

1

**Actuator Screen**

# Abstraction

**Actuator Screen** is a screen from the Ground Station desktop application to manage, monitor actuators and perform actions.

**Task Name**: Actuator screen

**Supervisor**: Mohammad Alkatib – Software Team Lead.

**Developed by**: Moad Jawabreh – Computer Engineering. **Documented by**: Moad Jawabreh.

**Technology stack:**

Programming Language: C#

WPF (Windows Presentation Foundation):

**Integrated Development Environments (IDEs):**

Visual Studio Code.

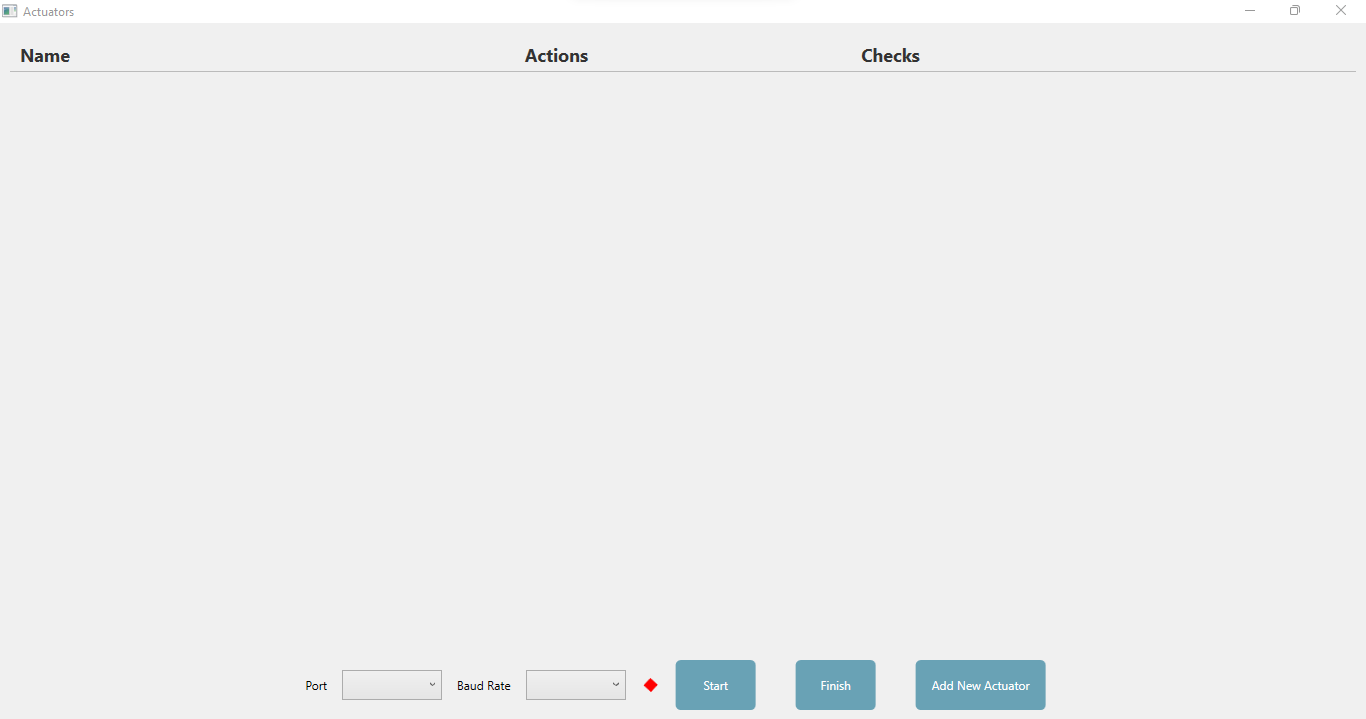
**Introduction:**

The Actuator Control System is a software application designed to manage and monitor actuators This system provides a user-friendly interface for users to interact with actuators, view their status, and perform actions such as enabling or disabling them. Every actuator has three actions (Up/Down/Close), (Left/Right/Close) and (Lock/Unlock/Close).

**Task:**

1. **Actuator screen**

Actuator Screen is the primary user interface component of the Actuator Control System. It provides users with a comprehensive view of all connected actuators and facilitates interaction with them.



**That’s the main window, it displays a few things:**

* **Table with columns for "Name", "Actions", and "Checks".**
* **Each row represents an actuator and its associated actions.**
* **The "Name" column**

displays the name of the actuator.

