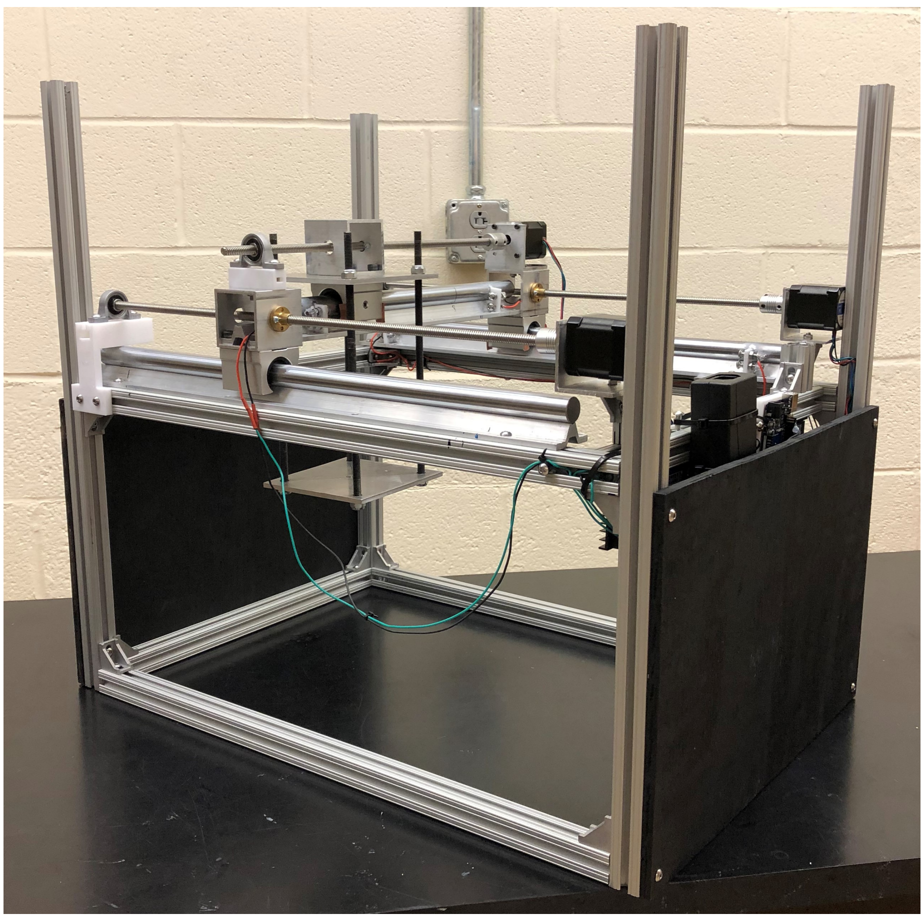
Radiation Detector Spatial Calibration System (RaDSCS) User Manual

