**Project Targets Document**

**Project Overview: Establishing a Python-PLC Communication Interface**

This project aims to establish a communication link between a Python script and a Programmable Logic Controller (PLC) using the pyads library.

The connection allows for issuing commands to control PLC operations and retrieve status information in real-time.

**Establishing a Connection with the PLC**

**Objective:**

* Create a connection using the provided AMS Net ID and port number.
* Open the connection to enable communication with the PLC.

**Parameters:**

1. **amc\_id:** The AMS Net ID of the PLC device (e.g., "169.254.2.227.1.1").
2. **pyads.PORT\_TC3PLC1:** The port number used for the first PLC in TwinCAT (usually 851).

**Return:** Once the connection is established, various control and monitoring functions can be executed. A confirmation message is printed if the connection is successful. If the connection fails, an error is raised.

There are two types of function:

* **Query Functions :**
  1. get\_device\_info
  2. get\_all\_parameters
* **Command Functions** :
  1. move\_single\_axis\_to\_location
  2. begin\_home\_operation
  3. stop\_move
  4. clear\_one\_axis
  5. clear\_all\_axes

**1.** **get\_device\_info(plc)**

**Objective:**

* Send a message with the action code to retrieve device information.
* Retrieve basic information about the PLC device in an array format.

**Parameters:**

1. **plc:** The PLC connection used to communicate with the device.

**Action Code:** 500 (represented by ACTIONCODE\_GET\_DEVICE\_INFO in action\_code.py).

**Return:** The function returns True along with an array of device information if successful. The device information includes:

1. Device ID
2. Number of Real Axes
3. Number of Digital Inputs (DI)
4. Number of Digital Outputs (DO)
5. Number of Virtual Axes
6. Communication Port Number
7. Diagnosis Port Number

If the command fails or the response is invalid, False is returned.

**2. get\_all\_parameters(plc)**

**Objective:**This function sends a request to the PLC using the action code ACTIONCODE\_GET\_ALL\_PARAMETERS to retrieve all parameters related to the device's axes, including their status and positions.

**Parameters:**

1. **plc:** The PLC connection object used to communicate with the device.

**Action Code:**502 (represented by ACTIONCODE\_GET\_ALL\_PARAMETERS).

**Return:**The function successfully returns the following:

**True** if the parameters were successfully retrieved, along with access to:

* 1. The number of axes available on the device.
  2. Status and position of each axis.

**Accessing Axis Information**: You can access the status and position of individual axes using the following methods:

1. **index 0** getXaxis()
2. **index 1** getYaxis()
3. **index 2** getZaxis()
4. **index 3** getTaxis()

**Axis Information Breakdown:**

Each element of the retrieved information corresponds to a specific axis and provides:

**Status of the Axis:**

* 1. **Moving:** is\_moving()
  2. **Ready:** is\_ready()
  3. **Forward Limit ON:** forward\_limit\_on()
  4. **Reverse Limit ON:** reverse\_limit\_on()
  5. **Homed:** is\_homed()
  6. **Movable from PC:** is\_movable\_from\_pc()
  7. **Emergency ON:** is\_emergency\_on()

**Position of the Axis:** The current position of each axis can be accessed using the get\_position() function.

**False** If the command fails or if an invalid response is received, the function returns False.

**3. move\_single\_axis\_to\_location(plc, axisIndex, location, speed, acceleration, deceleration)**

**Objective:**This function sends the move command to the specified axis with the defined parameters, including the target location, speed, acceleration, and deceleration.

**Parameters:**

1. **plc:** The PLC connection used for communication with the device.
2. **axisIndex**: The index of the axis to be moved.
3. **location:** Target location in motion units.
4. **speed:** Speed of the movement.
5. **acceleration:** Rate of acceleration.
6. **deceleration:** Rate of deceleration.