* **The "Actions" column**

displays the current action being performed.

* **The Checks column**

contains checkboxes that indicate completion and verification status.

* **Red Diamond**

to inquire about connection status or port opening.

* **Start Button**

Clicking this button initiates the action sequence for the selected actuator.

The button is disabled until an actuator is selected.

* **Finish Button**

Clicking this button exports the data to an Excel file.

The exported data includes actuator names, actions performed, and completion status.

* **Add New Actuator Button**

Opens a dialog window to add a new actuator and select its actions.

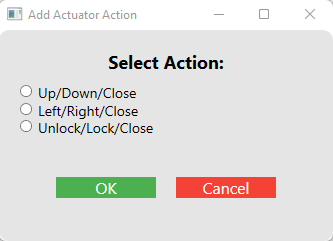
* **Port Selection Combobox**

Allows the user to select the serial port for communication.

* **Baud Rate Combobox**

Allows the user to select the baud rate for serial communication.

1. **Add Actuator Action Window**

****

* **Radio Buttons**

Allows the user to select a predefined set of actions for the new actuator.

Options include "Up & Down (Close)", "Left & Right (Close)", and "Unlock & Lock (Close)".

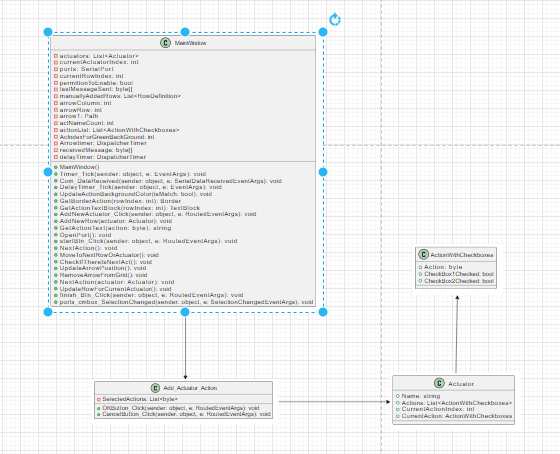
* **Ok Button**

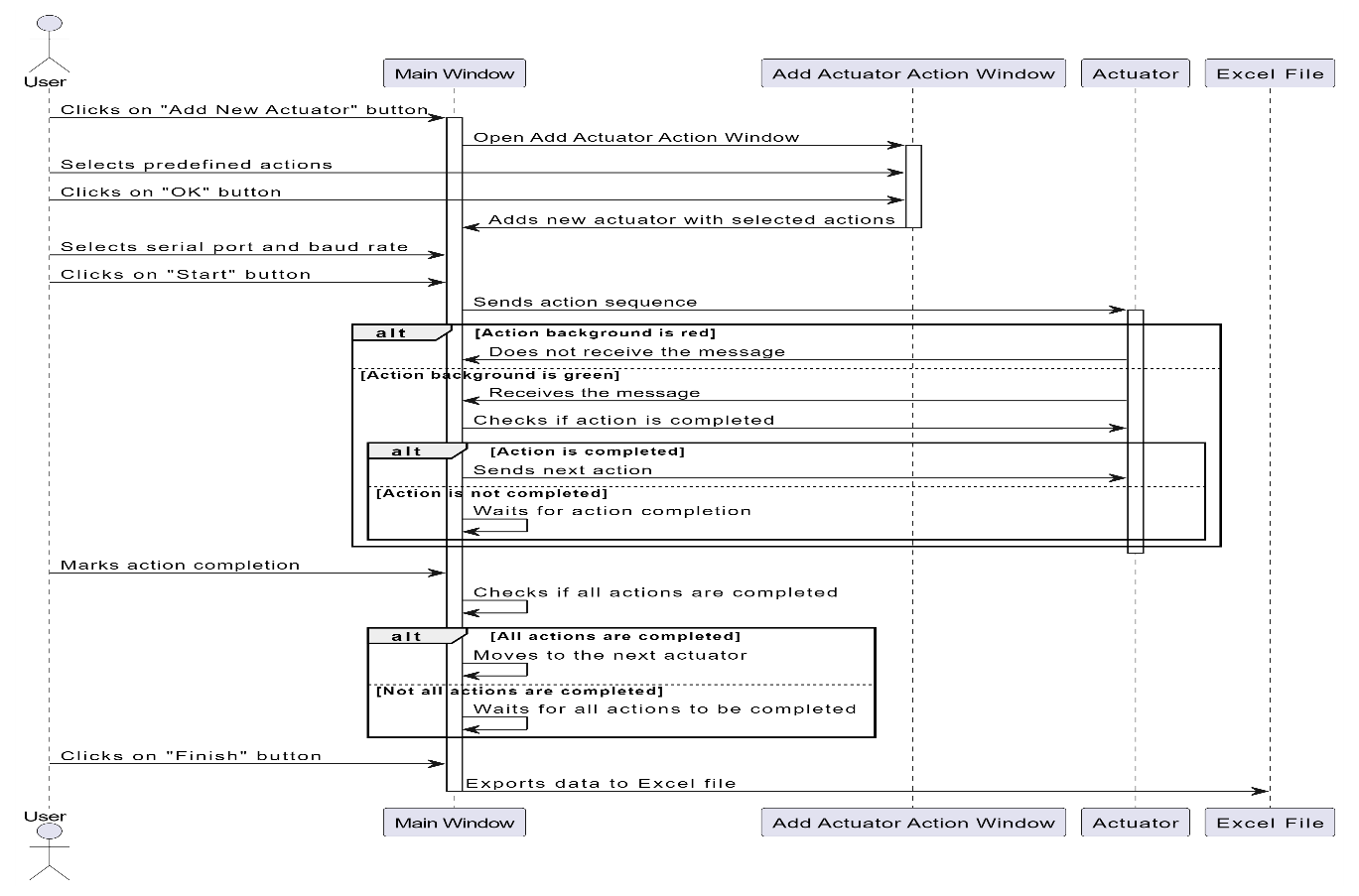
Confirms the selection and adds the new actuator with the chosen actions.