Zachary Hainsel
Elizabeth Davis
Clayton Kirberger

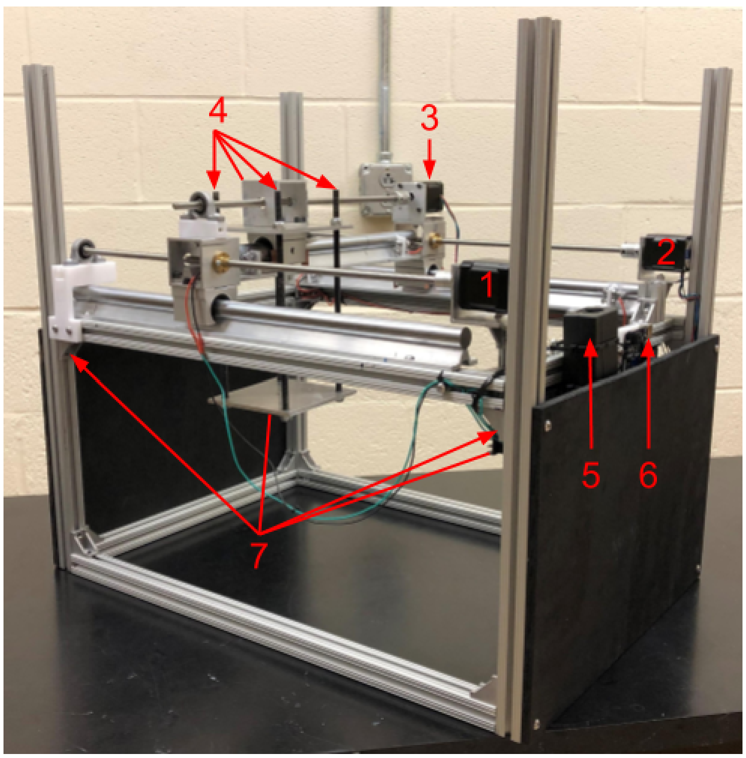
December 2019



1 Introduction

The Radiation Detector Spatial Calibration System (RaDSCS) is a device for the spatial characterization of radiation detectors. This device allows the user to scan and characterize radiation detector surfaces by moving a collimated gamma ray source with high spatial precision. The user can choose the size of the scanning region, the resolution of the scan, and the desired exposure time, as well as other scanning properties.

2 Setup and Usage



Part Name	Part ID Number
Stepper y Motor 1	1
Stepper y Motor 2	2
Stepper x Motor	3
4 Clamping Screws	4
Power Supply	5
USB to Arduino	6
4 Corner Screws	7

1. Position the device so that detector is close to the center of the scanning region. This will be more important for larger detectors. Make sure the power cable can reach the wall and the USB can reach your data acquisition computer but DO NOT plug them in yet.

2. Adjust the top of the frame which supports the electronics, guide rails, lead screws, and etc. so that it is at the desired height. To adjust the height, loosen the four corner screws, move the top of the frame to the desired height, then re-tighten the screws. Make sure not to fully remove the four corner screws as it is difficult to get them back in if you do. Also note that the closer the collimator is to the detector the better your results so you should try to lower the system as close to the detector surface as possible.

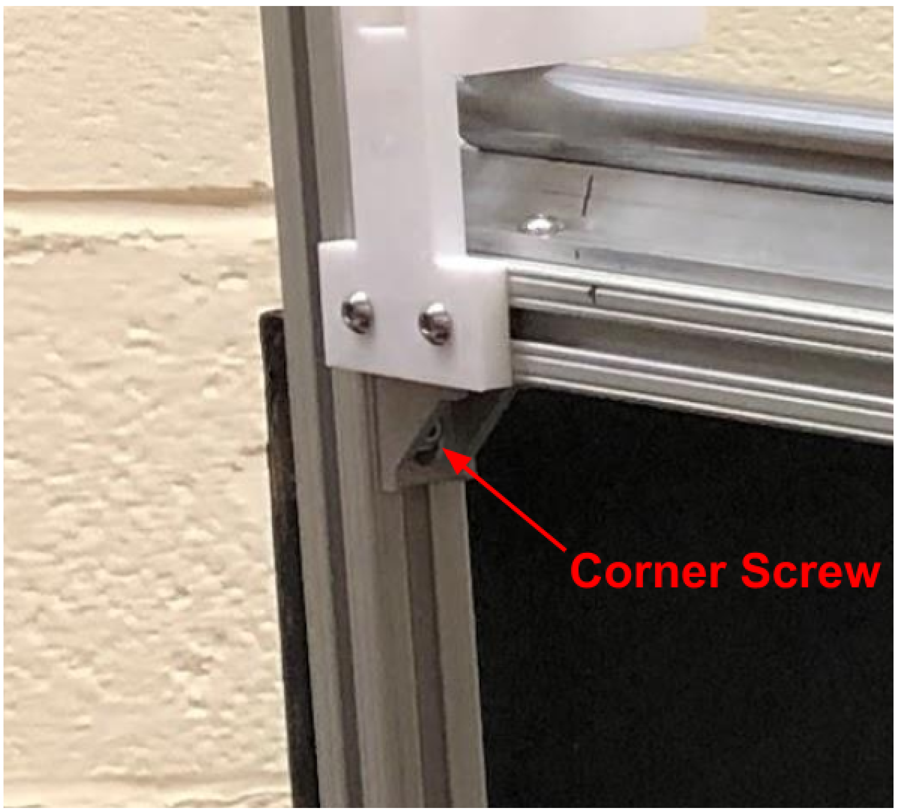
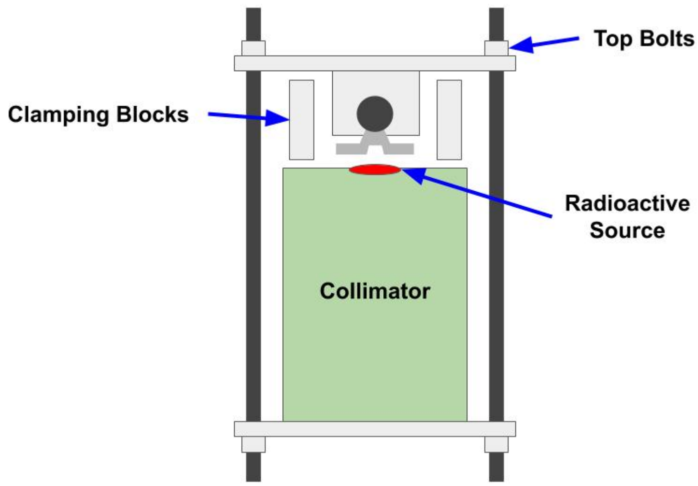


