**HIW 3.0 Checklist/Troubleshooting**

Questions? Contact Courtney via courtney.j.kim@berkeley.edu

\*At Zuckerman Institute, the IT team had to configure the computers in order for the robocopy to work (aka to copy over the files to a network folder secured by ZI). If you’re robocopying files to a shared network folder regulated by the institution, make sure you set up your computer with your IT team.

**<Modifying computer settings> - skip this if the computers are already configured**

* Turn off screen saver
* Make sure computers don’t go to sleep
  + Setting > System > Power & battery > Screen and sleep
  + Set all 4 options to “Never”
* Turn off auto-updates
  + Download WUB: <https://www.sordum.org/9470/windows-update-blocker-v1-7/>
  + Click on “Disable Updates” and then “Apply Now”

**\*\*\*Reminder before each run**

* Put half the regular amount of the brown strips and bedding (some mice will fill the odor port with bedding)
* Take out the food/water holder and put some food pellets on the floor of the cage
* Periodically check for computer data storage and make sure it isn’t full

**<Running the experiment>** - 4 cages and 2 computers

**BOX CONFIGURATION**

|  |  |
| --- | --- |
| Box 1 | Box 3 |
| Box 2 | Box 4 |

\*\* HIW-3 computer : Box 1 & 2

\*\* Delphi computer: Box 3 & 4

* This configuration is what we used at Columbia when we could only run 2 cages per computer. The following instructions assume this setup. Modify as needed.
* Make sure all the lines are hooked up
  + CoolDrive (N=4) is powered and connected to the Output Expander (N=4) and the final valves (N=4)
    - Power the CoolDrive FIRST before passing the logic
  + Camera (N=4) is connected to the Output Expander (N=4)
  + Camera is directly plugged into the computer (DON’T PLUG CAMERA CABLE TO THE USB EXPANDER HUB – IT WILL RESULT IN DROPPING FRAMES)
  + USB Expander Hub (N=2) is plugged into the computer
  + USB Expander Hub is plugged in with Harp expander cable (N=4)
  + Ethernet cable (N=2) is directly plugged into the computer with an adapter
  + Computers are powered
* Restart all the programs before each run (only in the prototyping stage when testing new github pull requests)
  + Bonsai
  + Github Desktop
  + Visual Studio Code
* Make sure to you’re using the correct Bonsai workflow
  + Open **[This PC]** >**[Windows (C:)]** > **[Users]** > **[wherever you put it]** > **[Delphi]** > **[.bonsai]**
  + Run the Setup file (only if you can’t see the Bonsai program icon in the folder) with powershell
  + **[Run as Administrator]** the Bonsai program in the same folder **[.bonsai]**
  + “Open file” in **[This PC]** > **[Windows (C:)]** > **[Users]** > **[wherever you put it]** > **[Delphi]** > **[src]** > **[DelphiMain]**
  + Open Github Desktop, and make sure…
    - Current repository = **[Delphi]**
    - Current branch = **[Main]**
  + In Github Desktop, click **[Pull origin]** if there have been any changes to the DelphiMain file
  + In Bonsai, click **[Reload extensions]** (the button on the far right in the same row as the start button) if there have been any changes
  + Open Visual Studio Code and make sure you’re working on the **[Main]** branch (it should say “Main” on the bottom left corner)
  + Check the **rule yml** files you’re going to use and make sure everything’s correct
  + Make 4 **[DelphiSession]** files if you don’t see them (copy and paste the already-existing DelphiSession file and rename them)
    - **HIW-3 Computer:** DelphiSession1, DelphiSession2
    - **Delphi Computer:** DelphiSession3, DelphiSession4
    - **\*\*If you’re only running 1 box, there should be one DelphiSession file created already, and you don’t have to do anything else.**
  + In VSCode, click on each **[DelphiSession]** and make sure all the parameters are correct

|  |  |
| --- | --- |
| **METADATA** | |
| Animal ID | Depends |
| loggingRootPath | \*\*Modify as needed! This is the local folder where the data will be saved.  **DelphiSession1:** C:\Users\jk4817\temp\_data\_box1  **DelphiSession2:** C:\Users\jk4817\temp\_data\_box2  **DelphiSession3:** C:\Users\jk4817.adcu\temp\_data\_box3  **DelphiSession4:** C:\Users\jk4817.adcu\temp\_data\_box4 |
| remoteTransferRootPath | \*Modify as needed! This is the remote folder where the data will be moved to from the local folder.  **DelphiSession1**: \\locker-smb.engram.rc.zi.columbia.edu\axel-locker\piriform\behavior\HIW3\_Robocopy\_Rawdata\Robocopy\_Box1\_Rawdata  **DelphiSession2**: \\locker-smb.engram.rc.zi.columbia.edu\axel-locker\piriform\behavior\HIW3\_Robocopy\_Rawdata\Robocopy\_Box2\_Rawdata  **DelphiSession3**: \\locker-smb.engram.rc.zi.columbia.edu\axel-locker\piriform\behavior\HIW3\_Robocopy\_Rawdata\Robocopy\_Box3\_Rawdata  **DelphiSession4**: \\locker-smb.engram.rc.zi.columbia.edu\axel-locker\piriform\behavior\HIW3\_Robocopy\_Rawdata\Robocopy\_Box4\_Rawdata |
| chargeTime | 0.2 |
| minimumPokeTime | 0.01 |
| maximumPokeTime | 10 |
| robocopyTimeInterval | 3600 |
| maxVideoLength | 60 |
| minOdorDelivery | 0.1 |
| maxOdorDelivery | 10 |
| switchTime1 | 0.01 (vacuum OFF), 0.035 (vacuum ON) |
| switchTime2 | 0.02 (vacuum OFF), 0.00 (vacuum ON) |
| useVacuum | Depends (write either true or false) |
| vacuumDelay | 0.025 |
| portValveCycleTime | 0.110 |
| vacuumDuration | 0.01 |
|  | |
| **CAMERA PROPERTIES** | |
| imagingRate | 60 |
| exposureTime | 10000 |
| serialNumber | “[FLIR camera serial number 8 digits]”  \*This is written on the camera. Make sure you don’t put in the number for another camera filming another box |
| preEventTime | 2.5 |
| postEventTime | 2.5 |
|  | |
| **HARP PROPERTIES** | |
| comPort | COM#  \*Check on Device Manager which COM port you’re using. This should not be the camera COM port but the output expander box COM port.  \*As long as the Output Expander cable stays connected to the computer via the *same* USB port, the COM# won’t change |
| showHarpLeds | false (depends) – keep it true while finding COM#  \*This is the LED on the output expander box. |
|  | |
| **LINE MAPPINGS** | |
| odorMap | - {name: "OdorA", line: 5}  - {name: "OdorB", line: 6}  - {name: "OdorC", line: 7}  \*\*Do not assign anything to line 9 |
| portLine | 0 |
| vacuumLine | 2 |
| auxLine | 0 |
| cameraTriggerLine | Pwm0ToOut1 |