* **Cancel Button**

Cancels the operation and closes the window without adding a new actuator.

1. **Class diagram**

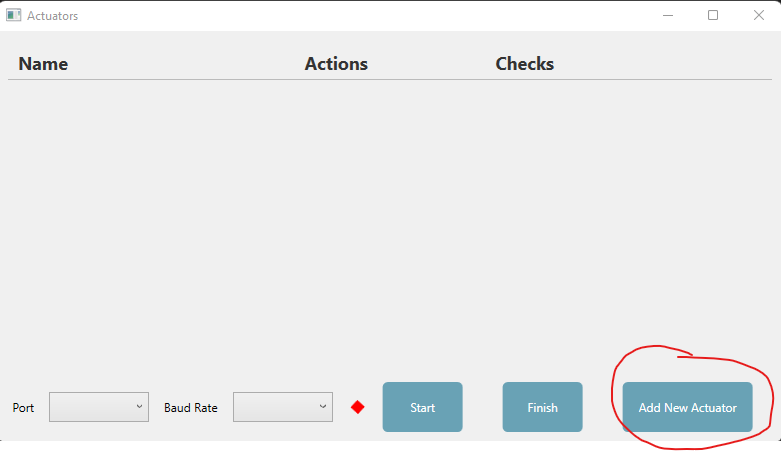
****



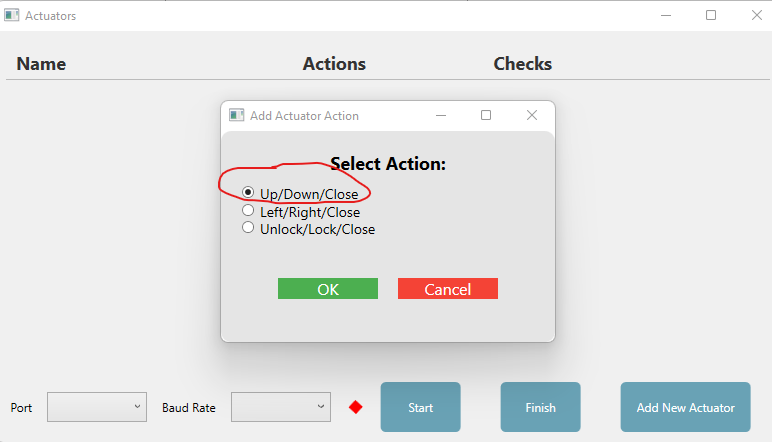
**Usage Instructions:**

**1-Adding a New Actuator**

Click the "Add New Actuator" button.

