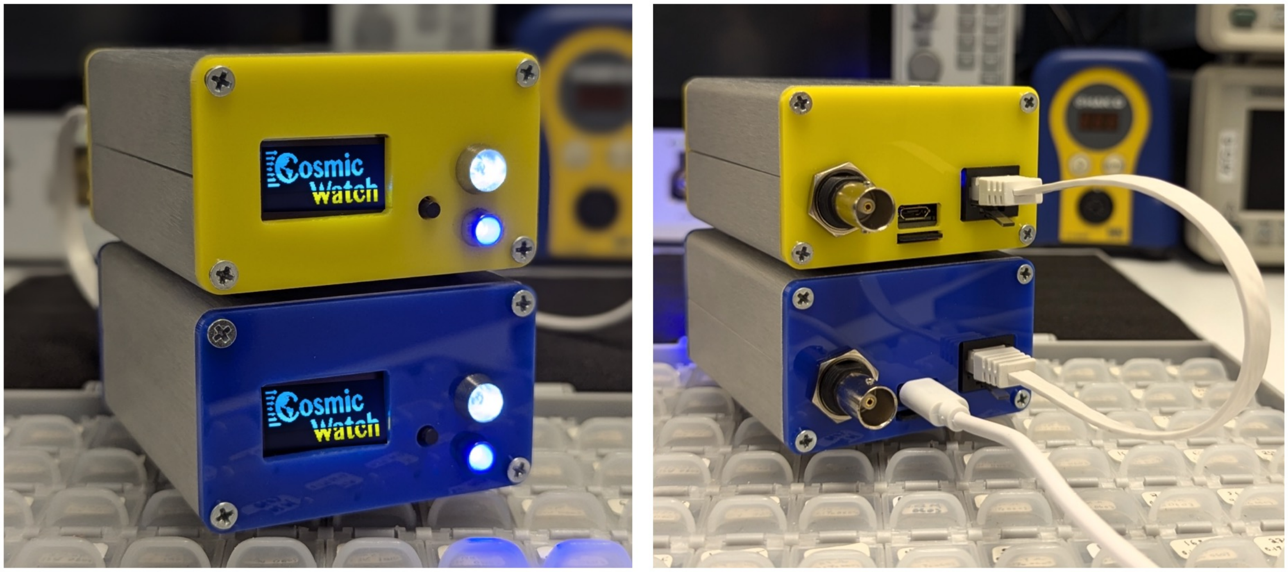


Figure 2.3: Two detectors booting into *coincidence mode.* In coincidence mode, every time a detector triggers on an event, it sends a digital pulse to the other detector. If the other detector also triggers on a pulse during at the same time, the event is flagged as coincident and is very likely to have been caused by a cosmic-ray muon. Coincident events flash the coincidence LED (blue LED on the front plate).

to trigger two detectors. This can be substantially attenuated by going inside a building, a floor (15 cm) or two of concrete does a pretty good job at significantly reducing this flux.

- Gamma rays can penetrate the aluminum enclosure and plastic scintillator, however they have a significant chance of Compton scattering (the dominant interaction), which will change the direction. If a gamma ray does interact with both detectors, this means that it likely Compton scattered off the scintillator slabs, lost sufficient energy to trigger the detector, then Compton scattered or photoelectrically absorbed in the second detector, also depositing sufficient energy to trigger that detector. This process will be rare, and represents a small part of the coincidence signal. It can further be reduced by looking at the SiPM pulse height, as the gamma-induced coincident events will be dimmer on average.

- Accidental coincidences, defined as two uncorrelated random events happening to hit both detectors within some time window, is also a rather rare occurrence at the standard background rates. This will be elaborated on in the Examples Measurement Document. The accidental coincidence rate scales linearly with both master count rates. As a rough figure, assuming both detectors have a 3 Hz trigger rate, the accidental coincidence rate is 54 μHz.

- A typical minimum ionizing muon passing through the 1 cm slab of scintillator will typically deposit more than 1.5-2 MeV of energy in the scintillator, without being deflected. This is a rather large amount of energy compared to the common energies of radiogenic backgrounds (sub MeV-scale). If the muon passes through both scintillators, it will likely trigger both

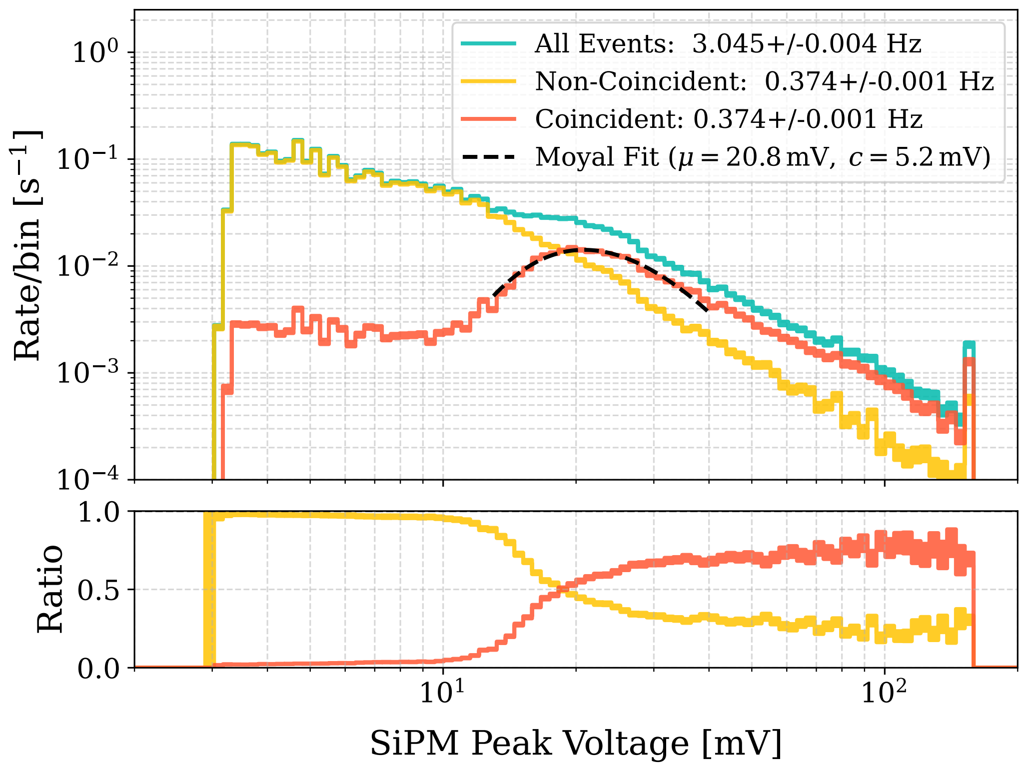


Figure 2.4: A histogram of the SiPM pulse height, of the full dataset from a single detector. The bump in the coincidence events, shown at roughly 20 mV, would correspond to the average minimium ionizing muons, or an energy deposition of 1.5-2Mev.

detectors simultaneously and be relatively bright in both detectors. Therefore, not only can we select these events out using coincidence logic, these events also tend to be brighter on average (more energy deposited in the scintillator means the SiPM pulse height is larger). A histogram of the SiPM pulse height (proportional to how much energy was deposited), is shown in Fig. 2.4, by plotting all events from a microSD card versus the subset of events that were labeled as coincidence ("1" in the coincidence column) and non-coincidence ("0" in the coincidence column). This figure can be recreated by running the plot.py script in the Data/ directory with the example data file in Example_Data/AxLab_C_001.txt. This script requires you to have matplotlib installed. You can change the filename to plot your own data.

