

OpenPLC Configuration Guide for Intralogistics Simulator

Overview

This guide covers the configuration and setup of OpenPLC webserver for use with the Intralogistics Simulator, including structured text variable definitions, Python SubModule (PSM) integration, and Modbus connection setup.

Table of Contents

- 1. [Initial Setup](#)
- 2. [Structured Text Configuration](#)
- 3. [PSM Hardware Layer](#)
- 4. [Modbus Connection Setup](#)
- 5. [Signal Dashboard Configuration](#)

Initial Setup

- 1. Access the OpenPLC webserver interface
- 2. Navigate to the Programs section
- 3. Ensure you have the following files ready:
 - Intralogistics Variables structured text file
 - PSM hardware layer script

<>

Running: Intralogistics Variables

OpenPLC User

Dashboard

Programs

Slave Devices

Monitoring

Hardware

Users

Settings

Logout

Programs

Here you can upload a new program to OpenPLC or revert back to a previous uploaded program shown on the table.

Program Name	File	Date Uploaded
Intralogistics Variables	280445.st	Feb 20, 2025 - 05:07PM
Blank Program	blank_program.st	May 24, 2018 - 06:02PM

List all programs

Upload Program

Seleccionar archivo Sin archivos seleccionados Upload Program

Structured Text Configuration

The structured text program defines all PLC variables used by the simulator. Upload the `intralogistics_variables.st` file which contains:

```
PROGRAM prog0
VAR
  (* Digital Inputs - Read Only *)
  PLC_CYCLE_STOPPED AT %IX0.0 : BOOL;
  PLC_CYCLE_RUNNING AT %IX0.1 : BOOL;
  PICK_ERROR AT %IX0.2 : BOOL;
  // ... other inputs

  (* Digital Outputs for Bin Selection *)
  PICK_BIN_01 AT %QX0.0 : BOOL;
  PICK_BIN_02 AT %QX0.1 : BOOL;
  // ... other bin selections

  (* Station Selection Outputs *)
  TO_RECEIVING_STA_1 AT %QX4.0 : BOOL;
  FROM_RECEIVING AT %QX4.1 : BOOL;
  TO_ASSEMBLY_STA_1 AT %QX4.2 : BOOL;
  FROM_ASSEMBLY AT %QX4.3 : BOOL;
END_VAR
END_PROGRAM
```

These variable definitions map directly to Modbus addresses and are monitored in the OpenPLC interface:

Running: Intralogistics Variables
OpenPLC User

Dashboard

Programs

Slave Devices

Monitoring

Hardware

Users

Settings

Logout

Status: Running

Stop PLC

Monitoring

Refresh Rate (ms):