Figure 1: Shows location of one of the four corner screws.

3. After ensuring that all four corner screws are tightened and at the same height, place the collimator and the radioactive source on the collimator mount. WARNING, THE COLLIMATOR IS HEAVY! Next place the two clamping blocks on top of the collimator and tighten the four clamping screws to clamp the collimator in place.



4. The Arduino will come installed with the necessary software. The data acquisition computer requires the Python based control software to be downloaded. The Python program, along with the Arduino program, can be found in the GitHub repository github.com/zachh12/NCSU_SD. For more information on the software see Section 3. Run SeniorDesignIO.py. Follow the onscreen prompts to choose center location, scan region, granularity, and exposure time. For a more detailed description of these parameters see Section 3.1.

3 Software

To access the software needed to use the RaDSCS, go the the GitHub repository: github.com/zachh12/NCSU_SD and clone the repository. The repository includes the file that comes uploaded to the Arduino and the Python file needed to run the program.

3.1 Software on Arduino

The Arduino program defines the pins used for the stepper motor driver and switches. There is a function written that executes a calibration sequence when it is called from the Python script.

The calibration sequence:

1. The calibration procedure begins with the collimator housing unit going to the home position in the bottom left corner of the device. First, the housing unit moves left. When it hits a switch it backs off. Then it moves down, once again backing off when it hits a switch.

2. Then, the housing unit begins to move right, counting the number of steps taken, until it hits the switch on the right side. It then backs off the switch. Now the process is repeated, except the housing moves up, counting the number of steps taken until it hits the switch at the top. Once again, it backs off the switch. The number of steps taken are called steps_x and steps_y and are used later to check that the system remains calibrated throughout the scanning process.

3. Finally, to end the setup procedure, the housing unit returns to the home position in the bottom left corner.

In the program, in addition to the calibration function are four different functions: movex, movex_toEND, movey, and movey_toEND. movex and movex_toEND move the single x motor and movey and movey_toEND move the two y motors. In these functions, there are conditions that will stop the motor if a switch is hit. movex and movey will move the motors in a specified direction and number of steps. movex_toEND and movey_toEND will continue to move the motors until a switch is hit. It also counts the number of steps it takes until it hits a switch and returns that number.

In the loop portion of the code, a serial port is open and it waits to receive inputs from the port. From the Python script, a string is sent to the Arduino. In the loop, the string is parsed into variables motor, steps, and direc. The motor variable denotes which function to use: if motor = 'Calibration' then run the Calibration function, if motor = 'x' then movex is used, if motor = 'y' then movey is used, if motor = 'xEND' then movex_toEND is used, and finally if motor = 'yEND' then movey_toEND is used. The steps variable tells the stepper motors how many steps to take in the movex or movey functions. Lastly, the direc variable tells the motors which way to turn in the movex, movey, movex_toEND and movey_toEND functions. A '1' means to turn counterclockwise and '0' means clockwise.