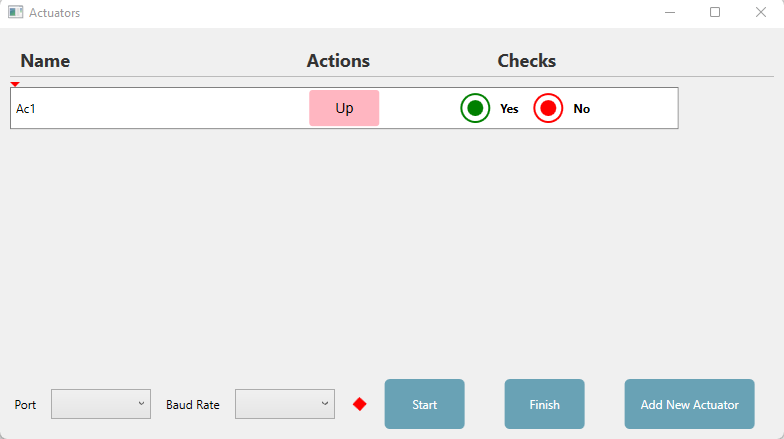
****

In the popup window, select the desired actions for the new actuator.

****

For example: here I chose UP/Down/Left.

Click the "OK" button to confirm and add the new actuator, or click "Cancel" to discard changes.

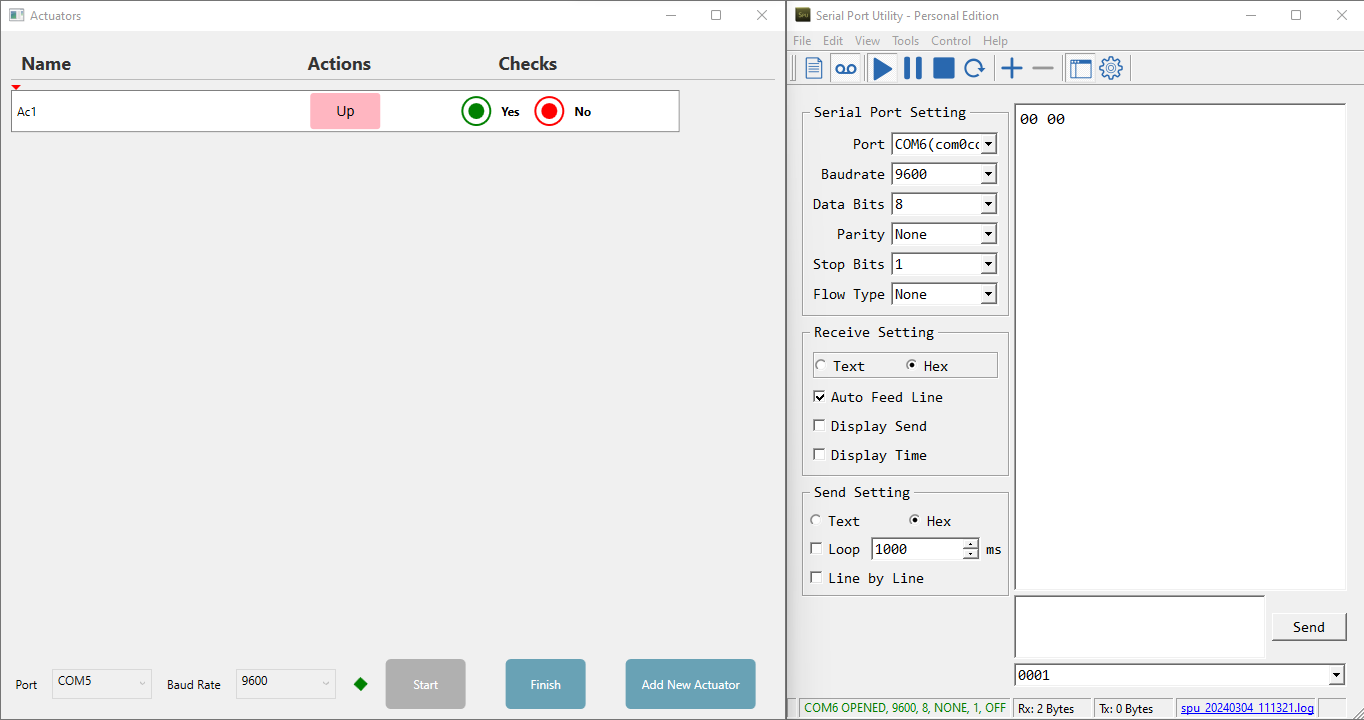
****

As we see here the row added here contains the actuator Name and the actions sequence and the Checks.