**Action Code:**5020 (represented by ACTIONCODE\_MOVESINGLEAXIS).

**Return:**

* If the operation status is not executed (ACTIONSTATUS\_EXECUTED or ACTIONSTATUS\_EXECUTEDWITHWARNING) or if the action code in the response does not match ACTIONCODE\_MOVESINGLEAXIS, the function returns:
  1. **False**
  2. **Warning Code:** Indicates any issues encountered during the operation.
* If the operation is successful, the function returns:
  1. **True**
  2. **Common Response:** Confirmation code from the PLC indicating the status of the command.
  3. **Warning Code:** Indicates any issues encountered during the operation.

**4. begin\_home\_operation(plc, axisIndex)**

**Objective:**Starts the homing operation for a specific axis. Homing is a procedure that returns the axis to a predefined starting position.

**Parameters:**

1. **plc:** The PLC connection object.
2. **axisIndex:** The index of the axis to be homed.

**Action Code:**5023 (represented by ACTIONCODE\_BEGIN\_HOME\_OPERATION).

**Return:**

* If the operation status is not executed (ACTIONSTATUS\_EXECUTED or ACTIONSTATUS\_EXECUTEDWITHWARNING) or if the action code in the response does not match ACTIONCODE\_BEGIN\_HOME\_OPERATION, the function returns:
  1. **False**
  2. **Warning Code:** Indicates any issues encountered during the homing operation.
* If the operation is successful, the function returns:
  1. **True**
  2. **Common Response:** Confirmation code from the PLC indicating the status of the homing command.
  3. **Warning Code:** Indicates any issues encountered during the homing operation.

**5. stop\_move(plc, axisIndex, deceleration)**

**Objective:**Stops the motion of a specified axis with the given deceleration rate.

**Parameters:**

1. **plc:** The PLC connection object.
2. **axisIndex:** The index of the axis to be stopped.
3. **deceleration:** Deceleration rate.

**Action Code:**5024 (represented by ACTIONCODE\_STOP\_MOVE).

**Return:**

* If the operation status is not executed (ACTIONSTATUS\_EXECUTED or ACTIONSTATUS\_EXECUTEDWITHWARNING) or if the action code in the response does not match ACTIONCODE\_STOP\_MOVE, the function returns:
  1. **False**
  2. **Warning Code:** Indicates any issues encountered during the stop operation.
* If the operation is successful, the function returns:
  1. **True**
  2. **Common Response:** Confirmation code from the PLC indicating the status of the stop command.
  3. **Warning Code:** Indicates any issues encountered during the stop operation.

**6. clear\_one\_axis(plc, axisIndex)**

**Objective:**Sends a command to clear any faults or errors related to the specified axis.

**Parameters:**

1. **plc:** The PLC connection object.
2. **axisIndex:** The index of the axis to be cleared.

**Action Code:**5025 (represented by ACTIONCODE\_CLEAR\_ONE\_AXIS).

**Return:**

* If the operation status is not executed (ACTIONSTATUS\_EXECUTED or ACTIONSTATUS\_EXECUTEDWITHWARNING) or if the action code in the response does not match ACTIONCODE\_CLEAR\_ONE\_AXIS, the function returns:
  1. **False**
  2. **Warning Code:** Indicates any issues encountered during the clear operation.
* If the operation is successful, the function returns:
  1. **True**
  2. **Common Response:** Confirmation code from the PLC indicating the status of the clear command.
  3. **Warning Code:** Indicates any issues encountered during the clear operation.

**7. clear\_all\_axes(plc)**

**Objective:**Clears the errors or faults for all axes in the system.

**Parameters:**

1. **plc:** The PLC connection object.

**Action Code:**5027 (represented by ACTIONCODE\_CLEAR\_ALL\_AXES).

**Return:**

* If the operation status is not executed (ACTIONSTATUS\_EXECUTED or ACTIONSTATUS\_EXECUTEDWITHWARNING) or if the action code in the response does not match ACTIONCODE\_CLEAR\_ALL\_AXES, the function returns:
  1. **False**
  2. **Warning Code:** Indicates any issues encountered during the clear operation for all axes.
* If the operation is successful, the function returns:
  1. **True**
  2. **Common Response:** Confirmation code from the PLC indicating the status of the clear command.
  3. **Warning Code:** Indicates any issues encountered during the clear operation for all axes