BatteryRobot Manual

Version 0.1.1

9th May 2024

Document Control Page

Document Name: Battery Robot Manual

Version Date: 9 May 2024

Version: 0.1.1

|  |  |  |
| --- | --- | --- |
| Document History (Prior version sub-revision history are suppressed) | | |
| Date | Initials | Version or Revision |
| 9th May 2024 | XA | Version 0.1 Initiate documentation |

Table of Contents

[Project Overview 4](#_Toc169464312)

[Robot Parts 4](#_Toc169464313)

[Potentiostat 4](#_Toc169464314)

[Electrolyte Synthesis 4](#_Toc169464315)

[Testing Electrolytes 5](#_Toc169464316)

[Requirements 6](#_Toc169464317)

[Folder/File Hierarchy 7](#_Toc169464318)

[Version Control 8](#_Toc169464319)

[Codebase 9](#_Toc169464320)

[Main.nproj 9](#_Toc169464321)

[Locators (locs) 9](#_Toc169464322)

[Classes and Instructions for Usage 10](#_Toc169464323)

[BatteryRobot 10](#_Toc169464324)

[T8 10](#_Toc169464325)

[PowderShaker 10](#_Toc169464326)

[PStat (Todo) 10](#_Toc169464327)

[Gamry 10](#_Toc169464328)

[Powder Protocols 10](#_Toc169464329)

[External Libraries 10](#_Toc169464330)

[Errors Raised for Safety Purposes 10](#_Toc169464331)

[General Limitations and Warnings 11](#_Toc169464332)

[Preventing Vial and Pipette Collisions 11](#_Toc169464333)

[Pipette Collection Order 11](#_Toc169464334)

[Moving Robot Arm Around Deck 11](#_Toc169464335)

[Appendix 1: All Locators 12](#_Toc169464336)

[Pipette Locs 14](#_Toc169464337)

# Project Overview

The main purpose of this project is to automate the process of electrolyte synthesis and testing. In doing so, maximum operational efficiency and reduced labour costs for battery development will be achieved when automation is scaled globally.

## Robot Parts

**Robot Arm(Rob)**

The robot arm is the main arm used for transport of vials and dispensing of solvents. It consists of a gripper and pipette holder. The gripper is used to move vials around different locations on the deck e.g. (rack, carousel, heat plate etc). Meanwhile the pipette holder is used to collect plastic disposable pipettes and draw/dispense solvents to synthesize electrolytes.

**Powder Dispenser(P2)**

The powder dispenser is attached to the carousel and shakes at specific amplitudes and durations to dispense a given mass of powder. It is capable of dispensing masses as fine as 10mg with 5% error when creating solutions.  
  
These errors can simply be accounted for by adjusting the formulation i.e. the amount of solvent used to form the solution.

**Heat Plate(T8)**The heat plate is a metal 4x3 rack that is used to heat up vials to specific temperatures. Beneath it, is a magnetic stirrer that spins at a given speed to stir the contents of the vials as it is being heated.

PotentiostatThe potentiostat functions independently from the robot. A pump from the robot is connected to the carousel that draws specific amounts of synthesized electrolyte and pumps it to electrodes that are connected to a Gamry potentiostat. 3 types of measurements can then be conducted on the electrolyte: **potentiostatic EIS, galvanostatic EIS, and cyclic voltammetry**. Testing data is output onto an Excel spreadsheet for each process.

## Electrolyte Synthesis

Electrolytes are synthesized on the carousel.

## Testing Electrolytes

# Requirements

North API is flexible with the Python versions and will run as long as an interpreter is provided.

However, Python 3.7.9 is required to install Gamry packages through the .whl files they provide. Therefore, the entirety of this project is written using Python 3.7.9, including the packages installed.

# Folder/File Hierarchy

The project is located under the BatteryRobot folder, whose hierarchy is as follows:

A diagram of a project

Description automatically generated

The PStat folder is ignored by Git, and only available on the lab PC as it contains an unreleased alpha version of the Gamry API.

# Version Control

Project uses Git for version control. Project repository is on GitHub under the Lopez Group email.

[GitHub link here](https://github.com/lopezgroup-NU/BatteryRobot)

To clone:

Request collaborator access from the lopezgroup-NU account. Once access is granted, clone the repository using either Github Desktop or the command line.

# Codebase

## Main.nproj

The most important file for this project is the \*.nproj file (for this project, it’s named Main.nproj). This file contains all the robot configurations, locators, and general project information.