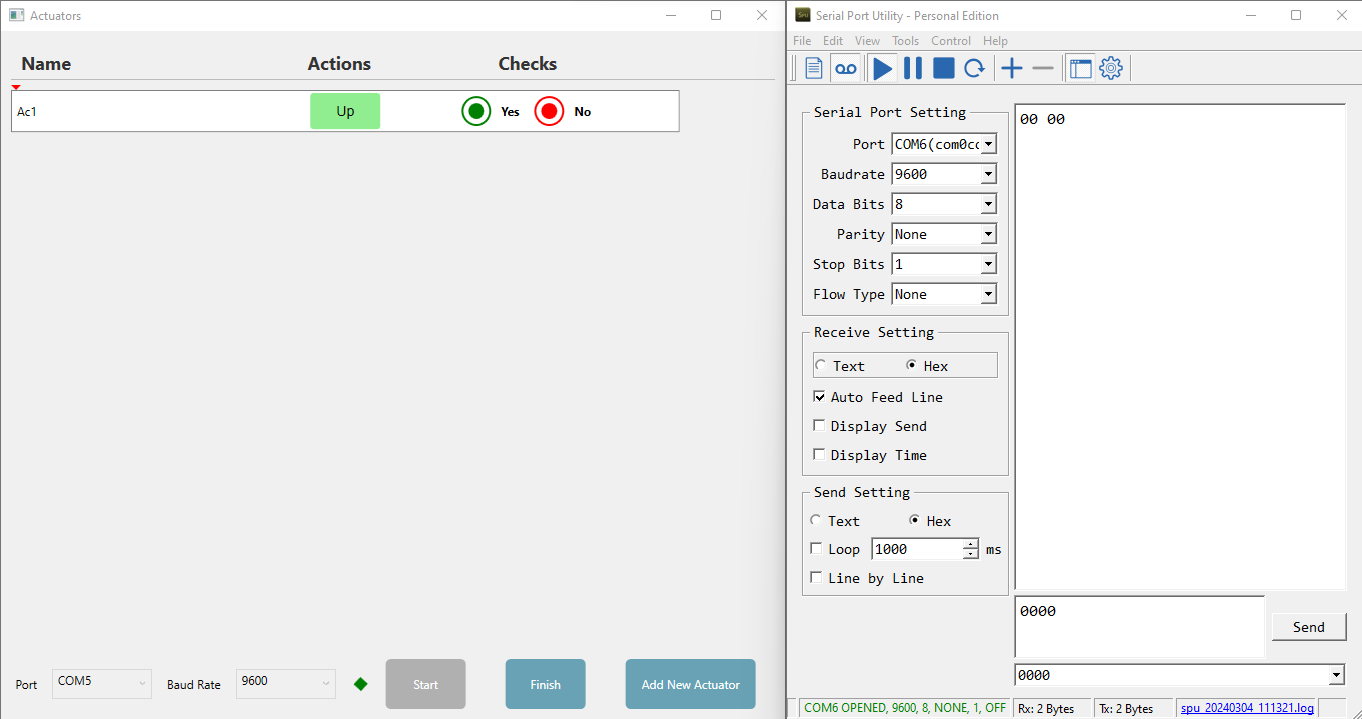
**Starting the Action Sequence**

Select a serial port and baud rate from the dropdown menus.

Click the "Start" button to begin the action sequence for the selected actuator.