3.1.1 Arduino Software Variables

motor	string used to determine which function to use to move x or y motors
steps	string used to determine how many steps to take in movex and movey functions
direc	string used to tell motor which way to turn in move functions

3.2 Python Software on Data Acquisition Computer

The Python software that runs on the data acquisition computer establishes a serial connection with the Arduino and first calls the calibration function to run from the Arduino. From the calibration it receives the number of steps it takes to go from one side of the x axis to the other and the number of steps it takes to got from one side of the y axis to the other (stepsx_tot and stepsy_tot) from the Arduino.

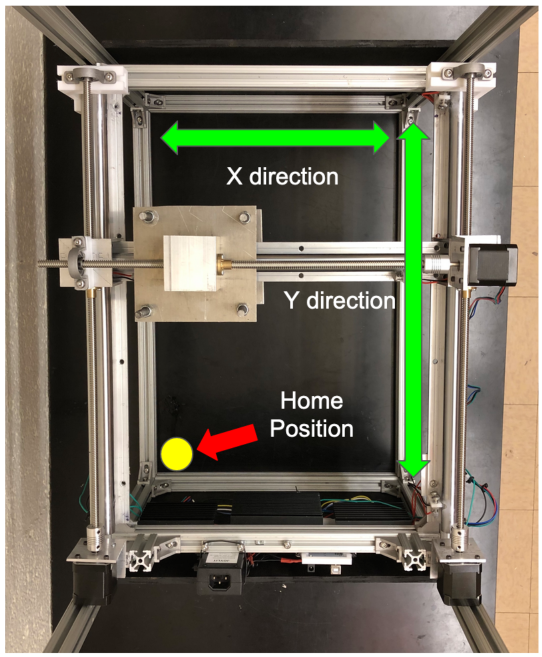


Figure 2: Top view of the device showing where the home position is and what is defined as the x and y directions.

Next, the user is prompted to use the keypad to position the housing unit from the home position to their desired position from which to determine the scanning region (the "zero position"). Once the desired position has been selected, defining the scanning area is necessary. The user is next prompted to select the distance from the left, right, top, and bottom of the current housing unit location for the scanning region.

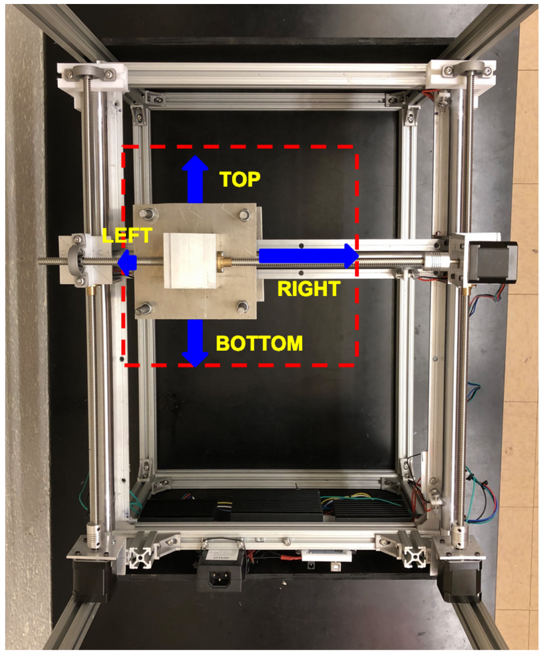


Figure 3: Top view of the device demonstrating how scanning region parameters are specified. Note how the bottom is defined as the side with the stepper drivers and Arduino.

The housing unit then moves to the top left corner of the specified scanning region. Next, the user is prompted to specify granularity (the distance between scanned regions/how far the housing unit moves to get to the next scanned region). Lastly, the user is prompted to specify how long they would like for the housing unit to remain at each scanned region.

Next, a text file is opened up where the positions and timestamps will be stored throughout the positioning process. The positioning process then begins in a unidirectional scanning motion.