2.3 Recording data options

Data can be collected from the Desktop Muon Detector in many ways:

1. **Via the 0.96" OLED screen**: The OLED display is ideal for users who only need overall event counts and average event rates, without detailed per-event data. It reports the total run time, number of triggered events, and the average count rate (accounting for dead time). If the detector boots into coincidence mode, two additional rows indicating the number of coincident events and coincidence average count rate are also reported. Additionally, the OLED display reports the microSD card file name (if present), and the detector name (modifiable in the configure.txt file located on the microSD card). The OLED can also be turned on/off using the configure.txt file.

2. **(Recommended) Through a microSD card** :If a microSD card is inserted, the detector creates a new file each time it is powered on or reset. The microSD card must be formatted as exFAT. MicroSD card filenames begin with the detector name (configurable to fewer than 20 characters via the configure.txt file); followed by a mode indicator: "C" if the detector booted in *coincidence mode*, or "M" if in *master mode*; and a number that increments sequential upwards from the last saved file. Data is stored in plain text format with a .txt extension, as illustrated in the coincident data from a detector named "AxLab" operating in *coincidence mode* in figure 2.6. A relatively small microSD card (≥4 GB) enables users to save several months worth of continuous data at sea level.

3. **Through the import_data.py script on your computer**: When the detector is plugged into a computer USB port, data is transmitted asynchronously at a baud rate of 115200 bps. Running the python script import_data.py (after you've installed the pyserial library), will allow you to save the data directly to a file of your choice. When run, the user is prompted to supply the path and name of the file to where the data is to be stored. It will then begin recording the data in real time to that output file on your computer. The benefit here is that the computer will take a real-time global timestamp from the computer. It's likely that your computer syncs with NTP (Network Time Protocol), which provides an accuracy to within roughly 1–10 ms. The RPi alone, will go out of sync with global time, at the rate of roughly 1 minute per day.

4. **Through a Serial Monitor on your computer**: There are also serial monitors available, that will allow you to see the data in real time. I would say the simplest serial monitor is the ArduinoIDE[1], accessible by clicking on the top right magnifying glass icon, after selecting the RPi in Tools->Ports. However, recent updates to the ArduinoIDE make it difficult to copy and paste the data. Instead, I now prefer to use VSCode, after installing the 'Serial Monitor' extension. The Serial Monitor is very useful when troubleshooting your detector. During bootup, the detector will print to the serial port a set of diagnostics.

5. **The Graphical User Interface**: CosmicWatch GUI 2.5 provides user friendly interface for visualizing real-time data when the detector is connected via serial-port, as well as post-acquisition data uploaded in the form of a .txt file. Users can upload data files from a microSD card to visualize the measured ADC spectrum, coincident and non-coincident event rates, and other relevant metrics. The interface includes binning options, live count

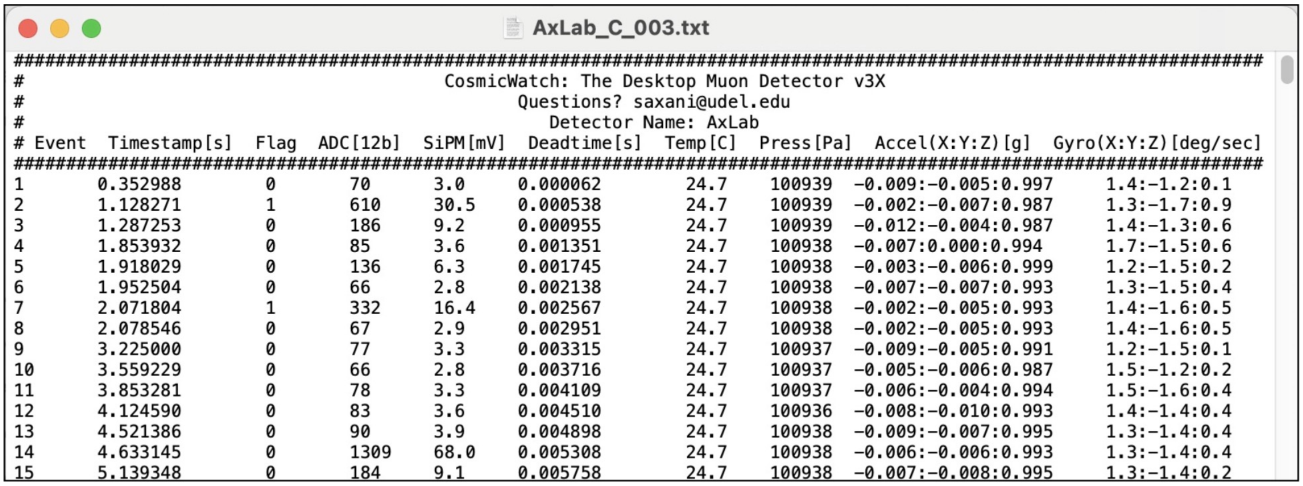
[1] https://www.arduino.cc/en/software

displays, and a modular design with tabs for viewing additional data such as temperature, acceleration, and dead time. The intuitive layout allows users to efficiently explore detector performance and atmospheric particle shower characteristics.

6. **The Website**: We also have a website, that lets' you record data. It also plots the data in realtime.

Figure 2.5: The CosmicWatch Graphical User Insterface.

The data on the microSD card or through the import_data.py script is saved in a simple to use .TXT file, in an easy-to-read format. Comments are always indicated with the "#" marker for ease of parsing. Each row is a different event, and each column represents some different property of the measurement. The columns are tab-delimited, to also make parsing easier. When the detector boots up, it first prints a set of diagnostics, measuring the voltage going to the SiPM, baseline, trigger thresholds and checking with peripherals. I'll expand on these diagnostics in the Troubleshooting Document. An example of the data is shown in Fig. 2.6. Each event is approximately 48 to 100 Bytes in size (depending on if you took the data on the microSD card or with the import_data.py script, respectively).



AxLab_C_003.txt

```
############################################################################################################
#                                      CosmicWatch: The Desktop Muon Detector v3X
#                                           Questions? saxani@udel.edu
#                                              Detector Name: AxLab
# Event  Timestamp[s]  Flag  ADC[12b]  SiPM[mV]  Deadtime[s]  Temp[C]  Press[Pa]  Accel(X:Y:Z)[g]  Gyro(X:Y:Z)[deg/sec]
############################################################################################################
1       0.352988       0     70       3.0      0.000062       24.7     100939   -0.009:-0.005:0.997      1.4:-1.2:0.1
2       1.128271       1     610      30.5     0.000538       24.7     100939   -0.002:-0.007:0.987      1.3:-1.7:0.9
3       1.287253       0     186      9.2      0.000955       24.7     100939   -0.012:-0.004:0.987      1.4:-1.3:0.6
4       1.853932       0     85       3.6      0.001351       24.7     100938   -0.007:0.000:0.994       1.7:-1.5:0.6
5       1.918029       0     136      6.3      0.001745       24.7     100938   -0.003:-0.006:0.999      1.2:-1.5:0.2
6       1.952504       0     66       2.8      0.002138       24.7     100938   -0.007:-0.007:0.993      1.3:-1.5:0.4
7       2.071804       1     332      16.4     0.002567       24.7     100938   -0.002:-0.005:0.993      1.4:-1.6:0.5
8       2.078546       0     67       2.9      0.002951       24.7     100938   -0.002:-0.005:0.993      1.4:-1.6:0.5
9       3.225000       0     77       3.3      0.003315       24.7     100937   -0.009:-0.005:0.991      1.2:-1.5:0.1
10      3.559229       0     66       2.8      0.003716       24.7     100937   -0.005:-0.006:0.987      1.5:-1.2:0.2
11      3.853281       0     78       3.3      0.004109       24.7     100937   -0.006:-0.004:0.994      1.5:-1.6:0.4
12      4.124590       0     83       3.6      0.004510       24.7     100936   -0.008:-0.010:0.993      1.4:-1.4:0.4
13      4.521386       0     90       3.9      0.004898       24.7     100938   -0.009:-0.007:0.995      1.3:-1.4:0.4
14      4.633145       0     1309     68.0     0.005308       24.7     100938   -0.006:-0.006:0.993      1.3:-1.4:0.4
15      5.139348       0     184      9.1      0.005758       24.7     100938   -0.007:-0.008:0.995      1.3:-1.4:0.2
```