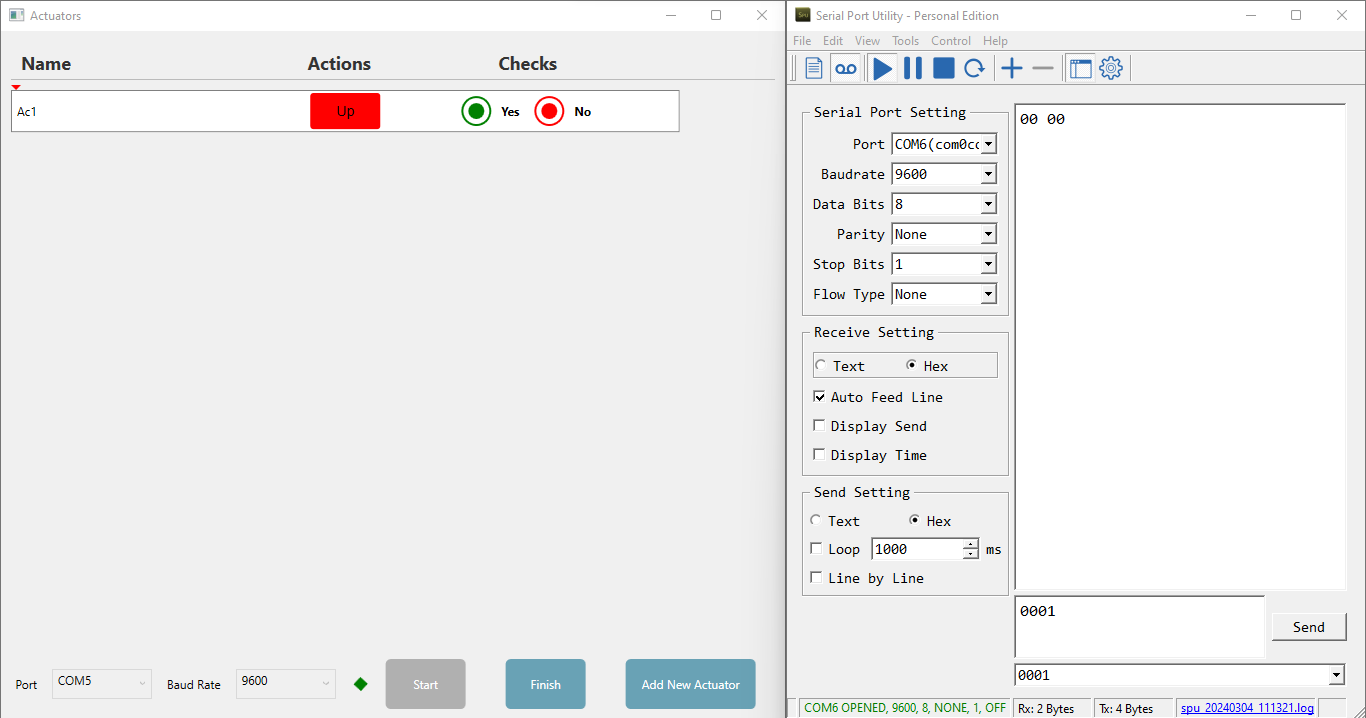
****

Here I clicked on the start button, the diamond will be green that’s mean the port is opening, it sent to another port 00 00, the first 00 is about the Ac Name (Id), the second 00 for the Action Id (Up), keep in mind that the background color is pink that’s mean that the Actuator didn’t receive any message, now if the Actuator receive the same message sent then the background color will be green:

****

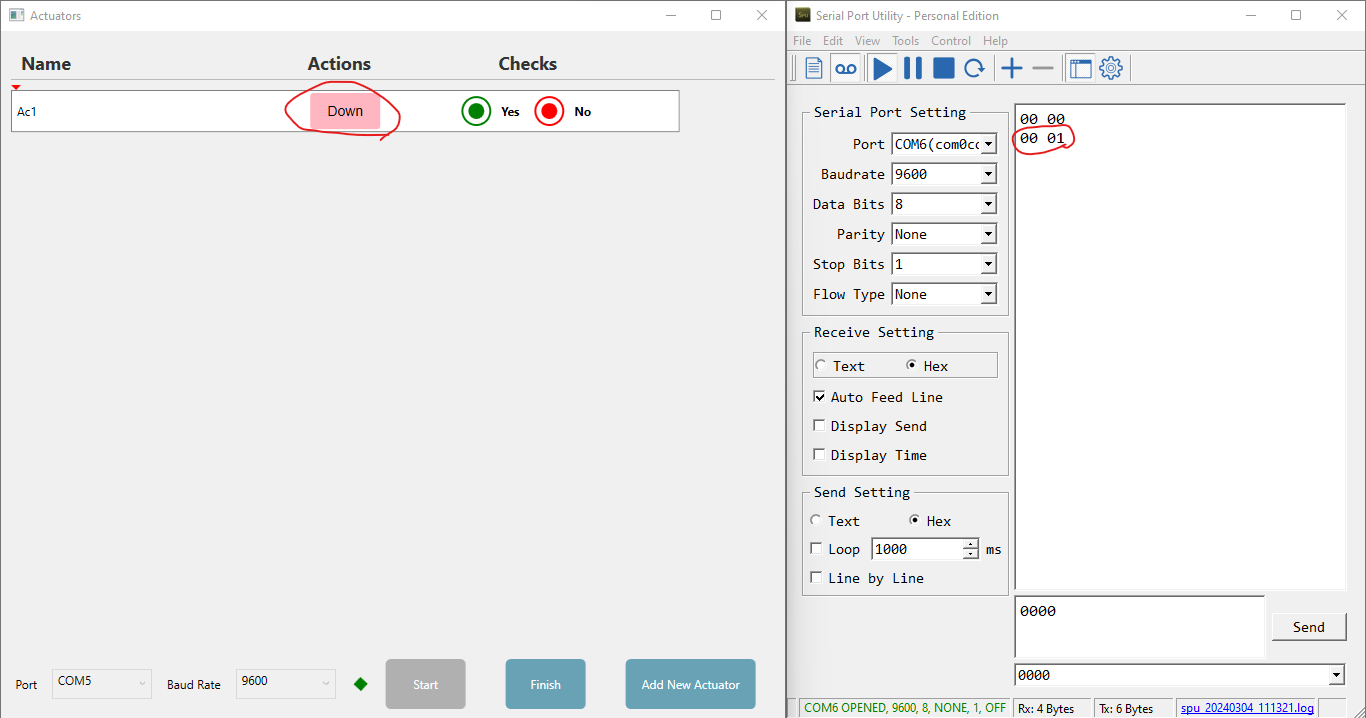
Now if you look here, you will see the action background is **green**, and the Radio Buttons is Enable now to check if the plane did what was required (UP), if yes, then we check on **green circle**, else we check on **red** circle**.**