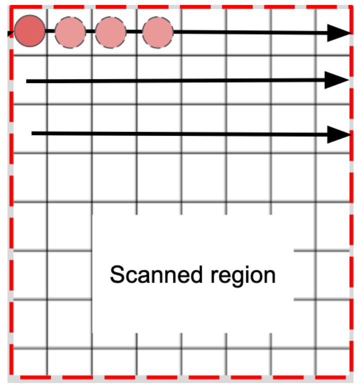


Figure 4: Diagram of the unidirectional scanning motion.

During this scan, at the end of row in the scanning region, the housing unit is moved to the right end until it hits the switch. Then it moves back left until it hits the switch on the other side and the steps it takes is recorded and compared with the number of steps from the calibration.

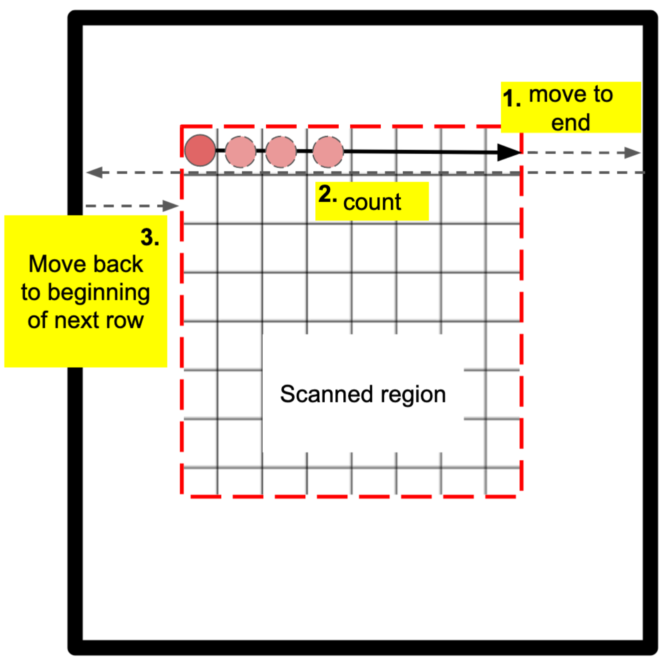


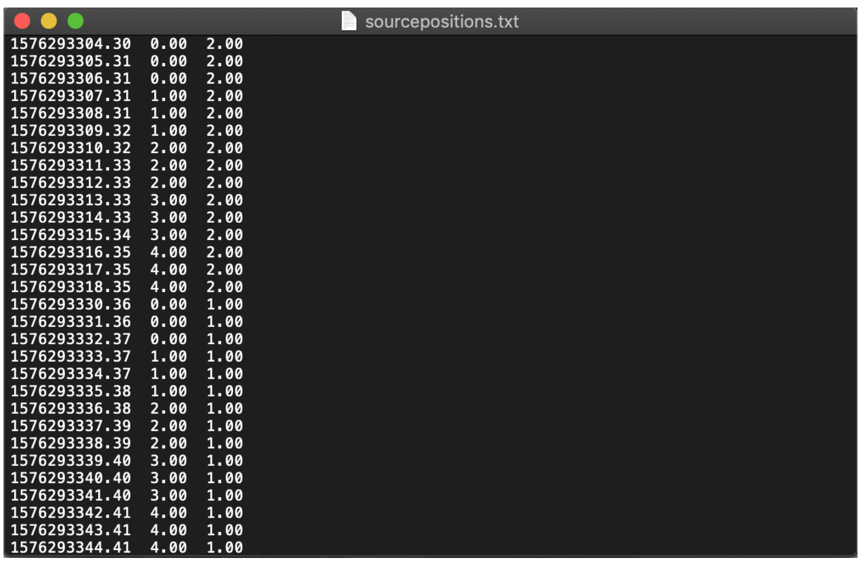
Figure 5: How calibration is checked during scan.