Update

Point Name	Type	Location	Write	Value
PLC_CYCLE_STOPPED	BOOL	%IX0.0		<div><div style="width: 100%; background-color: #28a745;"></div></div> TRUE
PLC_CYCLE_RUNNING	BOOL	%IX0.1		<div><div style="width: 0%; background-color: #dc3545;"></div></div> FALSE
PICK_ERROR	BOOL	%IX0.2		<div><div style="width: 0%; background-color: #dc3545;"></div></div> FALSE
PICK_TO_ASSEMBLY_IN_PROCESS	BOOL	%IX0.3		<div><div style="width: 0%; background-color: #dc3545;"></div></div> FALSE
PICK_TO_ASSEMBLY_COMPLETE	BOOL	%IX0.4		<div><div style="width: 0%; background-color: #dc3545;"></div></div> FALSE
PICK_TO_RECEIVING_IN_PROCESS	BOOL	%IX0.5		<div><div style="width: 0%; background-color: #dc3545;"></div></div> FALSE
PICK_TO_RECEIVING_COMPLETE	BOOL	%IX0.6		<div><div style="width: 0%; background-color: #dc3545;"></div></div> FALSE
PICK_TO_WAREHOUSE_IN_PROCESS	BOOL	%IX0.7		<div><div style="width: 0%; background-color: #dc3545;"></div></div> FALSE
PICK_TO_WAREHOUSE_COMPLETE	BOOL	%IX1.0		<div><div style="width: 0%; background-color: #dc3545;"></div></div> FALSE
PICK_BIN_01	BOOL	%QX0.0	<div> <div style="background-color: #007bff; color: white; padding: 2px 5px; font-size: 0.8em;">true</div> <div style="border: 1px solid #ccc; padding: 2px 5px; font-size: 0.8em;">false</div> </div>	<div><div style="width: 0%; background-color: #dc3545;"></div></div> FALSE
PICK_BIN_02	BOOL	%QX0.1	<div> <div style="background-color: #007bff; color: white; padding: 2px 5px; font-size: 0.8em;">true</div> <div style="border: 1px solid #ccc; padding: 2px 5px; font-size: 0.8em;">false</div> </div>	<div><div style="width: 0%; background-color: #dc3545;"></div></div> FALSE
PICK_BIN_03	BOOL	%QX0.2	<div> <div style="background-color: #007bff; color: white; padding: 2px 5px; font-size: 0.8em;">true</div> <div style="border: 1px solid #ccc; padding: 2px 5px; font-size: 0.8em;">false</div> </div>	<div><div style="width: 0%; background-color: #dc3545;"></div></div> FALSE
PICK_BIN_04	BOOL	%QX0.3	<div> <div style="background-color: #007bff; color: white; padding: 2px 5px; font-size: 0.8em;">true</div> <div style="border: 1px solid #ccc; padding: 2px 5px; font-size: 0.8em;">false</div> </div>	<div><div style="width: 0%; background-color: #dc3545;"></div></div> FALSE
PICK_BIN_05	BOOL	%QX0.4	<div> <div style="background-color: #007bff; color: white; padding: 2px 5px; font-size: 0.8em;">true</div> <div style="border: 1px solid #ccc; padding: 2px 5px; font-size: 0.8em;">false</div> </div>	<div><div style="width: 0%; background-color: #dc3545;"></div></div> FALSE
PICK_BIN_06	BOOL	%QX0.5	<div> <div style="background-color: #007bff; color: white; padding: 2px 5px; font-size: 0.8em;">true</div> <div style="border: 1px solid #ccc; padding: 2px 5px; font-size: 0.8em;">false</div> </div>	<div><div style="width: 0%; background-color: #dc3545;"></div></div> FALSE
PICK_BIN_07	BOOL	%QX0.6	<div> <div style="background-color: #007bff; color: white; padding: 2px 5px; font-size: 0.8em;">true</div> <div style="border: 1px solid #ccc; padding: 2px 5px; font-size: 0.8em;">false</div> </div>	<div><div style="width: 0%; background-color: #dc3545;"></div></div> FALSE
PICK_BIN_08	BOOL	%QX0.7	<div> <div style="background-color: #007bff; color: white; padding: 2px 5px; font-size: 0.8em;">true</div> <div style="border: 1px solid #ccc; padding: 2px 5px; font-size: 0.8em;">false</div> </div>	<div><div style="width: 0%; background-color: #dc3545;"></div></div> FALSE

PSM Hardware Layer

1. Navigate to the Hardware section in OpenPLC
2. Select "Python on Linux (PSM)" from the hardware layer dropdown
3. Configure the PSM script that handles I/O operations:

<>

Running: Intralogistics Variables

OpenPLC User

Dashboard

Programs

Slave Devices

Monitoring

Hardware

Users

Settings

Logout

Status: *Running*

Stop PLC

Hardware

OpenPLC controls inputs and outputs through a piece of code called hardware layer (also known as driver). Therefore, to properly handle the inputs and outputs of your board, you must select the appropriate hardware layer for it. The Blank hardware layer is the default option on OpenPLC, which provides no support for native inputs and outputs.

