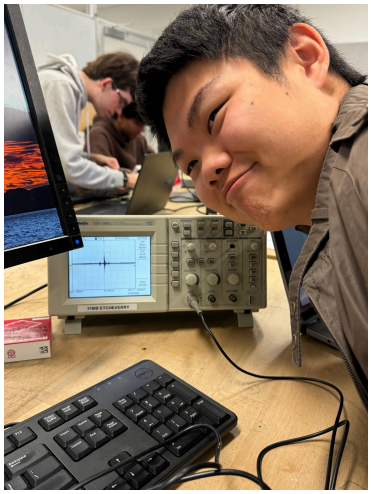
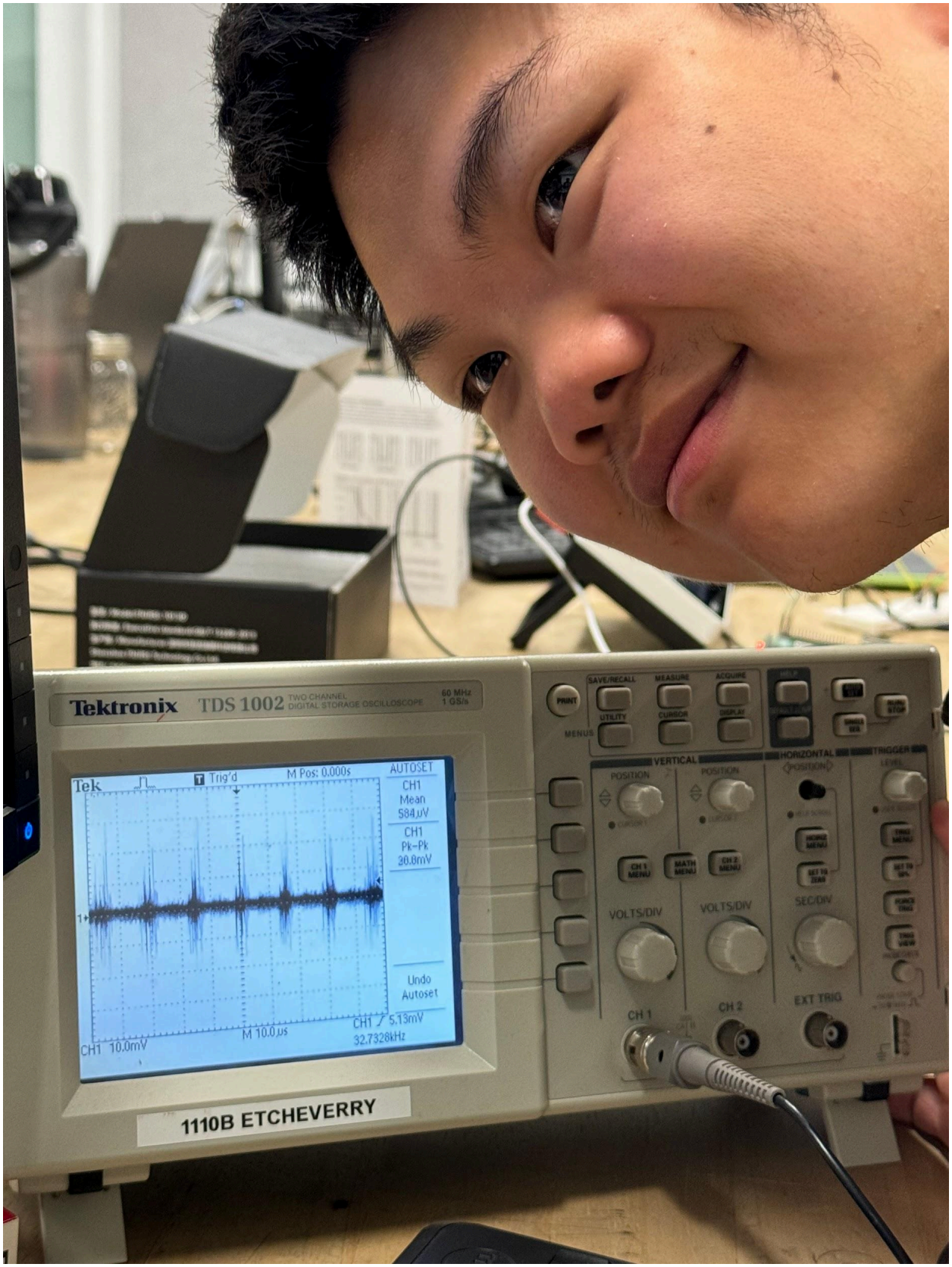
Lab Activity Week 9 – Radiation Sensors