Figure 2.6: Example data from a file AxLab_C_002.txt microSD card. This was the second file ("002") recorded in coincidence mode ("C") on the the detector named "AxLab." The definitions of the columns are listed in the header. Event 3 and 11 were identified as coincident events. Looks like the temperature where the data was taken was 23.8°C with a room pressure of 102 kPa. The last column shows the orientation of the detector. Looks like the z-direction linear acceleration was roughly 1.1 g, indicating that the detector was positioned upright.

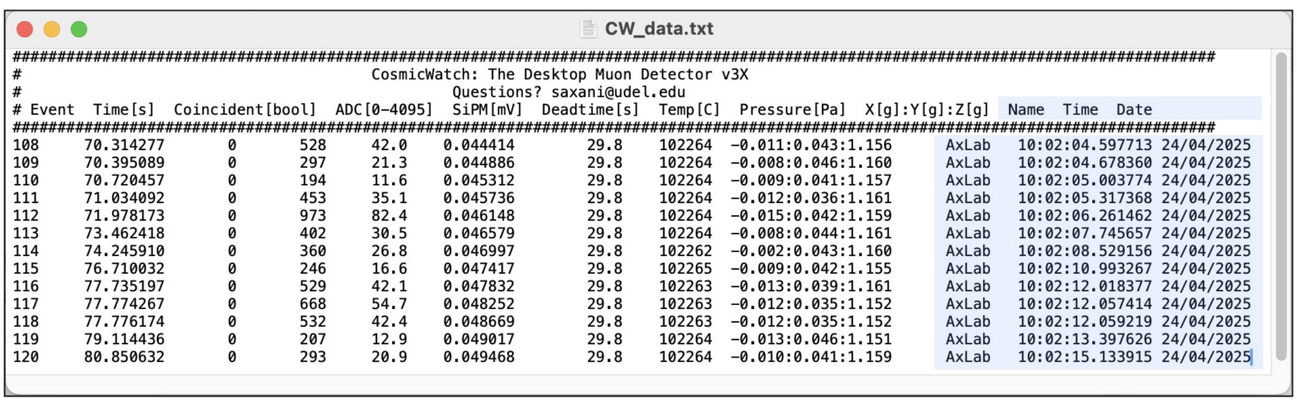
The data shown in Fig. 2.6 is formatted into tab-delimited columns, each of which is labeled in the header and defined as follows:

- **Event**: The event number. Sequentially counts upwards on each trigger.
- **Timestamp [s]**: The total run time (not live time), measured in seconds. This is only accurate to roughly tens of seconds per day given the uncertainty by the precision of the RPi internal clock.
- **Flag**: *Coincidence mode* flag. If the detector observes a coincidence signal while operating in *coincidence mode*, the event is flagged with a boolean "1." Non-coincident events are flagged with a "0."
- **ADC [12b]**: The 12-bit (0-4095) ADC measurement of the buffered peak detector output. The ADC is referenced to 2.5 V.
- **SiPM [mV]**: The calculated SiPM pulse height is based on the ADC measurement. This value is proportional to the energy deposited in the scintillator (see Sec. **??**).
- **Dead time [s]**: The cumulative dead time since the detector start. Dead time should be accounted for when making any rate measurement. To calculate the event rate, divide the total counts by the difference between the total run time and the total dead time. The denominator is what is called live time, and represents the amount of time the detector was actively searching for events.

- **Temp. [°C]**: The local temperature, in degrees Celcius, measured by the BMP280 sensor.
- **Press. [Pa]**: The barometric pressure, in pascals, measured by the BMP280 sensor.
- **Accel. (X:Y:Z) [g]**: The x-y-z acceleration measured in units of gravitational acceleration g ($g \approx 9.81\text{m/s}^2$).
- **Gyro. (X:Y:Z) [deg/s]**: The x-y-z angular acceleration in degrees per second.

If you record the data through the import_data.py script located in the Data directory, it will add three additional columns. You will also notice that the data taken prior to running the import_data.py script is not saved in the file. Example data from this script can be found in Fig. 2.7. The additional three columns of data are:

- **Name**: This is the name of the detector. You can modify this in the config file of the microSD card. We print the name, since you can actually record from multiple detectors simultaneously through the import_data.py script into a single file. This allows you to determine which detector saw which event.
- **Time**: The global time from the computer.
- **Date**: The global date from the computer.



CW_data.txt

```
##########################################################################################################################
#                                        CosmicWatch: The Desktop Muon Detector v3X
#                                                 Questions? saxani@udel.edu
# Event  Time[s]  Coincident[bool]  ADC[0-4095]  SiPM[mV]  Deadtime[s]  Temp[C]  Pressure[Pa]  X[g]:Y[g]:Z[g]  Name  Time  Date
##########################################################################################################################
108     70.314277       0       528     42.0    0.044414        29.8    102264  -0.011:0.043:1.156      AxLab   10:02:04.597713 24/04/2025
109     70.395089       0       297     21.3    0.044886        29.8    102264  -0.008:0.046:1.160      AxLab   10:02:04.678360 24/04/2025
110     70.720457       0       194     11.6    0.045312        29.8    102264  -0.009:0.041:1.157      AxLab   10:02:05.003774 24/04/2025
111     71.034092       0       453     35.1    0.045736        29.8    102264  -0.012:0.036:1.161      AxLab   10:02:05.317368 24/04/2025
112     71.978173       0       973     82.4    0.046148        29.8    102264  -0.015:0.042:1.159      AxLab   10:02:06.261462 24/04/2025
113     73.462418       0       402     30.5    0.046579        29.8    102264  -0.008:0.044:1.161      AxLab   10:02:07.745657 24/04/2025
114     74.245910       0       360     26.8    0.046997        29.8    102262  -0.002:0.043:1.160      AxLab   10:02:08.529156 24/04/2025
115     76.710032       0       246     16.6    0.047417        29.8    102265  -0.009:0.042:1.155      AxLab   10:02:10.993267 24/04/2025
116     77.735197       0       529     42.1    0.047832        29.8    102263  -0.013:0.039:1.161      AxLab   10:02:12.018377 24/04/2025
117     77.774267       0       668     54.7    0.048252        29.8    102263  -0.012:0.035:1.152      AxLab   10:02:12.057414 24/04/2025
118     77.776174       0       532     42.4    0.048669        29.8    102263  -0.012:0.035:1.152      AxLab   10:02:12.059219 24/04/2025
119     79.114436       0       207     12.9    0.049017        29.8    102264  -0.013:0.046:1.151      AxLab   10:02:13.397626 24/04/2025
120     80.850632       0       293     20.9    0.049468        29.8    102264  -0.010:0.041:1.159      AxLab   10:02:15.133915 24/04/2025
```

Figure 2.7: Example data taken from the import_data.py script. The three additional columns are highlighted in blue. They represent the detector name, the timestamp and data taken from the computer that is running the script.