A diagram of a graph

Description automatically generated with medium confidence

* Click **[ctrl + S]** on VSCode to save the changes
* Make necessary changes in Bonsai
  + Click anywhere on **[Workflow]** window, and ControlKeys will pop up on the right
  + Select the correct **rule yml file**
* Turn on the 2 nitrogen tanks
  + Make sure you have enough gas (full tank is 2500 lpm)
  + Pressure gauge on the tank’s flow regulator is ~20
  + Bottle – 0.075 lpm per bottle (0.225 per box when using 3 odors)
  + Air – 0.075 lpm
  + Top 2 regulators are for bottles, bottom 2 are for air going into final valves (this is for Columbia setup)
  + \*\*\*For rotameters, check the floats and the conversion document to dial in the correct flow rate
* Place the IR source with the light diffusing cloth underneath
* Start the Bonsai program and press **[Shift + F1]** to start the cameraand **[Shift + Space]** to load the rule/start the session
  + Make a poke with the program and smell the port, making sure the final valve is opening, the odors are changing correctly and the lines are not clogged
  + Open livestream from the **[Workflow]** window **[Value.Image]** and check that the camera focus/depth is desirable. Adjust focus/exposure accordingly by turning the knobs on the camera.
* Stop the program and check for data files in the robocopy destinations in the “piriform” folder (or whatever remote folder you’re using at Allen/Northwestern), and once you confirm that robocopy is running properly, delete these test files.
* Put the mice into the modified cages (with half amount of bedding/cardboard pieces and enough food), put the lids on (the ones without the carved space for water bottles), and put them on the rack
  + Make sure you’re putting the correct mice to the correct boxes! Record their animal IDs!
* Screw in the poke ports (M4 screws) to the cages
* Start the experiment (press start) and configure the ROI
  + Navigate to **[CameraController] 🡪 [RoiActivity]** and on the right side ‘Properties,’ click on **“Point[][] Array”** and the three dots (“…”) next to it
  + A screenshot of a computer

    Description automatically generated
  + Here, you can select the ROI that will trigger the stimuli delivery
    - Move around the box with left click and drag
    - Adjust box shape/size with right click and drag
    - A screenshot of a computer

      Description automatically generated
  + If you want to adjust the number of pixels (within the ROI) that should turn white to deliver the stimuli, navigate to **[CameraController] 🡪 [GreaterThan]** and adjust the number on the right panel
    - With the current ROI box size, 1500000 is the recommended value
* Starting the experiment…
  + Make sure to click **[Shift + Space]** to load the rule
    - When you’re changing the rule, double-check the rule file in VSCode that it is written correctly
  + Open livestream video and DelphiVisualizer for convenience
  + Make sure there are no dropping frames ([Camera Controller] window > click on both the Spinnaker node and the node preceding it – check that the two numbers align)
  + Make sure the blue service requisition card is on the cage

**When you’re done with the experiment…**

* Press the “STOP” button in DelphiVisualizer
* Check that the odor valves + final valve are closed

\*Record **who** ran **when** in **which** **box** in the “HIW3\_Experiment\_Records\_2024” Excel File in the piriform locker > HIW 3.0 (or whatever file you use at Allen/Northwestern)

**Tips**

* **Opening DelphiVisualizer**
  + **[Workflow] > [TableLayoutPanel]** node

**Troubleshooting**

* **Poke detection failure**
  + Check jumpers on Harp expander breakout and that they’re receiving input from the software and not the physical buttons on the breakout
  + In the Bonsai workflow, [Devices] [Delphi Controller] [LineMappings.AuxLine], start the workflow, and check that it’s Aux 0, not Aux 1
  + In the Bonsai workflow, [Devices] > [Delphi Controller] > [HasFlag], start the workflow, and check that it’s Aux 0, not Aux1
  + If you detect an error, change it in VSCode
    - [DelphiSession] > “auxLine” > Set to 0
* **Changes made in VSCode are not being reflected in Bonsai**
  + You’re most likely using the wrong directory.
  + [This PC] > [Windows (C:)] > [Users] > [jk4817] or [jk4817.adcu] > [Delphi]
  + Type “cmd” where the path is
  + Type “code .” in terminal + enter
  + VSCode will open; work on “DelphiSession.” Any changes you make here AND saved will be reflected on Bonsai.