OpenPLC Hardware Layer

Python on Linux (PSM)

OpenPLC Python SubModule (PSM)

PSM is a powerful bridge that connects OpenPLC core to Python. You can use PSM to write your own OpenPLC driver in pure Python. See below for a sample driver that switches %IX0.0 every second

```
1 import psm # type: ignore
2 import time
3 import random
4
5 # Global state variables
6
7
8 class PLCState:
9     def __init__(self):
10         self.cycle_running: bool = False
11         self.current_operation: str | None = None
12         self.operation_start_time: float = 0.0
13         self.error_state: bool = False
14         self.selected_bin: int | None = None
15
16
17 # Operation timeouts in seconds
18 OPERATION_TIMEOUT = 5 # How long an operation takes
19 ERROR_CHANCE = 0.05 # 5% chance of error per operation
20
21
22 def hardware_init():
23     """Initialize the PSM hardware layer"""
24     print("\n Initializing Beachside PLC simulator...")
25     psm.start()
```

Save changes

Restore Original Code

The PSM script provides:

- Real-time I/O handling
- Operation state management
- Error detection
- Signal monitoring

Modbus Connection Setup

1. Access the EpiBus interface
2. Create a new Modbus Connection with these settings:
 - Device Type: PLC
 - Host: openplc
 - Port: 502

EpiBus > Modbus Connection > US15-B10-B1-PLC

Ask Vaal (Enterprise PrAsk Vaal

Buscar o escribir un comando (⌘ + G)

Ayuda

US15-B10-B1-PLCHabilitado

Acciones <>⌂...Guardar

Ask about th

U

Asignado a +

Adjuntos +

Etiquetas +

Compartir +

0 · 0

SEGUIR

Usted editó esto por última vez · Hace 56 minutos

Usted creó este · Hace 1 semana

Device Type *
PLC
AI Assist

Host *
openplc
AI Assist

Habilitado

Puerto *
502

Miniatura
Adjuntar

Signals

	Nº	Signal Name *	Signal Type *	Modbus Address *	Analog Value	Digital Val...	PLC Addre...	
<input type="checkbox"/>	1	PICK BIN 01	Digital Output Coil	0	0	<input type="checkbox"/>	%QX0.0	
<input type="checkbox"/>	2	PICK BIN 02	Digital Output Coil	1	0	<input type="checkbox"/>	%QX0.1	
<input type="checkbox"/>	3	PICK BIN 03	Digital Output Coil	2	0	<input type="checkbox"/>	%QX0.2	
<input type="checkbox"/>	4	PICK BIN 04	Digital Output Coil	3	0	<input type="checkbox"/>	%QX0.3	
<input type="checkbox"/>	5	PICK BIN 05	Digital Output Coil	4	0	<input type="checkbox"/>	%QX0.4	
<input type="checkbox"/>	6	PICK BIN 06	Digital Output Coil	5	0	<input type="checkbox"/>	%QX0.5	
<input type="checkbox"/>	7	PICK BIN 07	Digital Output Coil	6	0	<input type="checkbox"/>	%QX0.6	
<input type="checkbox"/>	8	PICK BIN 08	Digital Output Coil	7	0	<input type="checkbox"/>	%QX0.7	
<input type="checkbox"/>	9	PICK BIN 09	Digital Output Coil	8	0	<input type="checkbox"/>	%QX1.0	
<input type="checkbox"/>	10	PICK BIN 10	Digital Output Coil	9	0	<input type="checkbox"/>	%QX1.1	
<input type="checkbox"/>	11	PICK BIN 11	Digital Output Coil	10	0	<input type="checkbox"/>	%QX1.2	
<input type="checkbox"/>	12	PICK BIN 12	Digital Output Coil	11	0	<input type="checkbox"/>	%QX1.3	
<input type="checkbox"/>	13	TO RECEIVING STA 1	Digital Output Coil	32	0	<input type="checkbox"/>	%QX4.0	
<input type="checkbox"/>	14	FROM RECEIVING STA 1	Digital Output Coil	33	0	<input type="checkbox"/>	%QX4.1	
<input type="checkbox"/>	15	TO ASSEMBLY STA 1	Digital Output Coil	34	0	<input type="checkbox"/>	%QX4.2	
<input type="checkbox"/>	16	FROM ASSEMBLY STA 2	Digital Output Coil	35	0	<input type="checkbox"/>	%QX4.3	
<input type="checkbox"/>	17	PLC_CYCLE_STOPPED	Digital Input Contact	0	0	<input checked="" type="checkbox"/>	%IX0.0	
<input type="checkbox"/>	18	PLC_CYCLE_RUNNING	Digital Input Contact	1	0	<input type="checkbox"/>	%IX0.1	
<input type="checkbox"/>	19	PICK_ERROR	Digital Input Contact	2	0	<input type="checkbox"/>	%IX0.2	
<input type="checkbox"/>	20	PICK_TO_ASSEMBLY_IN....	Digital Input Contact	3	0	<input type="checkbox"/>	%IX0.3	

