

Configuring the platecoordinates.dat file

The coordinates for each well in the wormbot array are stored in a file in the wormbot directory called platecoordinates.dat. The format of this file is as follows:

PlateRowWell,x, y

For example:

```
jpitt@jpitt-PowerEdge-T30:/wormbot$ more platecoordinates.dat
1A1,150,825
1A2,150,1950
1A3,150,3075
1A4,150,4200
1B4,1285,4200
1B3,1285,3075
1B2,1285,1950
1B1,1285,825
1C1,2420,825
1C2,2420,1950
1C3,2420,3075
1C4,2420,4200
2A1,150,6507
2A2,150,7632
```

Every well for every plate requires a coordinate in this file for the WormBot to be able to image that well. This should only need to be configured once after construction and should never need to change unless your 12-well plate dimensions change.

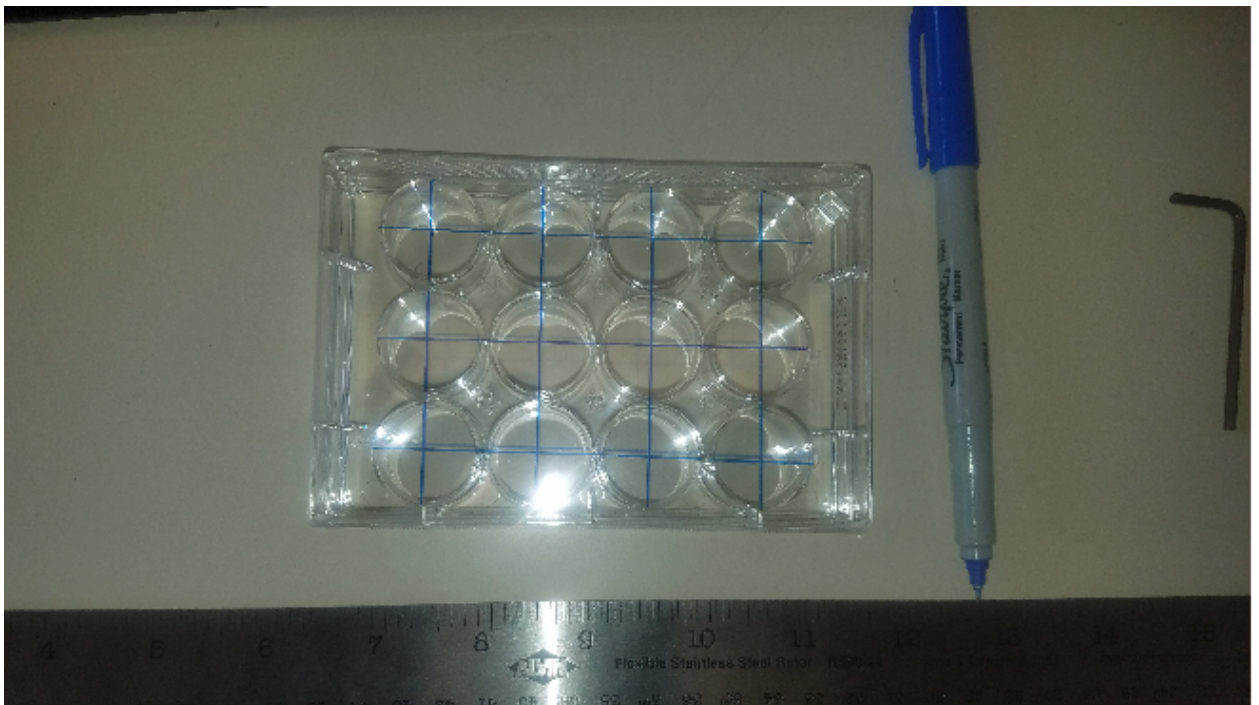
Note: our lab uses Olympus Plastics Suspension culture plates Cat#:25-101 purchased from Genesee Scientific

There are two approaches to generating the platecoordinates.dat file. One fast but less accurate, one slow and tedious, but highly accurate.