3.2.1 Python Software Variables

x_pos	x position relative to home position
y_pos	y position relative to home position
x_dist	x position relative to top left corner of scanning region
y_dist	y position relative to top left corner of scanning region

3.3 Data Output

While running, this device generates a text file to record the current position of the gamma radiation beam and the timestamp at 1 second intervals. The left column is the timestamp, the middle column is the x position, and the right column is the y position. These positions are relative to the home position in the bottom left corner.


sourcepositions.txt

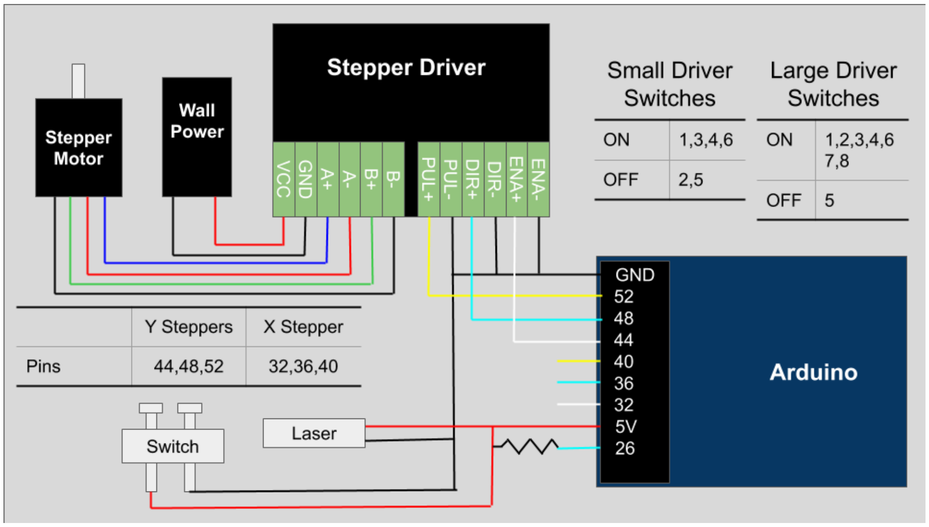
```
1576293304.30  0.00  2.00
1576293305.31  0.00  2.00
1576293306.31  0.00  2.00
1576293307.31  1.00  2.00
1576293308.31  1.00  2.00
1576293309.32  1.00  2.00
1576293310.32  2.00  2.00
1576293311.33  2.00  2.00
1576293312.33  2.00  2.00
1576293313.33  3.00  2.00
1576293314.33  3.00  2.00
1576293315.34  3.00  2.00
1576293316.35  4.00  2.00
1576293317.35  4.00  2.00
1576293318.35  4.00  2.00
1576293330.36  0.00  1.00
1576293331.36  0.00  1.00
1576293332.37  0.00  1.00
1576293333.37  1.00  1.00
1576293334.37  1.00  1.00
1576293335.38  1.00  1.00
1576293336.38  2.00  1.00
1576293337.39  2.00  1.00
1576293338.39  2.00  1.00
1576293339.40  3.00  1.00
1576293340.40  3.00  1.00
1576293341.40  3.00  1.00
1576293342.41  4.00  1.00
1576293343.41  4.00  1.00
1576293344.41  4.00  1.00
```

Figure 6: Text output example that is generated during the scan. The left column is the timestamp, the middle column is the x position, and the right column is the y position. These positions are relative to the home position in the bottom left corner.

4 Specifications

Smallest Granularity	0.5mm
Maximum Scan Region	20cm x 40cm
Positional Accuracy	+/- 0.25mm
Maximum Weight	50lbs

5 Electronics



Each stepper motor is connected to a stepper driver. These drivers all pull power from the 2 Amp 24V wall power supply. The side of the stepper driver which is connected to the Arduino is grounded at the Arduino ground for all 3 drivers.

6 Parts List

Part	Quantity
Arduino Mega 2560	1
NEMA 17 Stepper Motor	3
Stepper Motor Driver	3
2 Amp 24 Volt Power Supply	1
OrangeA Linear Rail 2x1000mm	1
SBR20 Linear Motion Slide Block	1
ReliaBot 400mm T8 Lead Screw	1
ReliaBot 550mm T8 Lead Screw	1

7 FAQ's

1. **Q.** Does it only work with gamma rays?
   A. While this device was designed with gamma rays in mind it can be used with other types of radiation so long as an appropriate collimator is installed.

2. **Q.** Can I use a heavier collimator if necessary?
   A. We have done test showing the RaDSCS moving up to 75lbs, so yes

you can use a heavier collimator but the effects on positional accuracy are unknown.