Chapter 3

How to build a CosmicWatch detector v3X

3.1 Overview

This chapter will go through how to build your own detector from scratch.

The construction of the detector will be divided into four parts. (1) Purchasing the components. (2) Populate the printed circuit boards (PCB). (3) Interface the SiPM with the scintillator. (4) Uploading firmware to the Raspberry Pi Pico micro-controller, testing, and your first measurement.

3.1.1 (1) Purchasing the components

All the components are listed in the purchasing_list.pdf document, located in the PCB directory of the GitHub repository. It includes the number of each component to purchase, along with a cost estimate for single purchases and 10x purchases, and a link to where the item can be purchased. We'll do our best to update this document as parts change name, however, nearly all components can be found through other manufacturers, except likely the SiPM. So, first, use the list to purchase the components.

You'll also have to purchase the printed circuit boards. In the PCB folder, there is the Gerber.zip file which can be sent to a PCB manufacturer to produce the boards. The boards are two layer, 10cm x 10cm. I like to use elecrow.com. It takes about two weeks to have the boards manufactured and shipped.

Likely the most difficult thing to find is the plastic scintillator. but the purhcasing list links to

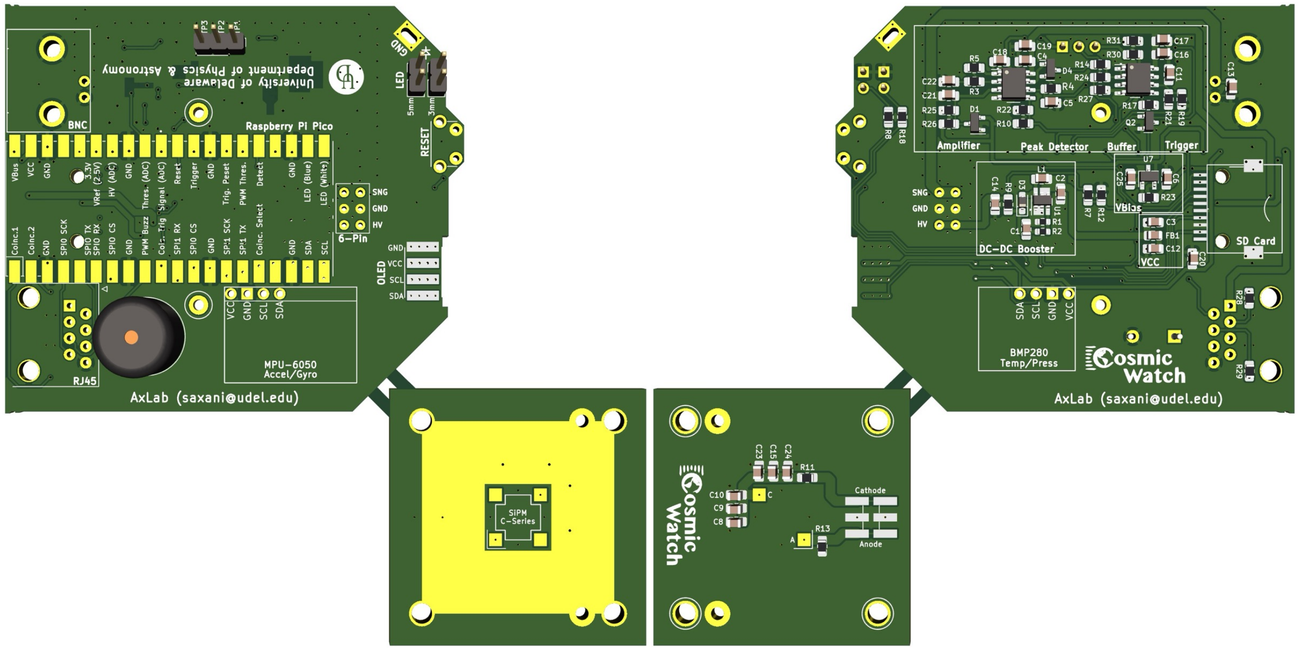


Figure 3.1: A unpopulated rendering of the Main printed circuit boards. Left: Top view. Right: Bottom.

someone on ebay that sells them.

Next, we'll solder the components to the board.

3.1.2 (2) Populating the PCBs

Here is a step-by-step description of how to populate the Main and SiPM PCBs (large and smaller PCB respectively, they simply snap apart). The components are almost all surface mount technology (SMT), which means that they are small and sit on top of the board rather then with legs passing through it (through-hole components). With the exception of two resistors, we will be using 0805 SMT components. This represents the size, where 0805 refers to the physical dimension 0.08 inches by 0.05 inches. The first time you use them, they may look small, but you'll get used to them with a bit of practice. A pro may commonly use 0402 components or even 0201 components. The 0805 SMT components are a good size to learn about soldering. We'll aim to make some Youtube videos going through a full build.

The identifier, reference value, and a description of each component can be found in the component_placement_sheet.pdf document, located in the GitHub PCB directory. This sheet shows you the preferred order in populating the Main and SiPM PCB. It starts with some of the the easier components first for practice, and build the boards in stages which enables testing throughout the build.

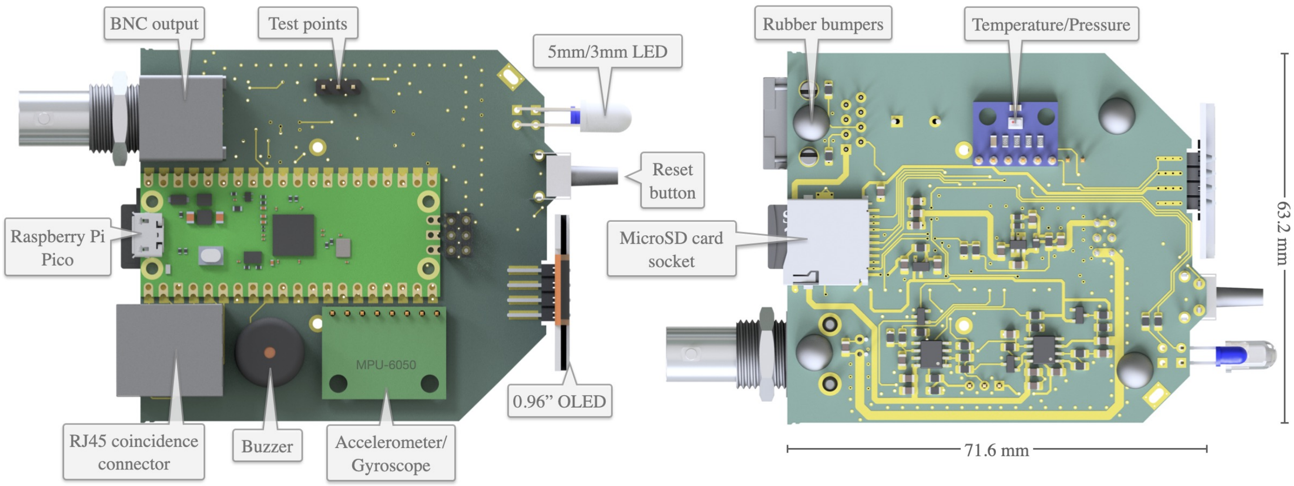


Figure 3.2: A 3D rendering of the Main printed circuit boards. Left: Top view. Right: Bottom.