Fast Configuration

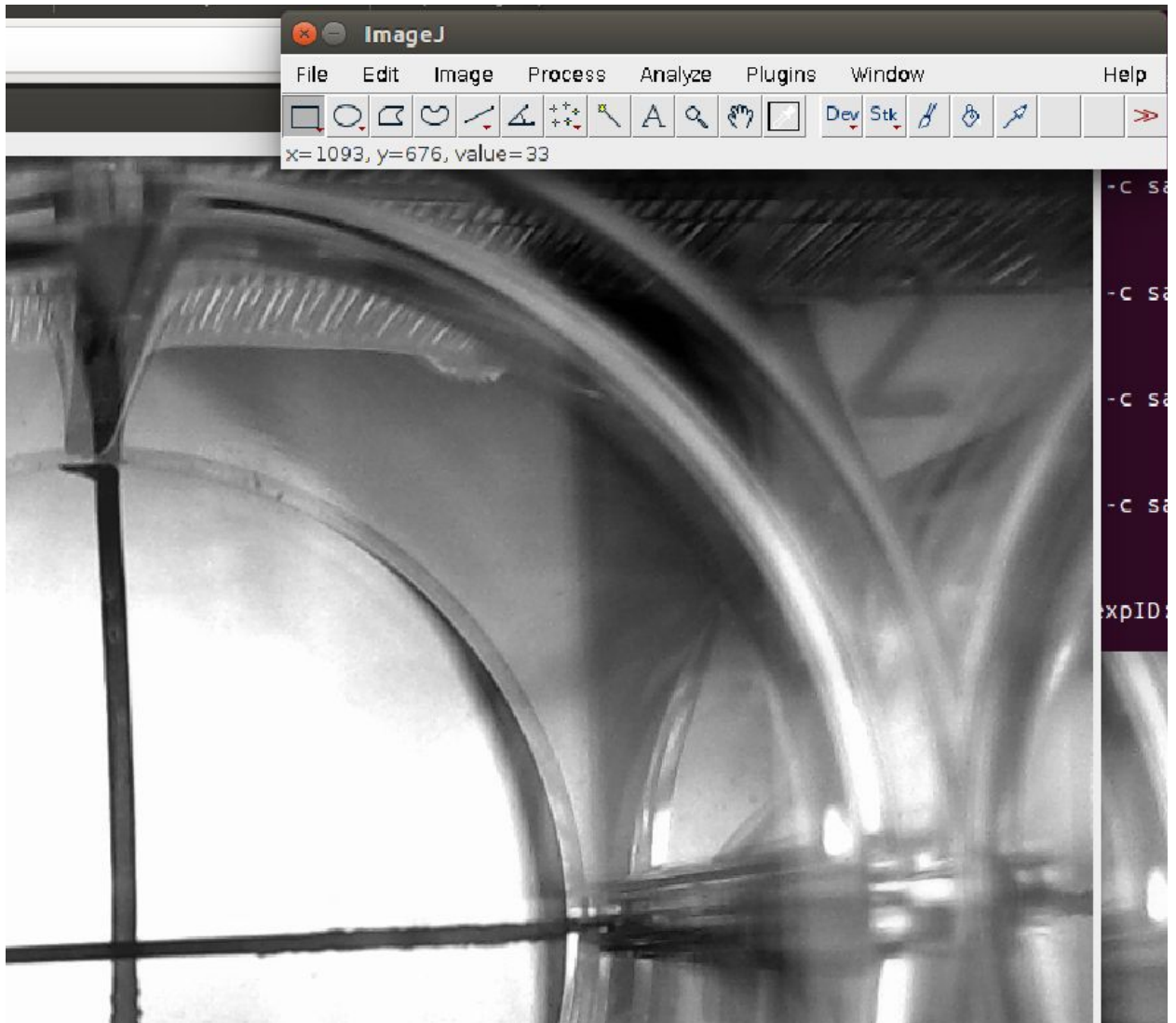
Because the WormBot light table is machined by CNC the coordinates of each well should be at least close to those found in the default platecoordinates.dat file that comes with the WormBot software. So to find the desired center coordinate of each well we will simply measure the differences between the current center positions and the desired position for 1 or maybe a few wells and then run a python script to generate a new platecoordinates.dat file.