## Locators (locs)

Locations represent recorded points in 3D on the deck, e.g. the location of the carousel, location of a rack etc. Locators can either represent a standard location or a grid. A standard location is a simple point in 3D on the deck, while a grid is a Python array of standard locations representing a 3D grid on the deck. Generally, grids for this project are defined in 2D.  
  
A grid locator named **my\_rack**, for example, could represent the 3x4 vial rack:

o o o o

o o o o

o o o o

Where the grid locator is the Python array [ [index of top right vial], [index of middle right vial], …. [index of bottom left vial] ]  
  
The indexing of a rack with n grid points is  
  
n-3 … 0

n-2 … 1

n-1 … 2

where **my\_rack[0]** represents the top right grid point, and **my\_rack[n-1]** represents the bottom left grid point.

All locators are imported from MainProject\src\main\Locator.py. However, locations are stored in the Main.nproj file, not the Locator.py file itself. They are only exported to the latter for script execution.

Locators can be edited through the “Locator” tab on the NorthIDE (see “View”). All the currently mapped Locs are accessible through Locators.py (See [Appendix 1](#_Appendix_1:_All)).

Locations can either be relative to the pipette holder or gripper of the robot. On the “Locator” tab, each Locator will have an icon denoting which part of the robot it is relative to.

## Classes, Methods, and Instructions for Usage

The classes below are defined for each component of the project as defined in the Project Overview section above. These files are imported into main.py in the main folder and used to design a workflow for electrolyte synthesis and testing.

### BatteryRobot

Class for North’s robot arm. Child class of NorthC9.

### T8

Class for North’s heatplate. Child class of NorthC9, same network as BatteryRobot.

### PowderShaker

Class for North’s powder dispenser. Child class of NorthC9, same network as BatteryRobot.

### PStat (Todo)

Class for Potentiostat. Class not implemented, but methods are. Available in utils/PStat

## Gamry

Stuff about Gamry goes here.

## Powder Protocols

All powder protocols are defined in MainProject\src\main\settings\powder\_protocols.py.

## External Libraries

Pandas

Gamry Python API

## Errors Raised for Safety Purposes

Some Exceptions (errors) are raised during runtime to avoid collision between components of the robot.

# General Limitations and Warnings

## Preventing Vial and Pipette Collisions

Whenever a vial is being held by the robot arm, pipettes cannot be held, nor can any processes requiring pipettes be executed, and vice versa. This is to prevent the collision of the vial/pipette with other deck components. The holding of pipettes and vials are tracked by the robot’s attributes: **holding\_pipette** and **holding\_vial**.

## Pipette Collection Order

Pipettes must be collected in specific order:

1. Start from bottom of rightmost column
2. Collect pipettes up this column
3. Proceed to the bottom of the adjacent column
4. Repeat

The order, structured as a Python array (zero-indexed), is as follows:  
  
pipette\_order = [i for i in range(2, 48, 3)] + [i for i in range(1, 48, 3)] + [i for i in range(0, 48, 3)]

This order of pipettes must be used to get pipettes from the pipette grid as specified in the **pipette\_grid** locator.

## Moving Robot Arm Around Deck

In general, **go\_to\_safe()** should be used when moving the robot arm around the deck, as it moves the robot arm to the top of the deck, then to the correct x-and-y-coordinates and back down to the desired z-coordinate. This ensures the robot arm does not collide with deck components.

# Appendix 1: All Locators

The locators for the project are first recorded in main.nproj, which are then exported to Locator.py during runtime. To reference these locators simply import all from Locator.py located in the main folder.   
  
Gripper Locs

**vial\_carousel (standard) –** Locator used to move vial to position between clamps on carousel.

**carousel\_cap\_approach (standard) –** This locator is used when capping an open vial in the carousel. Robot gripper must be holding a vial cap, and carousel clamps must contain an open vial. This locator represents the position directly above vial that gripper must move the vial cap to in preparation for capping the vial.

**safe\_zone(standard) –** A general safe zone with no nearby objects for robot arm to rest when robot arm is not being used.

**screw\_1 (standard) –**

**screw\_2 (standard) –**

**microplate\_holder\_approach (standard) –** Position near screw that robot arm must go to before moving directly to the holder.

