

B03 — PV array with MPPT tracking

FMU model and guide for PLC programming

Version 1.0

Student exercise — Homa Virtual Plant (Model-in-the-Loop)

Homa inženjering

www.homa.hr · info@homa.hr

© 2026 Homa inženjering. All rights reserved.

Contents

1. Exercise objective.....	3
2. Process description (what the model represents).....	3
3. FMU interface — signals.....	3
3.1. FMU inputs.....	3
3.2. FMU outputs (measurements the PLC reads).....	3
4. What the PLC must achieve (control objectives).....	4
5. Suggested control structure (guideline).....	4
6. Recommended sample times.....	4
7. Test scenarios and acceptance criteria.....	4
8. Integration into Homa Virtual Plant.....	5
9. Notes on the model.....	5

1. Exercise objective

You are given an **FMU model** representing a photovoltaic (PV) array with a converter. Your task is, through **Homa Virtual Plant** to connect a PLC to this model and write a **PLC program that controls the process** according to the given objectives.

This is a **Model-in-the-Loop (MIL)** setup: the model mimics a real plant, your PLC drives it, and you develop and test your program on a virtual system. The model is “open” — it contains no controller of its own; **all control is performed by your PLC** through the model inputs.

2. Process description (what the model represents)

The PV array converts solar irradiance (`u_irradiance`) into DC power. The power depends on the irradiance and module temperature (`u_module_temperature`).

Available / in the process:

- **P-V curve** has a peak — the maximum-power point at the optimal voltage (`Vmp`).
- **MPPT task**: the PLC moves the DC-voltage reference (`u_dc_voltage_reference`) to find and hold that peak.
- **The fast current loop** stays in the converter (not part of this exercise).

3. FMU interface — signals

This is the most important part. Pay attention to the signal direction:

Signal direction: what is an **INPUT** for the FMU is an **OUTPUT** (command) for your PLC, and vice versa. The actuators your PLC **writes**; the measurements your PLC **reads**.

3.1. FMU inputs

a) Actuators — controlled by your PLC (PLC outputs):

Variable (FMU)	Type	Unit	Range	Description
<code>u_dc_voltage_reference</code>	Real	V	~50 – 400	DC-voltage reference (MPPT actuator)
<code>u_enable</code>	Real (0/1)	–	0 or 1	Converter enable

b) Disturbances / scenario — the environment, NOT controlled by your PLC:

Variable (FMU)	Type	Unit	Range	Description
<code>u_irradiance</code>	Real	W/m ²	0 – ~1000	Irradiance (sun)
<code>u_module_temperature</code>	Real	K	~270 – 330	Module temperature

You set the disturbances on the **PLC HMI** (as a parameter / simulated sensor), and the PLC forwards them to the FMU. They are not an output of your controller.

3.2. FMU outputs (measurements the PLC reads)

Variable (FMU)	Type	Unit	Range	Description
y_dc_voltage	Real	V	~1 – 400	DC voltage (follows the reference)
y_dc_current	Real	A	0 – ...	DC current
y_dc_power	Real	W	0 – ...	DC power (the quantity you maximize)
y_estimated_vmp	Real	V	~50 – 300	Estimated maximum-power voltage

4. What the PLC must achieve (control objectives)

Your PLC program must:

1. Find and hold the maximum-power point — maximize y_dc_power by varying u_dc_voltage_reference.
2. Track changes in irradiance and temperature (Vmp shift) without losing the peak.
3. Limit the perturbation step to avoid oscillation around the peak.
4. Safely connect/disconnect (u_enable) and react to sudden changes.

The objectives are given; **you design the control method yourself** (a suggested structure is provided below as an aid).

5. Suggested control structure (guideline)

MPPT algorithm (100 ms – 2 s):

- Perturb & Observe: change u_dc_voltage_reference in small steps; if the new y_dc_power is larger, continue in the same direction, otherwise reverse.
- Step size: a trade-off between speed and oscillation; reduce the step near the peak.

Protection / limits:

- Limit the voltage range; y_estimated_vmp use as a coarse guide.

6. Recommended sample times

Layer	Recommended cycle
MPPT algorithm	100 ms – 2 s
Protection / limits	~100 ms
FMU co-simulation communication step	~0.1 – 1 s

The model dynamics are fast (~0.2 s), so take a smaller communication step (e.g. 0.1 s) if you want to see the transients.

7. Test scenarios and acceptance criteria

Scenario	How to trigger	Expected PLC behaviour	Pass criterion
Irradiance ramp	u_irradiance 200 → 900	MPPT tracks the new peak	y_dc_power near the available maximum
Temperature ramp	u_module_temperature 300 → 320	adjust the voltage (Vmp drops)	no loss of the peak
Perturbation	small voltage steps	stable tracking	no persistent oscillation

Scenario	How to trigger	Expected PLC behaviour	Pass criterion
Connect/disconnect	u_enable 0 ↔ 1	safe start/stop	no jumps
Sensor noise	add noise on y_dc_power	MPPT robustness	stays at the peak

General criteria: controlled variables within limits, no instability / excessive cycling, correct response to faults.

8. Integration into Homa Virtual Plant

5. Load `Plant.fmu` (the compiled model) into Homa VP.
6. Map the FMU variables: **actuators** (`u_*`) as PLC outputs, **measurements** (`y_*`) as inputs.
7. Disturbances (`u_irradiance`, `u_module_temperature`) are set on the HMI as operating conditions and tests; the PLC forwards them to the FMU.
8. Select a communication channel (e.g. OPC UA / S7) and run the closed loop.

The model is configured for **StopTime 600 s** (Interval 1 s). Real-time pacing is driven by Homa VP via a speed factor.

9. Notes on the model

Important: the model is a simplified starter plant for developing and testing PLC logic — **it is not a validated industrial model**. The parameters are illustrative; calibrate them before real use.

- The dynamics are approximate; the purpose is a control exercise, not exact process analysis.
- The fast current loop (PWM) stays in the converter; here only the supervisory MPPT is done.