1. Setup an experiment to collect data on a few empty wells across multiple plates where you've marked the center of the well with a sharpie. Make sure that one of the experiments is in Plate 1 well A1.



2. After you collected a couple frames of data for each well, use the Scheduler to stop those experiments. Open the raw (unaligned) images in ImageJ for the experiment in plate 1 well A1. The raw file names are in the /wormbot/XXX directory where XXX is the experiment ID for the image, the images are numbered as frame000000.png. So the path to this image in experiment ID 122 was /wormbot/122/frame000000.png or 127.0.0.1/wormbot/122/frame000000.png if you want to grab it from the web browser.

Once the file is opened use the cross hair cursor to find the coordinate of the center of the well. In the below example it was at 1093,676.



3. Calculate the dx and dy for desired center. The resolution of the camera we are using is 1920 x 1080 so the center should be at 960,540 which in this case gives us a dx,dy of -133,-136. The Robot axes are orthogonal to the image axes and each pixel corresponds to 1.25 Wormbot units, so the robot dx = -camera dy/1.25 and robot dy = -camera dx/1.25. For our example Robot dx = 109 and Robot dy = 106

4. Edit the `/wormbot/generate_coordinates.py` script. Open a terminal and type:

```
sudo gedit /wormbot/generate_coordinates.py
```

```
jpitt@jpitt-PowerEdge-T30:/wormbot$ sudo gedit /wormbot/generate_coordinates.py
```

This will open a text editor allowing you to modify the `generate_coordinates` script. Apply your Robot `dx` and `dy` values to the variables `x_0` and `y_0`

```
Open ▾ [F1]
nums = [1,2,3,4,4,3,2,1,1,2,3,4]
letters = ["A", "B", "C"]
x_0 = 150
y_0 = 825
dx_well = 1135
dy_well = 1125
dx_plate = 3828
dy_plate = 5682

with open("platecoordinates.dat", "w") as f:
    for i in range(12):
        plate_x = x_0 + (i / 3) * dx_plate
        plate_y = y_0 + (i % 3) * dy_plate
        for j in range(12):
            well_x = plate_x + (j / 4) * dx_well
            well_y = plate_y + (nums[j] - 1) * dy
```

In our case it will become:

```
Open ▾ [F1]
nums = [1,2,3,4,4,3,2,1,1,2,3,4]
letters = ["A", "B", "C"]
x_0 = 259
y_0 = 931
dx_well = 1135
dy_well = 1125
dx_plate = 3828
dy_plate = 5682

with open("platecoordinates.dat", "w") as f:
    for i in range(12):
        plate_x = x_0 + (i / 3) * dx_plate
```

Hit the save button on the right or press `Ctrl S` to save the edited script

5. Run the script by opening a terminal and typing:

```
sudo python /wormbot/generate_coordinates.py
```

```
jpitt@jpitt-PowerEdge-T30:/wormbot$ sudo python /wormbot/generate_coordinates.py
```

6. You now have a new platecoordinates.dat file in place, verify it has been changed by inspecting the file with the head command:

```
head platecoordinates.dat
```

```
jpitt@jpitt-PowerEdge-T30:/wormbot$ head platecoordinates.dat
1A1,259,931
1A2,259,2056
1A3,259,3181
1A4,259,4306
1B4,1394,4306
1B3,1394,3181
1B2,1394,2056
1B1,1394,931
1C1,2529,931
1C2,2529,2056
jpitt@jpitt-PowerEdge-T30:/wormbot$
```

7. To verify that the new coordinates are acceptable setup a few new experiments with the scheduler and repeat steps 2 and 3 above and verify that the dx and dy are now close to zero. Usually this is sufficient to get the wells centered, however if they are not, you may need to use the more tedious approach described below...

