Modeling and Automation of Industrial Processes

MSc in Electrical and Computer Engineering
Scientific Area of Systems, Decision, and Control

*2022 / 2023*

Training Laboratory Work [1]

The main objective of this training laboratory is to allow the participants to get familiar with the Programmable Logic Controllers (PLCs) Schneider **Premium**, **TSX P57 1634M** or **2634M**, available in the laboratory (see Fig.1). The PLCs have digital input and output modules with the following references: **DEY16D2**, **DSY16T2**, **DMY28FK**. In particular are relevant: a) Using the software development system, *Unity Pro*, b) Setting the PLC initial configuration, c) Using the user manuals from Schneider, d) *Uploading / downloading* programs to/from the PLC, e) Editing and debugging solutions, f) *Interfacing* with external devices.

(a) Setup with one input and one output modules

(b) Setup based on a single input and output module
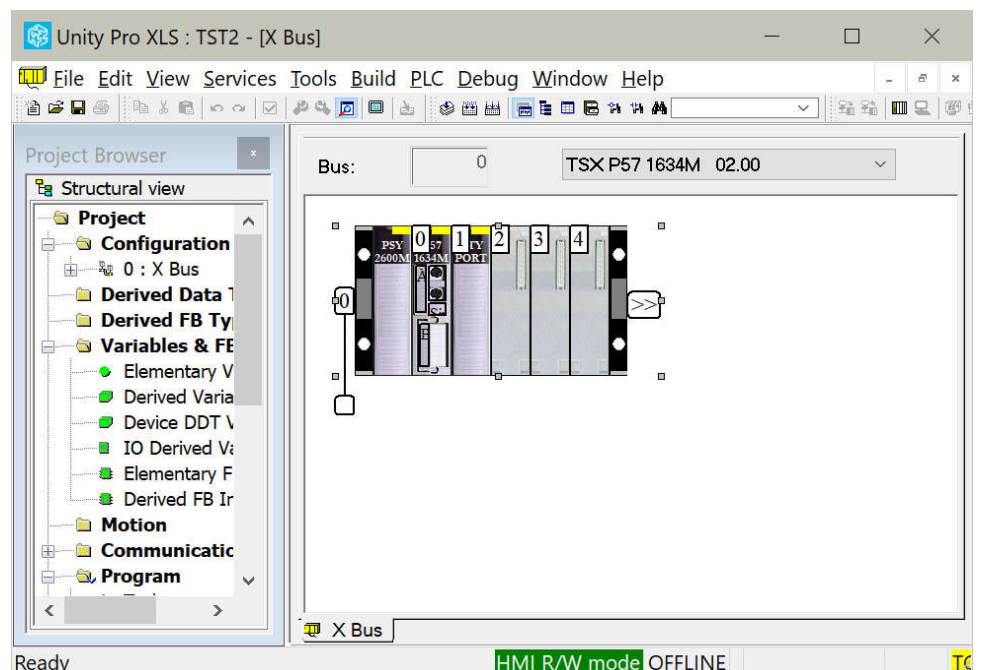
(c) Programming environment, Unity Pro.

Fig.1: PLC and terminal setups (a, b). The programming environment, shown in (c), is yet to be configured with the hardware I/O modules available in the laboratory.

This document is composed by two parts: in the first part are created simple PLC projects and in the second part is tried a data logging method.

[1] *This training problem has no direct contribution to the final grade.*

Part A - Introduction to PLC Programming

The first part of the Training Lab is structured as a hands-on guided tutorial, where you will learn how to:

- Create a new Unity Pro project
- Configure it for the right PLC, and PLC modules
- Start, connect to and use the Unity Pro Simulator
- Set input and output variables
- Create and run a simple Ladder program
- Connect to and use the PLC
- Run a program on the PLC
- Use the Lab panel

A1. *(Simulation with Unity Pro)* Do a simulated experiment using the Unity Pro v13 simulator. Use ladder to create a simple logic function based on two digital inputs and one digital output. Suggestion: start Matlab and run the command `login_mapi` to place you in a given folder already containing a startup project.