We use the identifier R to represent resistors, C for capacitors, L for inductors, U for integrated circuits, and D for Diodes. Also note that the larger components, like the RP Pico, temperature/pressure sensor, and OLED, is drawn on the silk screen of the PCB in order to help you orientate the part. **Make sure you put each component on the side of the board with it's silk screen markings.** A rendering of the PCB is shown in Fig. 3.2.

Soldering SMT components to the PCBs

Soldering involves melting a lead-based piece of metal (solder) to for the connection between the component and the PCB. While you can use your preferred soldering equipment, our preference is the **Hakko FX888DX-010BY soldering iron with the HAKKO T18-S4 soldering tip**, at a temperature of approximately 750°F. There exists non-lead based solders, but we've only found them to be terrible. We use **KESTER SOLDER 24-6040-0027 Wire Solder, 0.031" Diameter** solder. This soldering iron is hot, but not hot enough to damage the PCBs or components (except maybe the plastic ones, like the housing of the BNC connector). You will hold the soldering iron with your dominant hand, and the solder with your sub-dominant hand. We call the components position on the PCB the "footprint," and each metal part of the footprint a "pad."

> Tip: It's best to orientate the PCB rather than awkwardly orientate your hands while soldering. Move the PCB into a comfortable position for each solder joint.

Soldering a component in place always involves the same procedure for all components. Practice this routine with each component. The philosophy is to get one pad connected to the component, inspect the orientation, then when satisfied, continue with the other pad(s) and finally go back to the original one to clean it up. That is, the first solder joint is just to hold it in place for subsequent solders. It's easy to move a piece that only has one soldered joint, and very difficult if you have multiple pads constraining the component. Here's a more detailed procedure:

1. Determine the component you are mounting on the board from the component placement sheet. Find that component on the board.

2. Add solder to just one of the footprints pads.

3. Pick the component up with tweezers, bring it close to the footprint. Make sure that the component is flat against the PCB.

4. Melt the solder on the pad from Step 2 and bring your component into place with the tweezers. Keep holding it with the tweezers.

5. When in place, remove the heat, and let the solder solidify. Then, let go with the tweezers.

6. Inspect. Is it flat against the board? Is it in the correct orientation?

7. Finalize. If it looks good, solder the other pads. If it doesn't, re-melt the single solder connection and re-orientate.

8. Perfect. Now that the component is secured to the board, you can go back and add new solder to the joint to ensure a good connection.

9. New solder fixes most issues; if the solder develops a spike, add new solder; if you have a suspect connection (cold joint), add new solder; if need to remove a component, sometimes adding fresh solder helps beleive it or not.

Follow and master the above. By the end, you should be able to add a single component in seconds.

NOTE: Some components have a direction. We've noted it on the reference list. Diodes have directions. The integrated circuits (IC) have direction. All resistors, capacitors, and inductors do not.

You should pretty easily get to Step 5 without to many difficulties. If at any point one of the tests isn't working, it means that you either misplaced a component (unlikely) or you have a poor or missing solder connection on a component (very likely). The first step to troubleshooting will always be to check the solder joints. Add some fresh new solder to joints, to make sure you have good, solid connections.

If you made it past Step 5, you now need to mount the SiPM. Since the SiPM represents half the cost of the detector, we will describe the procedure in verbose below.

Mounting the SiPM

The hardest component to mount is the SiPM. First, you need to make sure you have the SiPM correctly orientated. There is a fiducial marking on the SiPM PCB (white L-shape, lower-left relative to the text on the PCB). This lines up with the Pin 1 of the MicroFC SiPM. Pin 1 is hard to identify, but it is represented by an extra metal leg sticking radially outwards on the side of the SiPM. Have a look at Page 10 of the MicroFC documentation if needed. Here's the procedure for mounting the MicroFC SiPM:

1. Add a small bump of solder to pad one of the SiPM footprint on the SiPM PCB.
2. Pick up the SiPM with your fingers. Avoid using tweezers, as they may scratch the surface.
3. Identify Pin 1 on the SiPM, orientate it with the PCB fiducial marking.
4. Use your fingers to hold the SiPM, perfectly aligned to the silkscreen drawn on the PCB, while you heat the bump of solder. Once it melts, the SiPM should be lying flat agains the SiPM PCB, secured in place once the heat is removed.
5. Inspect orientation. If misaligned heat the pad with the bump of solder and adjust positioning. Else, add solder to the other pads. The solder will flow under the SiPM and connect the legs.

Testing the SiPM connection The SiPM acts as a diode. That is, it allows current to flow in only one direction, and has a voltage drop. Measuring the voltage drop between the Cathode and Anode (positive terminal on the Anode and negative terminal on the Cathode) on the SiPM PCB (annotated on the back of the PCB) tells you if the SiPM is connected. Using the diode setting on a multimeter, you should see a voltage drop of approximately 0.5 V. If you do not see the voltage drop, it could be (1) that you have the component orientated incorrectly (Pin 1 on the SiPM is not on Pad 1 of the footprint), (2) the solder is not fully connecting the SiPM to the board, (3) you have your test leads on the multimeter backwards, swap hands and try again. You want to make sure the SiPM is connected before continuing. If it is not, removing all the tape later takes quite a bit of time.

With the SiPM mounted, Step 6, you are essential done with soldering and you now need to put the pieces you built together. Attach the two standoffs to the SiPM PCB. We will now cut and interface the scintillator.

Cutting the plastic scintillator

You'll need to cut (or purchase) a 5cm×5cm×1cm× piece of scintillator. I cut the scintillator on a Mill, so that you get a nice smooth surface from the end mill. I prefer to use a 3/4" two flute end mill at 1000 rpm, however just about anything will work. Then, after cutting to the correct size, I drill, using a number 48 drill bit (diameter = 1.93 mm) the four holes, in a square pattern with length 30 mm. Finally, I melt (heat treat) the milled opaque surfaces using a hot air gun. This will make a nice transparent surface that will help promote total internal reflection (the more photons you collect, the better your detector can estimate energy deposited). The dimension of the scintillator can be found in the CAD folder of the GitHub repository.

Interfacing the SiPM to the scintillator

Once you have all the components populated, including the SiPM, it's time to interface the SiPM to the scintillator. Make sure you have the aluminum standoffs on the SiPM PCB, securely fastened (past hand tight) with the size 0 screw.

First, wrap the scintillator in a single layer of aluminum foil, leaving a roughly 2x2 cm^2 open area for the SiPM face. Add a small amount of black electrical tape to hold it in place. It's hard to see the four mounting holes with when the scintillator is wrapped in aluminum foil; you can simply apply some pressure to the location of the holes to have their outline pop through the foil/tape. I would use something sharp (liek the tip of the tweezers) to poke holes through the foil/tape, to help align the SiPM board.

Apply a generous amount of optical gel (silicon-based optical PMT coupling gel) to the surface for the SiPM. Drop in the four #2 screws in to the outermost SiPM PCB holes, and screw the board into the scintillator. Remember that the SiPM is roughly 0.65 mm thick, so there should be a gap of roughly that size between the PCB and scintillator. Silicon optical gel can be tough to find, I've used simple silicone pads, which worked alright. Really, you want something with an index of refraction of 1.4 that will break the air barrier between the SiPM and scintillator.