Note: Coordinates are assigned to experiments when they are entered in the Scheduler...therefore any changes you make to the platecoordinates.dat file will only affect the positioning of future experiments. Experiments already underway will use the coordinates at the time they were setup.

Manually determining the center points for all 144 wells

If the above approach was not successful or you need greater accuracy you can open a serial connection to the WormBot and pass it commands directly in order to position the camera and determine the optimal center point for each well. This will involve using 2 programs simultaneously, the Arduino IDE to issue commands and the Gvvcview application to view the camera in real-time, you will also need to have a text editor with the platecoordinates.dat file open to record the values for each well.

1. Stop the controller by opening a terminal and typing:

```
sudo killall controller
```

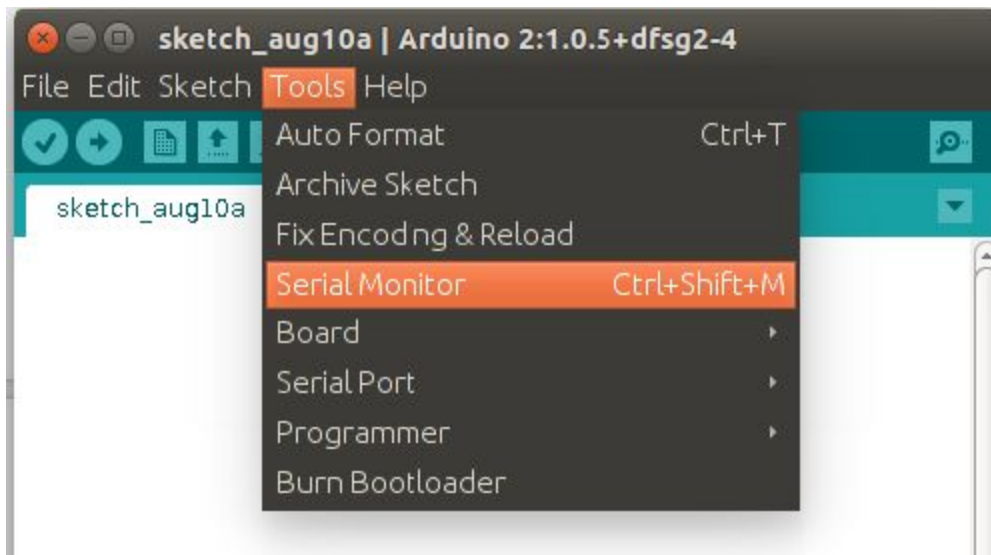
```
jpitt@jpitt-PowerEdge-T30:/wormbot$ sudo killall controller  
[sudo] password for jpitt:
```