3. Configure the signal mappings:

- Bin selection signals (Digital Output Coils)
- Station selection signals (Digital Output Coils)
- Status signals (Digital Input Contacts)


Signal Dashboard Configuration

The Modbus Signal Dashboard provides real-time monitoring of all signals:

Modbus Signal Dashboard

All Device Types

All Signal Types

 **US15-B10-B1-PLC**

Enabled

PLC

Host: openplc:502

Status	Signal Name	Type	Value	Modbus Address	PLC Address
●	PICK BIN 01	Digital Output Coil	⏻	0	%QX0.0
●	PICK BIN 02	Digital Output Coil	⏻	1	%QX0.1
●	PICK BIN 03	Digital Output Coil	⏻	2	%QX0.2
●	PICK BIN 04	Digital Output Coil	⏻	3	%QX0.3
●	PICK BIN 05	Digital Output Coil	⏻	4	%QX0.4
●	PICK BIN 06	Digital Output Coil	⏻	5	%QX0.5
●	PICK BIN 07	Digital Output Coil	⏻	6	%QX0.6
●	PICK BIN 08	Digital Output Coil	⏻	7	%QX0.7
●	PICK BIN 09	Digital Output Coil	⏻	8	%QX1.0
●	PICK BIN 10	Digital Output Coil	⏻	9	%QX1.1
●	PICK BIN 11	Digital Output Coil	⏻	10	%QX1.2
●	PICK BIN 12	Digital Output Coil	⏻	11	%QX1.3
●	FROM ASSEMBLY STA 2	Digital Output Coil	⏻	35	%QX4.3
●	FROM RECEIVING STA 1	Digital Output Coil	⏻	33	%QX4.1
●	TO ASSEMBLY STA 1	Digital Output Coil	⏻	34	%QX4.2

Key features:

- Filter by device type and signal type
- Real-time value updates
- Signal status indicators
- PLC address mapping display

Signal Types

- Digital Output Coils (QX): Control signals for bin and station selection
- Digital Input Contacts (IX): Status and feedback signals
- Addresses follow IEC 61131-3 conventions

Testing the Configuration

1. Start the OpenPLC runtime
2. Monitor the signal dashboard for activity
3. Test basic operations:
 - Bin selection
 - Station movement commands
 - Status feedback

Look for the green "Running" indicator in the OpenPLC interface to confirm proper operation.

Troubleshooting

Common issues and solutions:

1. Connection Errors
 - Verify openplc hostname resolution
 - Check port 502 accessibility
 - Confirm network configuration
2. Signal Mapping Issues
 - Verify address alignment between ST and Modbus
 - Check signal type configuration
 - Confirm proper data types
3. Runtime Errors
 - Monitor PSM script logs
 - Check OpenPLC error messages
 - Verify hardware layer selection

Additional Resources

- OpenPLC Documentation: <http://www.openplcproject.com/docs>
- EpiBus Documentation: <https://github.com/appliedrelevance/epibus>
- Reference Implementation: View the example configuration files in the project repository