**microplate\_holder (standard) –** Screw that acts as a holder for gripper to hold on to when attempting to move microplate. Used after microplate\_holder\_approach

**lidholder\_holder (standard) –** Screw that acts as a holder for gripper to hold on to when attempting to move lidholder.

**microplate\_screw\_approach\_1 (standard) –** This locator is used when inserting a screw to the left tapped hole of the microplate. Robot gripper must be holding a screw. This locator represents the position directly above the left tapped hole of the microplate.

**microplate\_screw\_approach\_2 (standard) –** Similar to microplate\_screw\_approach\_1, except this locator represents the right tapped hole to screw into.

**powder\_1 (standard) –** Rightmost inactive powder cartridge. Locator represents part of cartridge that gripper grips onto to move cartridge around.

**powder\_2 (standard) –** Similar to powder\_1. Second-from-right inactive powder cartridge.

**powder\_3 (standard) –** Similar to powder\_1. Third-from-right inactive powder cartridge.

**powder\_4 (standard) –** Similar to powder\_1. Leftmost inactive powder cartridge.

**active\_powder\_cartridge (standard) –** Powder cartridge that is currently being used by the carousel for poweder dispensing. Locator represents part of cartridge that gripper grips onto to move cartridge around.

**rack\_official (grid) –** The main vial rack at the center of the deck. It is a 6x8 rack called rack\_official. However, the same physical rack on the deck is also split into two seprate grid locators: rack\_dispense\_official(below) and p\_aspirate\_low (see Pipette locs).

rack\_dispense\_official represents the upper 4 rows of the rack. This section of the rack is allocated to hold the vials that will be used to make solutions.

p\_aspirate\_low represents the lower 2 rows. This section is allocated to hold the solvents that will be used

to make the solutions contained in the vials of rack\_dispense\_official.

Even though they are part of the same physical 6x8, their indexing as Locators are independent of each other.

Meaning, rack\_dispense\_official has indexes from 0-31, where every 4 indexes is one of its columns,

and p\_aspirate\_low has indexes from 0-15, where every 2 indexes is one of its columns.

Layout (indexes) of rack\_dispense\_official:

(col 8) (col 1)

28 . . . 0

29 . . . 1

30 . . . 2

31 . . . 3

Layout (indexes) of p\_aspirate\_low:

(col 8) (col 1)

14 . . . 0

15 . . . 1

**rack\_dispense\_official (grid) –** Represents the upper 4 rows of the rack. This section of the rack is allocated to hold the vials that will be used to make solutions. See **rack\_official** above.

**rack\_support(grid) –** Supporting rack to rack\_official. This rack has an identical layout to rack\_official, just that it is at a different location on the deck

**heatplate\_official (grid) –** Represents the metal 3 by 4 heatplate located at the left side of the deck.

Pipette Locs

**p\_remover\_capture\_approach (standard) and p\_remover\_capture (standard) -**

When used pipette is being held by robot arm, these two locators used in succession moves the pipette to below the pipette remover to prepare for removal. Once robot arm moves to this point, it should move vertically up using **move\_z()** to remove the used pipette.   
  
Proper usage of the two locators (taken from **remove\_pipette()**)are as follows:

self.goto\_safe(p\_remover\_capture\_approach)

self.goto(p\_remover\_capture)

self.move\_z(400)

**carousel\_dispense (standard) –** moves pipette to prepare for dispensing liquids into a vial positioned between the two clamps of the carousel.

**p\_aspirate\_low (grid) –** Represents the lower 2 rows of rack\_official. This section is allocated to hold the solvents that will be used to make the solutions contained in the vials of rack\_dispense\_official. Used when volume of solvent in vial is low, as it moves pipette down to the bottom of the vial. See **rack\_official** above.

**p\_aspirate\_mid (grid) –** Similar to p\_aspirate\_low, except this is used when vial is about half-full with solvent. Pipette is positioned about halfway down the vial to minimize sorption of solvent to outside of pipette

**p\_aspirate\_high (grid) –** Similar to p\_aspirate\_low, except this is used when vial is about almost full with solvent. Pipette is positioned near top of vial to minimize sorption of solvent to outside of pipette

**pipette\_grid (grid) –** Rack holding all pipettes to be used by robot. See Limitations and Warnings on pipette collection.

**microplate\_official (grid) –** Grid representing the wells of the microplate. Pipette will be positioned at the center of each well

**microplate\_side\_official(grid) –** Grid representing the side of the wells of the microplate. Pipette will be positioned at the side of each well