2. Open the Arduino IDE:

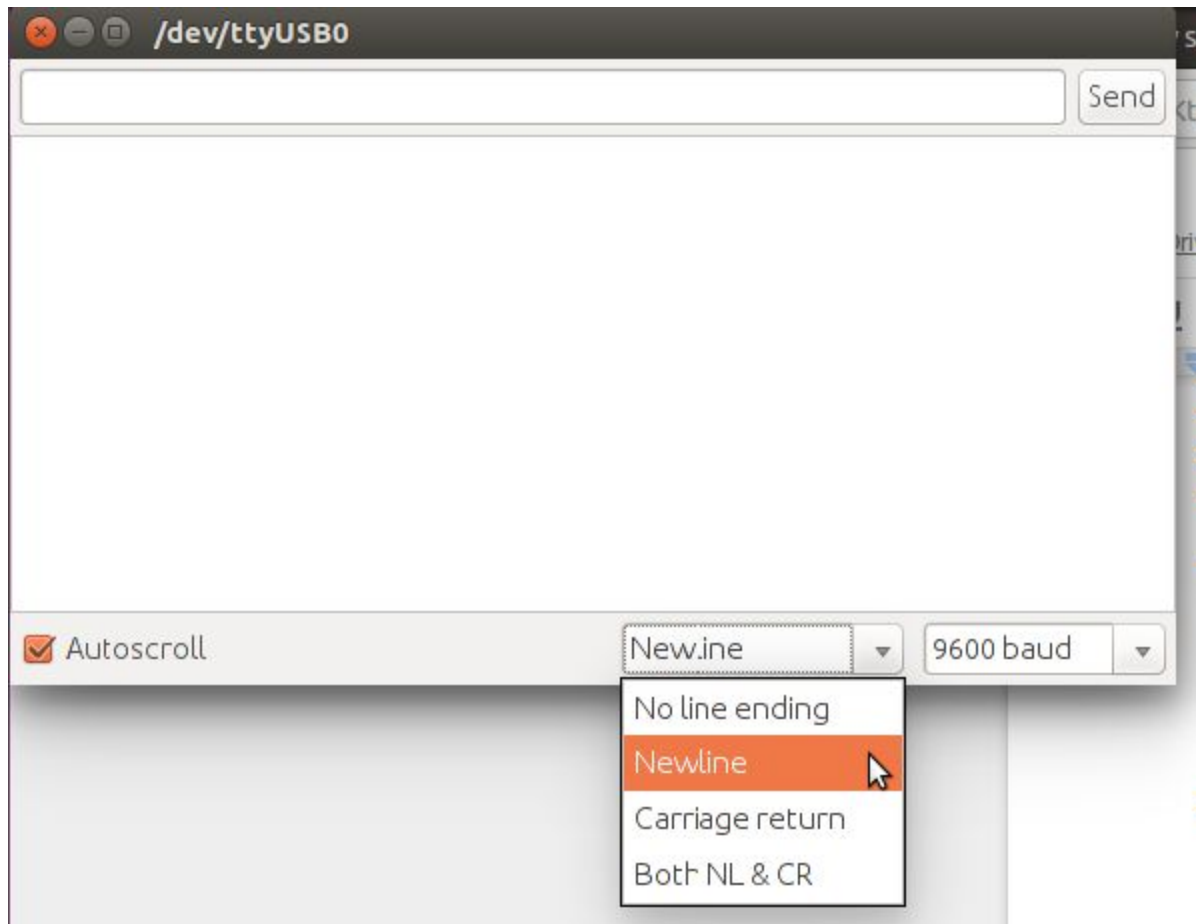
```
sudo arduino
```

```
jpitt@jpitt-PowerEdge-T30:/wormbot$ sudo arduino
```

3. From the Arduino Tools menu select Serial Monitor

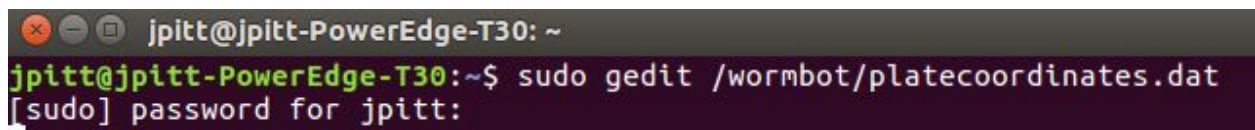


4. In the **Arduino Serial Monitor window** make sure the baud rate is set to 9600 and that Newline is selected from this dropdown menu:



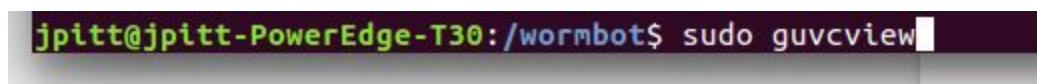
5. Open a terminal and start a text editor to edit the platecoordinates.dat file:

```
sudo gedit platecoordinates.dat
```