Now if the Actuator does not receive the same message sent then the background color will be Red and the checks return to disable:

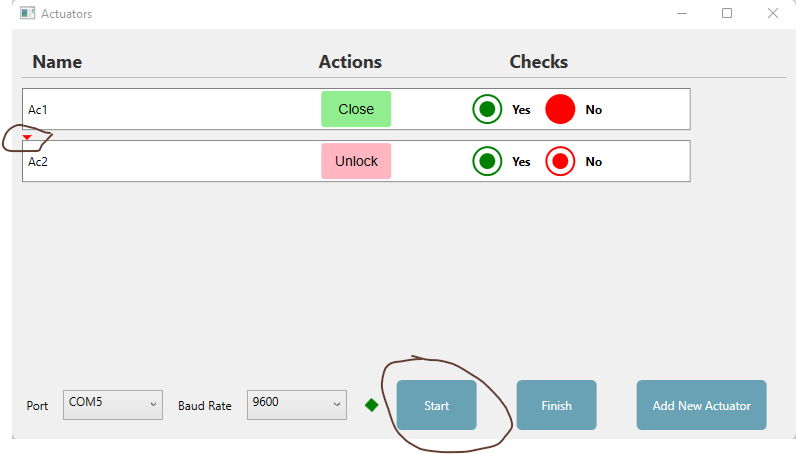
****

We cannot go through this Action until it receives the same message that it sent.

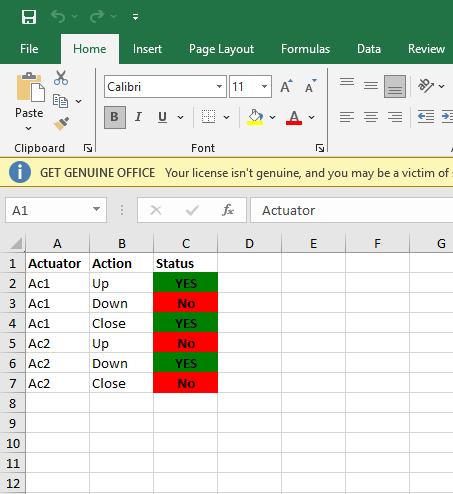
Now in the case that the Action background is green (message sent = message received), we have to check on the Radio button Based on completion of the required Action, if we check yes, after 2 seconds the action will change to another action and it will send the message for that another action (Down here), and so on, let’s take a look:

****

Until we finish the actions and arrive to Close, if there are more actuators, the arrow directly goes to the next actuator and the start button will be enable again to start the next Actuator:

****

Once all actions are completed or stopped, click the "Finish" button to export the data to an Excel file.



Ensure the correct serial port and baud rate are selected before starting the action sequence.