

Engineering 11

Spring 2025

Final Project Proposal

Radiation Mapping

Kevin Bui, Mihir Gaudani, Elizabeth Perez-Lopez

Mapping and identifying sources of background radiation around campus.

I. Question

Which areas of campus have higher/lower background levels, and which sources contribute to those levels?

II. Methodology

- Map out radiation levels in Etcheverry Hall, Social Sciences Building, Dwinelle Hall, and Pimental Hall.
- Determine potential causes for any observed significant variation.
- Map radiation levels somewhere else in the local area that is of interest.

III. Materials

- a. Raspberry Pi
- b. Breadboard
- c. Pocket Geiger sensor
- d. Jumper Wires
 - i. Male-to-Male
 - ii. Female-to-Female

IV. Steps for Radiation Sensor Set-Up

1. Turn on the Raspberry Pi (Rpi)
2. Connect one of the 5V pins on the Rpi to the + column on the breadboard
3. Connect one of the ground pins on the Rpi to the - column on the breadboard
4. Run a connector from the + column to one row on the main part of the breadboard (referred to as +5V row in future instructions)
5. Run a connector from the - column to a different row on the main part of the breadboard (referred to this row as the GND row in future instructions)
6. Connect the Vin from the Pocket Geiger sensor to row with the +5V row
7. Connect the GND from the Pocket Geiger to the row with the GND row
8. Connect the SIG pin on the Pocket Geiger to another row on the breadboard
 - a. Run a 10k Ω resistor from the SIG row to the +5V row on the breadboard
 - i. This is the “pull-up” resistor to help shape the signal pulse for detecting radiation

9. Connect a male-to-male connecting wire to the SIG row and the GND row, and connect the main part of the oscilloscope input (or the red banana clip) to the SIG wire and the ground clip (or black banana clip) to the GND wire
10. You should now see a voltage offset on the oscilloscope (push the auto-set button if you don't see any channel 1 voltage reading)
 - a. Note that you can use various dials on the oscilloscope to change the voltage step size or the time step size (vertical and horizontal controls, respectively)
11. You should adjust the trigger position –the arrow at the right of the screen pointing towards the signal line – you can control it with the “trigger” dial at the right of the dial control area
 - a. Adjust the trigger position so that the trigger is below the voltage output level you see from the device. You should start to see pulses triggered on

V. Steps for Data Collection

1. Connect your Raspberry pi to the display/keyboard/mouse (you will need to power cycle the Rpi for the display to work)
2. Connect the SIG from the radiation sensor to any open GPIO pin on the Rpi (so not one of the pins that is pre-set for a specific kind of signal input) – consult an online pin map
3. Start making a Python script to get data from this radiation sensor
 - a. Edit on your laptop and pull changes to your Raspberry Pi using git
 - b. You can get to your git repository on your Raspberry Pi using the terminal/command line with:
 - i. `cd repo-name/`
4. You will be using this Python package to setup an event trigger for detecting a pulse using the GPIO pin on the Raspberry Pi:
 - a. <https://sourceforge.net/p/raspberry-gpio-python/wiki/Inputs/>
 - b. This example on how you might write your script:
<https://grantwinney.com/using-pullup-and-pulldown-resistors-on-the-raspberry-pi/>
5. Following the example linked to above, write a script that prints out the time-stamp each time a count is detected by your sensor
 - a. NOTE: you will want to use a falling-edge event detection
 - b. NOTE: Call-back methods are functions that only run when some external property changes, in this case, the change in voltage on the GPIO pin
6. Update this script to also keep track of the number of counts detected and have it print out the number of counts collected each minute
 - a. NOTE: as a suggestion – use an infinite while loop as you have before, but sleep for a full minute

- i. You will need to have a global variable that is the counts recorded and you will need to reset this in your while loop
7. Extra: Modify your script to match your other scripts
 - a. Run for a fixed amount of time
 - b. Save counts recorded (each minute) and time-stamps to a file
 - c. Use input arguments to set a run time and output file name

VI. Steps

1. Once completed the above sections, you are ready to collect data from chosen Sites