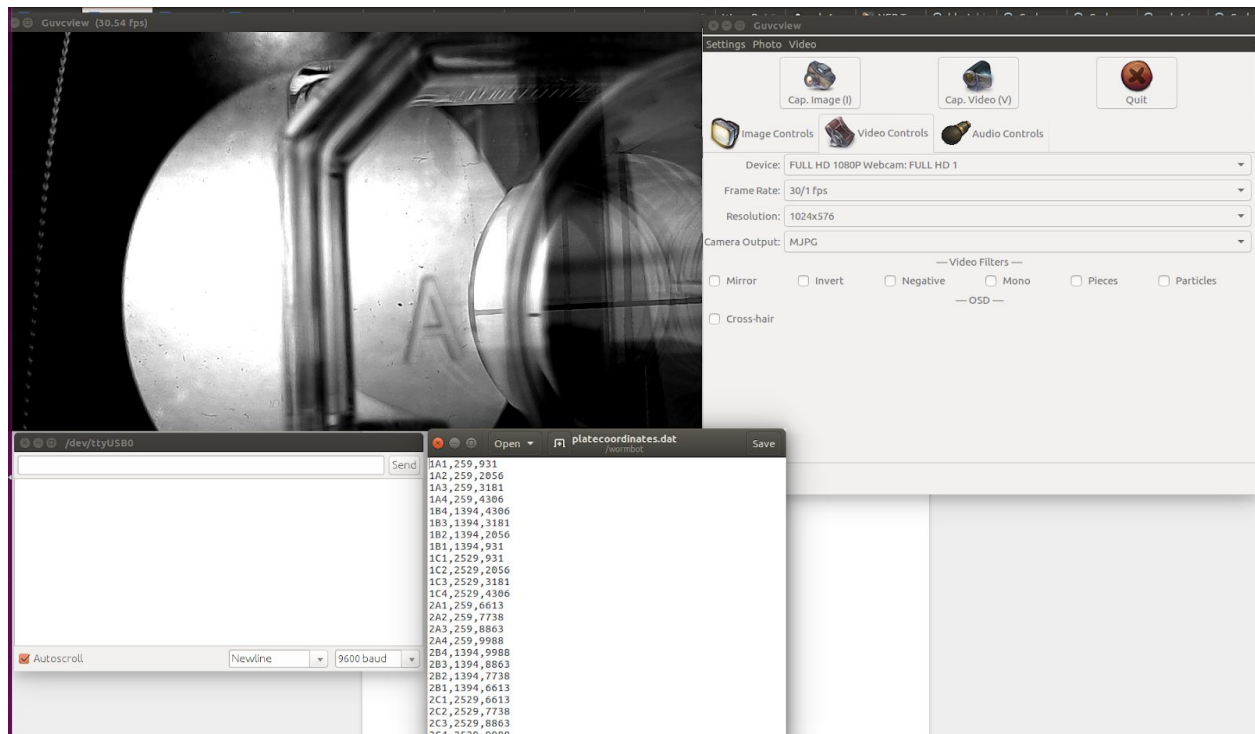


6. Open a new terminal and start the Gview application:

```
sudo gview
```



7. In the **gucvview video controls tab** set the resolution to something lower for the camera (1024x576 is good) so that you can get all three windows on the screen at once, like this:



8. **Controlling the WormBot manually.** The WormBot accepts a series of commands through the serial connection that you can enter manually in the serial monitor to move the camera around. Each time the robot completes a command successfully you will see an RR show up in the serial monitor window. These are the available commands (after typing the command hit enter or click the send button, **note commands are Case Sensitive**):

ZZ Sends the camera head to the origin 0,0 Xmin,Ymin

LL Sends the camera head to the Xmax,Ymax position

P Prints the current camera head coordinates to the serial connection

MXn Moves the camera to the x position given by the integer n for example MX0 will move to X=0, MX10000 will move to X=100000

MYn Same as MX but for Y direction

Mn,n Moves the camera to the coordinate n,n. Example M0,0 will go to 0,0 M430,1975 will go to x=430 y=1975

W A S D These 1 character commands nudge the camera head by 1 step directions are: D=X+1, A=X-1, S=Y+1, W=Y-1

9. Using the above commands move the camera head to the center point of a well and once centered enter the P command and enter that x,y coordinate in the platecoordinates file. Remember, the robot coordinates are flipped relative to the camera so to move the center point up and down changed the Robot X coordinate, for left right the Y coordinate. Repeat until finished. Save the platecoordinates.dat file.

