**Final Report - Muon Detector Upgrade**

**Conrad Newfield with assistance from Siqi Wang**

**1 Introduction**

The purpose of this project was to upgrade the muon detector by using a new detection system to increase the efficiency of the detector. The previous two versions used a CMOS within a webcam. This version uses a plastic scintillator and a silicon photomultiplier (SiPM). This upgrade increased the efficiency by several orders of magnitude.

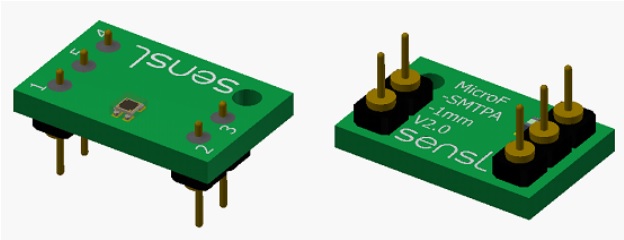
**2 Detection System**

There are two types of scintillators: organic and inorganic. Inorganic scintillators are often more expensive, have lower rapidity, and can be hard to shape. But they often emit more light than organic ones. Organic (plastic) scintillators respond much quicker, are cheaper, and can be easily molded to the desired shape.

This project uses an SiPM to detect the photons generated by a plastic scintillator. When a muon (or any charged particle) passes through the scintillator, it deposits some of its kinetic energy. The scintillator is doped with fluorescing molecules that excite due to the energy from the muon. Within nano-seconds, these molecules de-excite, producing light, often within the visible range. The numbers of photons produced per particle depends both on the energy and path of travel.

There are only a couple dozen photons produced per muon when using a 1cm cubic scintillator. Historically, photomultiplier tubes were used to generate more photons, enough to be detected. But these are bulky and often expensive. So, for this project, an SiPM is used. It consists of many microcells, each having a silicon P-N junction. A depletion region exists between the P and N sides where electrons and holes annihilate. When a photon enters the depletion region, it can deposit enough energy to move an electron. Biasing the P-N junction creates a wider depletion region. With a large enough voltage, one single photon will trigger an avalanche of electrons, generating a small but measurable current.

The SiPM comes with two issues: dark count and crosstalk. Dark count occurs when an electron attains enough energy from thermal variation to trigger an electron avalanche. An avalanche caused by a thermal electron and a photon or indistinguishable. The dark count rate varies per SiPM but can be found and calibrated for by using the manufacture’s documentation. Crosstalk occurs when one microcell’s activation causes an adjacent cell to trigger. This causes a second, slightly delayed avalanche. Both issues can be considered and calibrated for by using the SiPM’s datasheet.

 The SiPM used for this detector is the MicroFC-SMTPA-10035 (cf. figure 1). It has an active area of 1mm­­­2.

**3 Signal Processing**

Figure . MicroFC-SMTPA-10035

The signal from the SiPM is around 100-200µs. It has a rise time of about 5 µs. The first step of processing the signal was simply getting it into the Raspberry Pi (Raspi). The Raspi has only digital inputs but the signal from the SiPM is analog. Therefore, an Analog-to-Digital Converter (ADC) is used. The MCP-3008 is an ADC from Adafru.it that has a max sampling rate of 200ksps (kilosamples per second) at 5V. But drawing 5V from the Raspi is dangerous because voltages over 3.3V can damage the system. Therefore, the 3.3V pin was used meaning the max theoretical sample rate dropped to around 100ksps. But due to Linux and other inefficiencies, the max sampling rate seemed to be around 70ksps based on online forums from users of the MCP-3008. Using code from this [site](http://www.jumpnowtek.com/rpi/Analyzing-raspberry-pi-spi-performance.html) called “mcp3008-poll” designed to test the sampling rate of the ADC, we achieved a rate of 32.5ksps. The problem with this code was it didn’t save the measurements from the ADC.

The code from Adafru.it to attain measurements from the ADC is written in python. This code gives a sampling rate of about 1.6ksps. Due to the faster nature of C, and that the code producing the 32.5ksps was written in C, C was chosen to attain and analyze samples from the ADC.

After much trial and error, Siqi created a script in C that gives around 60ksps. This code collects samples for however many minutes the user chooses, and then analyzes the collected data afterward. Doing data analysis during the measuring would reduce the sampling rate but also allows for data collection during the entire time code is run.

The noise from the SiPM is significant and often rises to the voltage peak of the signal. The differentiating feature that allows for muon detection is the rise and fall time of the signal. The noise is changes slowly over time with occasional extremely fast voltage spikes. Our data analysis takes the difference between every voltage and the previous value, creating an array with one less value than the voltage array. If the voltage difference is greater than a certain value and the next two differences are small decreases, it’s very likely to be a muon.

**4 Wiring**

The ADC wiring was taken from this [tutorial](https://learn.adafruit.com/raspberry-pi-analog-to-digital-converters/mcp3008). This project uses the hardware wiring since the software wiring only works with the code from Adafruit. The SiPM was a combination of the recommended bias voltage filter from the SensL documentation. The two capacitors and two resistors form a second order low pass circuit to stabilize the input voltage. Pins 1 and 3 of the SiPM are the anode and cathode respectively. Pins 4 and 5 are always connected to ground. The output from the SiPM is wired to channel 0 of the SiPM. This can be changed but then the code must be altered slightly.

**5 Code**

For the Raspi to interact with the ADC, the SPI driver must be enabled. This is the reason for uncommenting “#dtparam=spi=on” in the tutorial. The specific type of ADC must be entered into the boot file for the Raspi to be able to receive data. This is why “dtoverlay=mcp3008:spi0-0-present” is added after enabling the SPI.

There were memory issues (segmentation faults) throughout the code-writing process. One issue seemed to be the amount of the memory the allowing our code to access. This memory was expanded by using “ulimit -s 100000000000”.

**6 Future Developments**

There are several significant improvements for this project that are feasible for a future intern.

**6.1** **Powering the SiPM**

Currently we are using a DC power supply for the voltage bias. This is bulky and expensive. The best upgrade would likely be a PCB that rectifies and reduces the voltage from an outlet.

**6.2 Larger Scintillator**

The group has a larger (~15cm x 15cm x 2cm) scintillator slab with an optical fiber running through it. This optical fiber has two outputs. The larger scintillator would detect approximately 225 times as many muons as the current scintillator over a given period. The two outputs of the optical fiber could be used to compare the signals, filter noise and give coincidence.

**6.3 Reincorporate GPS and Other Sensors**

The code from the previous two versions of this project was written in Python. It must be converted to C and modified to run the SiPM along with the other sensors (GPS, temperature probe, pressure probe, etc.).

**6.4 Calibration**

The code currently only works best when the voltage is set to 28V. The code should be altered to fit with any voltage used in the range of the SiPM (~24.7V to 29.5V). Additionally, the code should be continually calibrated to eliminate false reading and match the theoretical muon flux. This can be done by running an oscilloscope in parallel with the ADC. With the correct threshold (level) on the oscilloscope, it’s possible to simply count the numbers of events detected by the SiPM.

**6.5 Dark Box**

Since middle schools don’t often have oscilloscopes, it is difficult to check if the SiPM and scintillator are properly wrapped together. To solve this problem, a box (plastic or cardboard) should be made to house the Raspi, breadboard and SiPM. It should only have holes for the wires that are connected externally.

**6.6 Elimination of Breadboard**

Breadboards are bulky and are more prone to malfunctioning than a PCB. The aforementioned PCB should include the ADC and SiPM wiring.