Finally, with the SiPM screwed into the scinitllator, you now need to wrap it in black electrical tape. Remember, we are looking for only a few photons, so we need this thing to be fully optically isollated. Even the smallest light leak will let it millions of photons per second, prohibiting you from seeing the small flashes of light produced by particles running into the scintillator.

Tip: do a good job wraping the scintillator. Cut the corners of the black electrical tape such that they fold over nicely. Best to wrap well once, then try to fix it later.

Uploading the firmware

This is pretty simple. You need to program the micro-controller so that it knows what it's doing. Follow these steps to upload the firmware to the RP Pico. If you want to know how the firmware works, we've decribed it in Sec. B:

1. Hold the boot select button down on the RP Pico while plugging it into a computer. This is a little white button on top of the board.
2. Release the button.
3. The RP Pico will appear as a removable drive on your computer. If it doesn't make sure you read Step 1 correctly. If it still doesn't work, it's likely because your USB cable does not have data-transfer. You might need to find a better cable. Some are just for power.
4. Open the drive.
5. Drag and drop the CosmicWatch_v3X_22.uf2 file from the Firmware directory in the GitHub, into the drive.

If the firmware was uploaded correclty, the RP Pico will disconnect from your computer, and you should see your OLED screen on your detector light up.

If the full construction went according to plan (probably 30% chance that you won't run into problems), you should be able to plug your SiPM and scintillator combo into your Main PCB. When you supply power to the RP Pico, the detector should trigger at roughly 2.5 Hz. The master rate will depend on how well you build the detector. Essentially, the detector triggers on some number of photons, and the number of photons that the SiPM sees will depend on things like the quality of the surface you heat treated on the scintillator, the optical coupling, the scintillator-to-scintillator light yield differences, that quality of your aluminum foil. About 0.4 Hz of this rate are muons; the rest is mainly backgrounds related to environment radioactive decays. If you like, you can insert the detector into the case, add the LED holders to the front face plate, and rubber feet to the bottom of the case. The case is purely cosmetic and protective. If your measurement does not need a case, you can run the detector without it.

The fun begins when you have two detectors built. When booted up together, and plugged in with at RJ45 cable, they will talk to each other. Both detectors will flash a bright LED pulse for 1 seconds, indicating that they see each other and have entered "coincidence mode." In coincidence mode, events that are observed by both detectors (within a three microsecond time window) are tagged as coincident and are overwhelmingly very likely caused by a cosmic ray muon passing through both detectors.

Note: when you supply power to one of the detectors, the RJ45 cable powers the second detector.

If you run into problems see the Troubleshooting Document.

Chapter 4

Conclusion

The CosmicWatch Desktop Muon Detectors offer a versatile platform for exploring diverse natural phenomena. This document provided a comprehensive overview of the underlying physical processes influencing the detectors and demonstrated how valuable insights can be derived from the collected data. The detectors can be employed to investigate a wide range of phenomena related to the geomagnetic field, atmospheric conditions, cosmic-ray shower composition, particle attenuation in matter, radioactivity, and statistical properties of Poisson processes. Students are encouraged to further develop the concepts introduced or design their experiments. Feedback is highly welcomed to enhance and refine this manual. Enjoy your exploration with the CosmicWatch detectors!

Appendix A

Detailed description of the electronics

A.1 The analog circuit

This sub section describes how the analog electronics work. Behind the analog electronics designed to extract information from the SiPM. The amplitude of the SiPM pulse is directly proportional to the energy deposited by the particle that triggered it. Hence, the objective of the analog electronics is to mold the pulse in a way that enables the slow inexpensive micro-controller to measure certain parameters associated with the pulse amplitude.

Fig. A.1 illustrates the four stages of the analog electronics. The SiPM waveform (depicted in red) undergoes an initial amplification of approximately 20x (shown in blue). The amplification signal is then directed into two circuits: a peak detection circuit (green) and a comparator stage which compares the amplified signal to a set trigger threshold voltage level (horizontal dotted line). The peak detector circuit retains the peak value for a sufficient duration, allowing the RP Pico MCU to measure a single value (indicated by the light purple star). The measured ADC value at this point represents the ADC measurement.

Let's break these down.

A.1.1 The amplifier

The SiPM signal is AC coupled to the input of a non-inverting single supply amplifier. After AC coupling with C21, we add a 25 mV voltage bias to the SiPM waveform, to move the waveform away from ground. This is because amplifiers often have trouble amplifying small signals near their power rails. For our amplifier of choice, the TPH2502, the rails are set to V- = GND and V+ = VCC (4.5V). The biased SiPM waveform is sent to the non-inverting input of the first op amp in the TPH2502. The amplification is set through the feedback and gain resistors R5 and

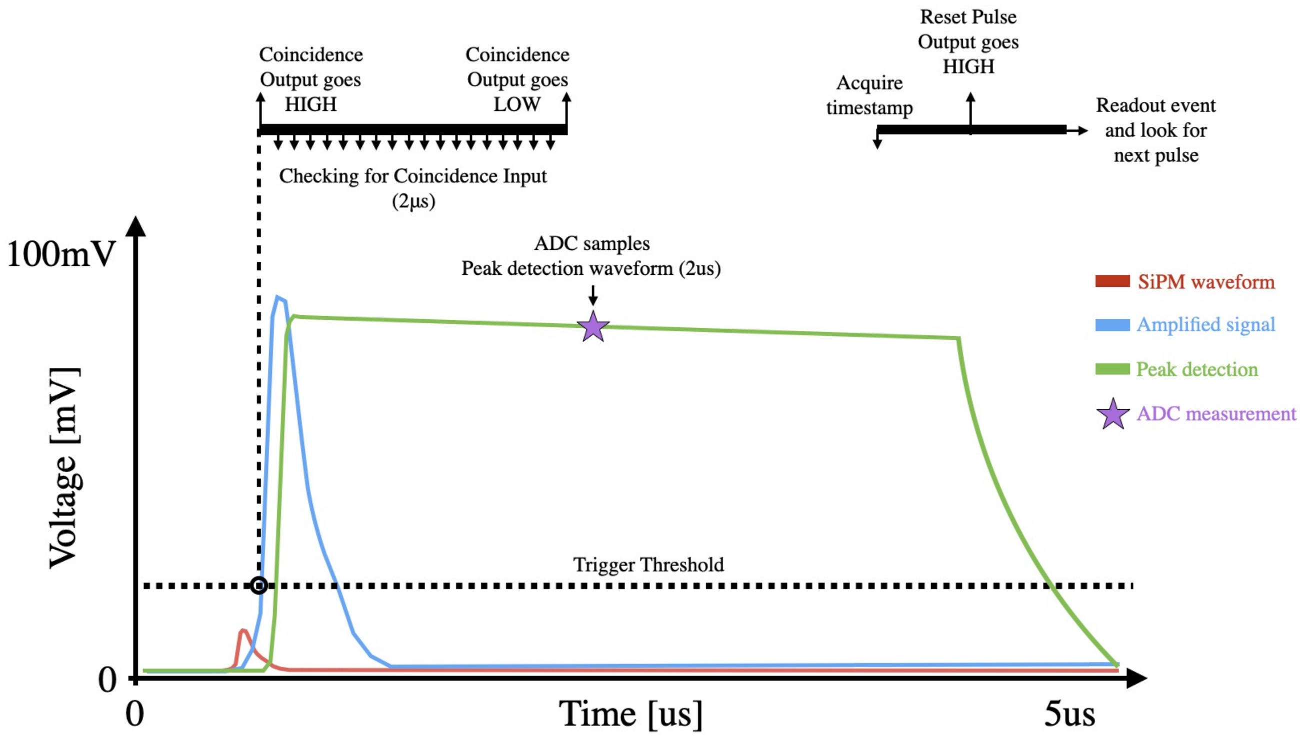


Figure A.1: A description of the analog circuitry and CPU routine. The SiPM pulse (red) is amplified (blue), and peak detected (green). When the amplified pulse crosses the trigger threshold (software defined voltage level), it triggers the detector to 1) check for coicnidence signal from coincident detector, 2) samples the amplitude of the peak detector waveform, and 3) takes a time stamp and reads out the event information.