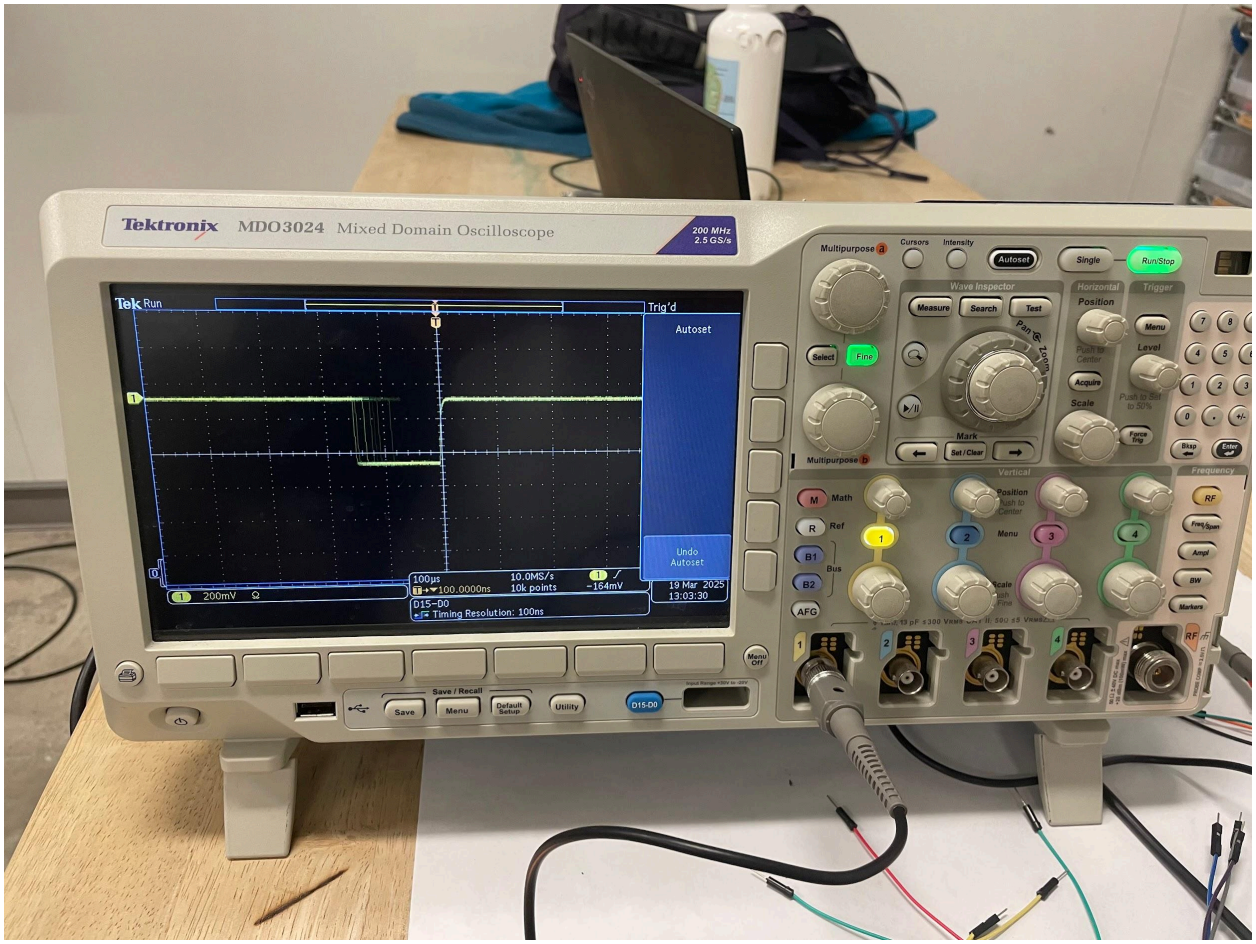
Take pictures of your oscilloscope output and include those images in your lab submission. Turn in any work by sharing your google doc with me or uploading to your GitHub repository.

Radiation sensor in a circuit:

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
   a. I will refer to this row as the +5V row in future instructions
5. Run a connector from the - column to a different row on the main part of the breadboard
   a. I will refer 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 10kΩ 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
   a. The oscilloscope cable should be going to channel 1, or you can connect it to channel 1. Turn the oscilloscope on if it is not already on.
10. You should now see a voltage offset on the oscillascope (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 – arrow at right of 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 devise. You should start to see pulses triggered on.
    b. Try bringing a source of radiation near the device and see what happens to the signal.
12. Take a screen shot or picture of your oscilloscope output




Tektronix TDS 1002 TWO CHANNEL DIGITAL STORAGE OSCILLOSCOPE
60 MHz 1 GS/s
PRINT
SAVE/RECALL
MEASURE
ACQUIRE
UTILITY
CURSOR
DISPLAY
MENUS
VERTICAL
HORIZONTAL
TRIGGER
POSITION
POSITION
VOLTS/DIV
VOLTS/DIV
SEC/DIV
CH 1
CH 2
EXT TRIG
Tek
Trig'd
M Pos: 0.000s
AUTOSET
CH1
Mean
584 µV
CH1
Pk-Pk
30.8mV
Undo
Autoset
CH1 10.0mV
M 10.0 µs
CH1 5.13mV
32.7328kHz
1110B ETCHEVERRY



13. Comment on what you are seeing as pulses – are they what you expected? Why do you think the pulse has the shape that it does?

The pulse mimics a graph of a heart rate with repetitions. Which make more sense if has a signal source that produces constant partical.gy

Radiation sensor data acquisition:
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. As with previous scripts, do your editing on your laptop and pull changes to your raspberry pi using git
b. Recall that 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 provides a bit more detail 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 us 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