R3, respectively. The gain of the amplifier, to first order is:

$$\frac{V_{out}}{V_{in}} = 1 + \frac{R_F}{R_G} \tag{A.1}$$

In our case, we get a voltage gain of 13.4. Gain is often represented in decibels, which puts it into a logarithmic scale. With this scale, every +6 dB corresponds to doubling the voltage gain. The conversion is:

$$G[\text{dB}] = 20Log_{10}(\frac{\text{V}_{out}}{\text{V}_{in}}) \tag{A.2}$$

Which means our gain also corresponds to 22.6 dB.

The capacitor C22 acts as a high pass filter, that is, it allows high frequencies to pass through, but low frequencies to be blocked. The cutoff for this is around 3 kHz. The amplified pulse is now AC coupled to the second stage of the dual TPH2502 operational amplifier through C18. This is to remove the DC bias that was amplified (25 mV × 13.4 = 335 mV), as well as the voltage offset introduced through the op amp itself. After removing this offset, we introduce another, well controlled offset, again through the voltage divider setup between R22 and R10, again biasing the waveform by 25 mV. It is this amplified signal that is fed into the peak detector (second stage of the TPH2502 (U2)), and the comparator circuit. You can look at the amplified pulse using the testpoint TP1.

A.1.2 The peak detector

The peak detector aims to hold peak voltage of the amplifier for sufficient time that we can measure it using the relatively slow RP Pico. The non-inverting input to the peak detector is the 25 mV biased amplified pulse. the output is driven through a diode (D4) to charge a capacitor (C5). The TPH2502 is great at this. It can output quite a large amount of current for its physical size. The inverting input is connected to the charging capacitor. After the pulse subsides, the diode D4 becomes back biased and doesn't conduct. Therefore the charging capacitor can't discharge, and it holds it's peak voltage. We then buffer this signal using another TPH2502, that feeds the signal through a voltage divider to match the maximum output from the peak detector given the powering rails, to the ADC dynamic range (0 to 2.5 V). The voltage divider R24 and R27, reduce the peak detector amplitude to about 565% it's original size. The peak detector works quite well, and took quite a bit of tuning, both in terms of finding an adequate op amp, as well as optimizing the components. You can look at the amplified pulse using the test point TP2.

After the detector has made it's measurement of the peak detector voltage, a digital Trigger Reset command shorts the capacitor C5 through a 1 kΩ resistor with a MOSFET transistor (Q2).

A.1.3 The comparator

The amplifier also feeds a comparator circuit, the non-inverting input to the second stage of the TPH2502 (U5). When the amplified signal is larger than the inverting input voltage (the trigger threshold), the op amp goes high and outputs a 4.5V pulse, whose length is as long as the amplified pulse stays above the threshold. The trigger threshold, the voltage at which the detector will trigger it's readout is set using the pulse width modulation (PWM) of the RP Pico. The voltage is set by a 1 MHz PWM frequency, whose duty cycle is tunable in the configuration file. This digital signal is sent through a voltage divider and low pass filter, setup by R19 and R21, and C11. This makes a pretty nice and clean DC voltage that sets the trigger threshold.

A.2 Digital circuit

The digital side of the circuit consists of the RP Pico, the SD card writer, the temperature and pressure sensor, the OLED screen, the buzzer, the coincidence connector, the LEDs, and the accelerometer. Eventually, we'll make an expansion module that can connect up to the coincidence connector to introduce additional peripherals.

A.2.1 Raspberry Pi Pico

The Raspberry Pi Pico is a small, low-cost, high-performance microcontroller board built around the RP2040 chip, which was designed in-house by the Raspberry Pi Foundation. Key features:

- Processor: Dual-core ARM Cortex-M0+ at 133 MHz
- Memory: 264 KB of SRAM and 2 MB of onboard Flash storage
- Connectivity: 26 GPIO pins (General Purpose Input/Output). Supports I^2C, SPI, UART, PWM, and ADC (analog-to-digital conversion).
- Micro-USB port for programming and communication.
- Power: Runs on 1.8–5.5V (very flexible for projects).
- Programming: You can program it in C/C++ or MicroPython easily. MicroPython is too slow for this project.
- Drag-and-drop programming: when you connect it by USB, it appears like a USB drive — you just drag your code onto it.
- Size: About 21mm × 51mm — tiny, like a stick of gum!

A.2.2 SD card writer

The SD card uses the SPI communication protocol on SPI1 of the RP Pico.

A.2.3 Temperature and pressure sensor

The BMP280 is a high-precision, low-power digital barometric pressure and temperature sensor developed by Bosch Sensortec. It is widely used in applications such as weather monitoring, altitude measurement, indoor navigation, and IoT devices. We talk to it through i2C protocal. Below is a detailed description of its key features, specifications, and functionality:

- I^2C address: 0x76 or 0x77 (depending on the SDO pin state).

- Compact Size: Packaged in a small LGA (Land Grid Array) footprint: 2.0 mm x 2.5 mm x 0.95 mm.
- Low Power Consumption: ~2.7 µA at 1 Hz sampling rate.
- Pressure Resolution: 0.16 Pa
- Temperature Resolution: 0.01°C
- Absolute Accuracy: Pressure: ±10 hPa (typical). Temperature: ±3°C (typical).

Appendix B

The Firmware

At the top of Fig. A.1, an approximation of the routine executed by the CPU is presented. When the amplified signal triggers the detector readout, we immediately send out the coincidence output signal and search for a coincidence input signal. We ask whether the coincidence input is HIGH 32 times (each ask takes 72 ns) and if any of them are HIGH, the event is flagged as a coincident event. After determining if the event was coincident, the ADC samples the peak detector waveform. A single ADC sample takes approximately 2μs. We then set the coincidence output to LOW, acquire the timestamp of the event, and reset the peak detector, and populates the variables that hold the event information. Finally, we print the event information to the serial port and flash the LEDs appropriately. The above all happens on the first core of the RP Pico.

The second core run continuously in a while loop. The loop sequentially updates the OLED screen, then checks the microSD card buffer and determines if it's time to record the batched data, and finally updates the environemnetal sensors.

Fig. B.1 displays the printed circuit boards (PCBs) utilized in the v3 detector. The left (right) side showcases the bottom (top) of the PCBs. The PCB comprises two distinct pieces: the SiPM PCB and the Main PCB. Each component of the circuitry is appropriately labeled. The blue box denotes the analog circuitry, as described previously, while the purple box represents power handling. This includes a DC-DC booster that increases the 5V voltage from the USB connection to 30V and a 2.5V regulator that references the ADC and adds some biases to variosus parts of the analog circuitry.

The last figure in this section, Fig. B.2, presents a vector image of the complete circuit. If you have any questions, please feel free to reach out to Spencer at saxani@udel.edu, as we will forgo a detailed description here.

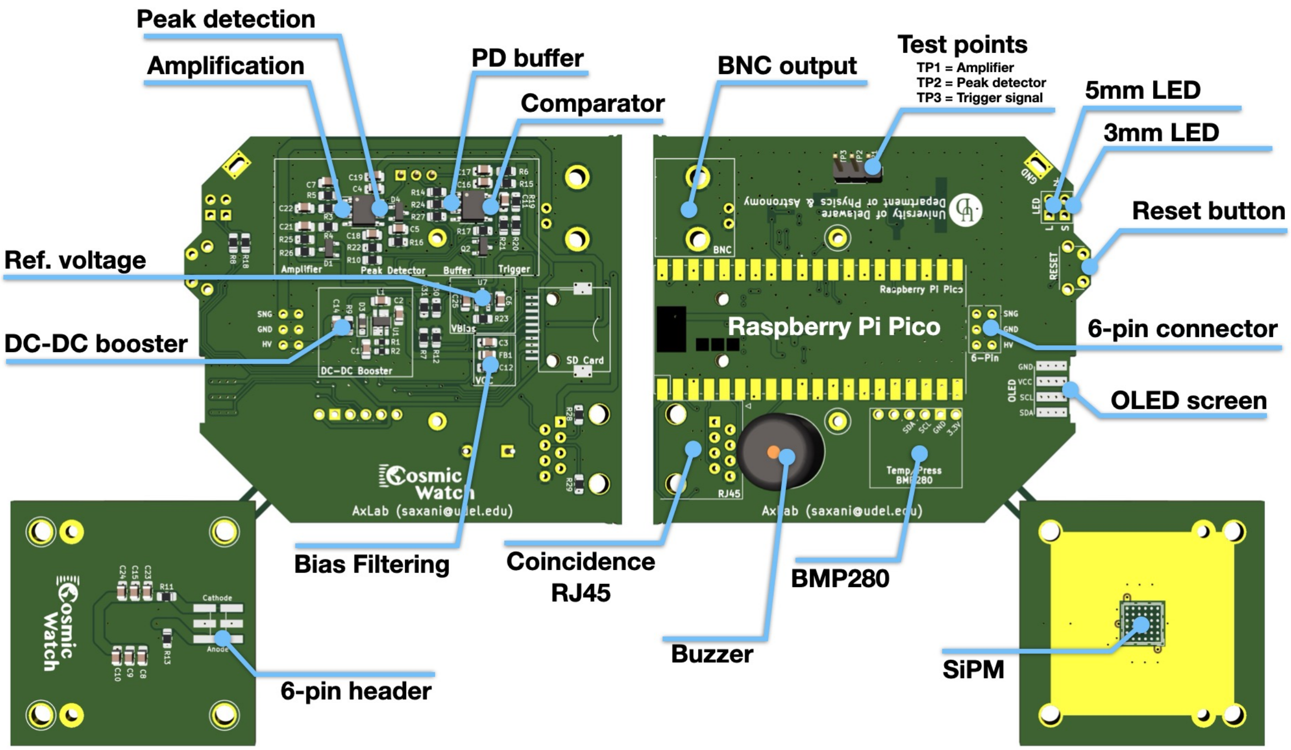

Peak detection
Amplification
PD buffer
Comparator
BNC output
Test points
TP1 = Amplifier
TP2 = Peak detector
TP3 = Trigger signal
5mm LED
3mm LED
Reset button
Ref. voltage
DC-DC booster
Raspberry Pi Pico
6-pin connector
OLED screen
Bias Filtering
Coincidence
RJ45
BMP280
Buzzer
SiPM
6-pin header

Figure B.1

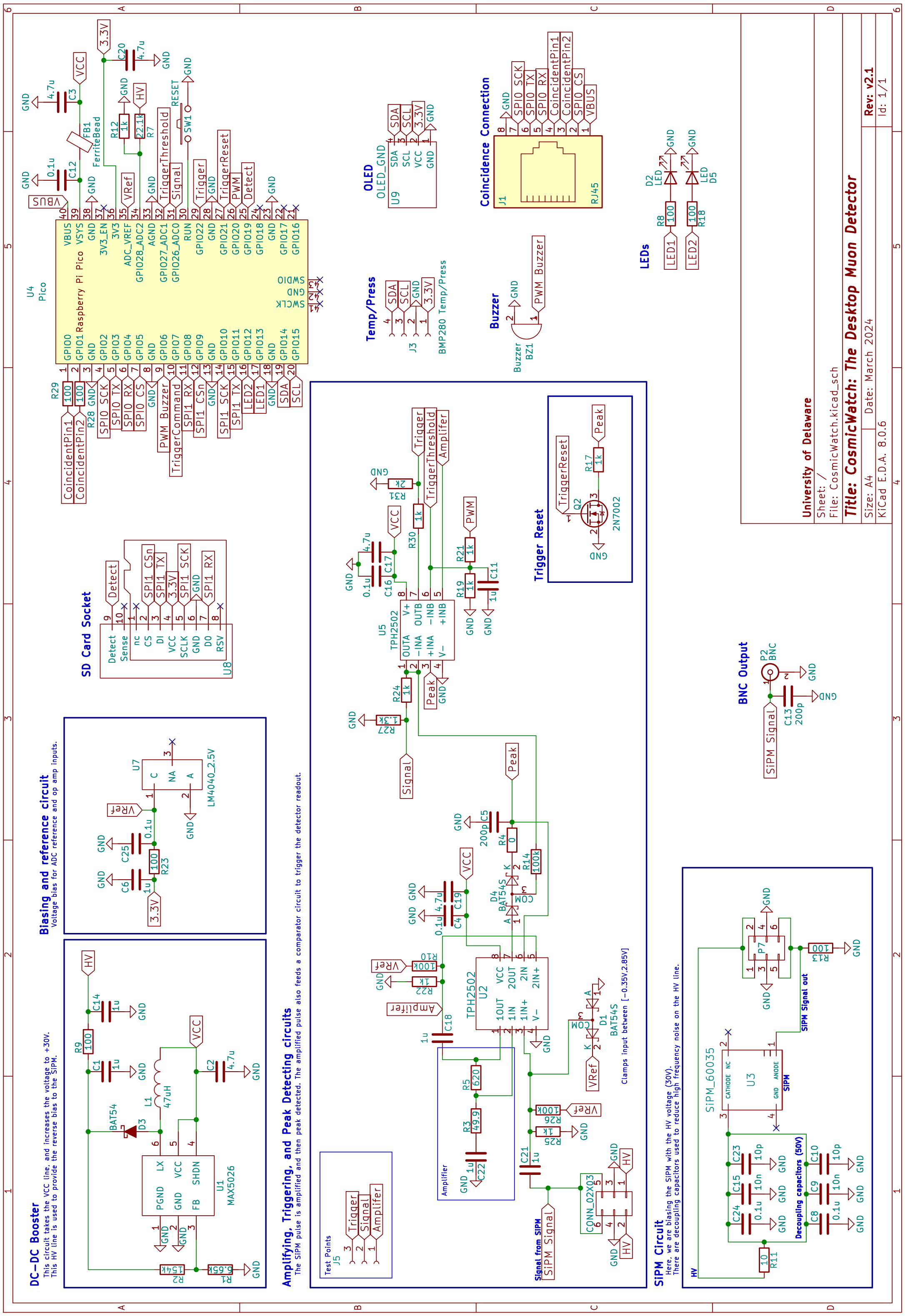


Figure B.2: The circuit diagram of the v3X Desktop Muon Detector. Vectorized version available on GitHub.

Bibliography