A2. *(See the simulator window)* Find in Unity Pro, help menu, information about `PLC Simulator`. Note that the Unity Pro window usually hides behind it the simulator window. How do you see the simulator window? See Fig.2.
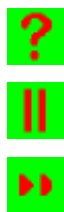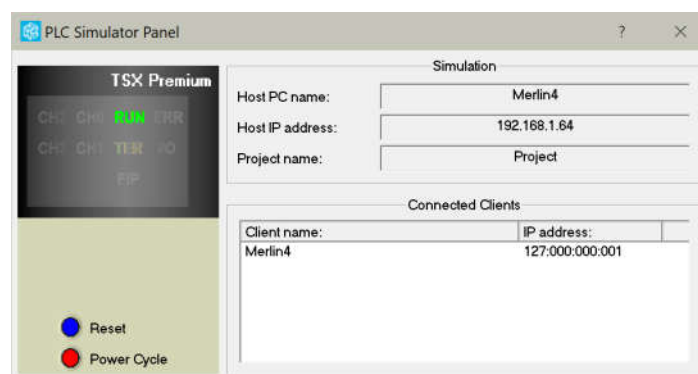
(a) Taskbar icons   (b) The simulator window

Fig.2: Find the simulator window by finding and double clicking one of the icons (a) appearing in the Windows taskbar. The simulator window (b) can be right clicked for setting options.

A3. *(Experiment with the real PLC)* Check that you can use the programming environment to set output values in the real hardware, namely by turning ON/OFF the output %q0.4.0 (or %q0.3.16, depending on the setup available on your workbench).

A4. *(Take a note to revisit whenever you need)* Make a Zoom video showing the making of a Unity Pro project from declaring the processor and modules till setting/unsetting an output.

Part B - Data Logging using the PLC Memory

One method of debugging and observing a PLC operation is to use part of the PLC memory to store events. Typically, one stores time series of binary variables such as digital inputs (e.g. %i0.2.0), digital outputs (e.g. %q0.4.0) and/or Boolean memories (e.g. %m0). See details in:

http://www.isr.tecnico.ulisboa.pt/~jag/course_utils/plc_log/plc_data_log.html

B1. Download the file `data_log_up13.zip` from the referred webpage, uncompress the zip file to a folder and verify that `mem_dump_load_tst.m` plots a saved sequence of digital signals. The plot should be similar to the one in Fig.3.
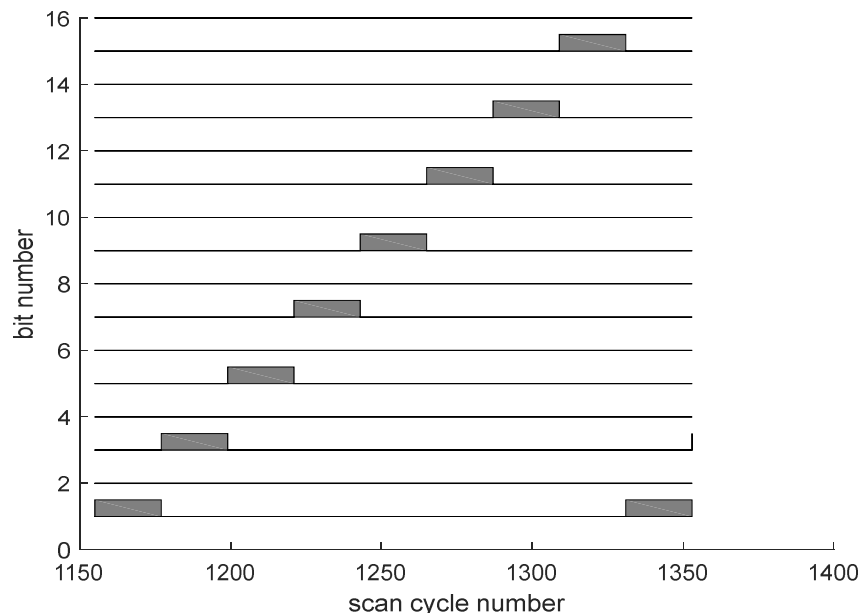

Fig.3: A sequence of digital signals saved in the file `data_log3_tmp.DTX`.

B2. Still on the folder where the zip file was uncompressed, use Unity Pro v13 to open the file `data_log3.stu`. Run the just opened project in simulation. Within Unity Pro v13, already running the project, find the window `tst_data_log` which has PLC code that can start multiple

demonstrations, see Fig.4. Right click on the memory variable `%m92` and set it to 1. The color of `%m92` changes from red to green. After approximately 5 seconds use Unity Pro (menu) to do `PLC > Save Data from PLC to File` and overwrite `data_log3_tmp.DTX`. See that in Matlab `mem_dump_load_tst.m` plots a different sequence of digital signals.
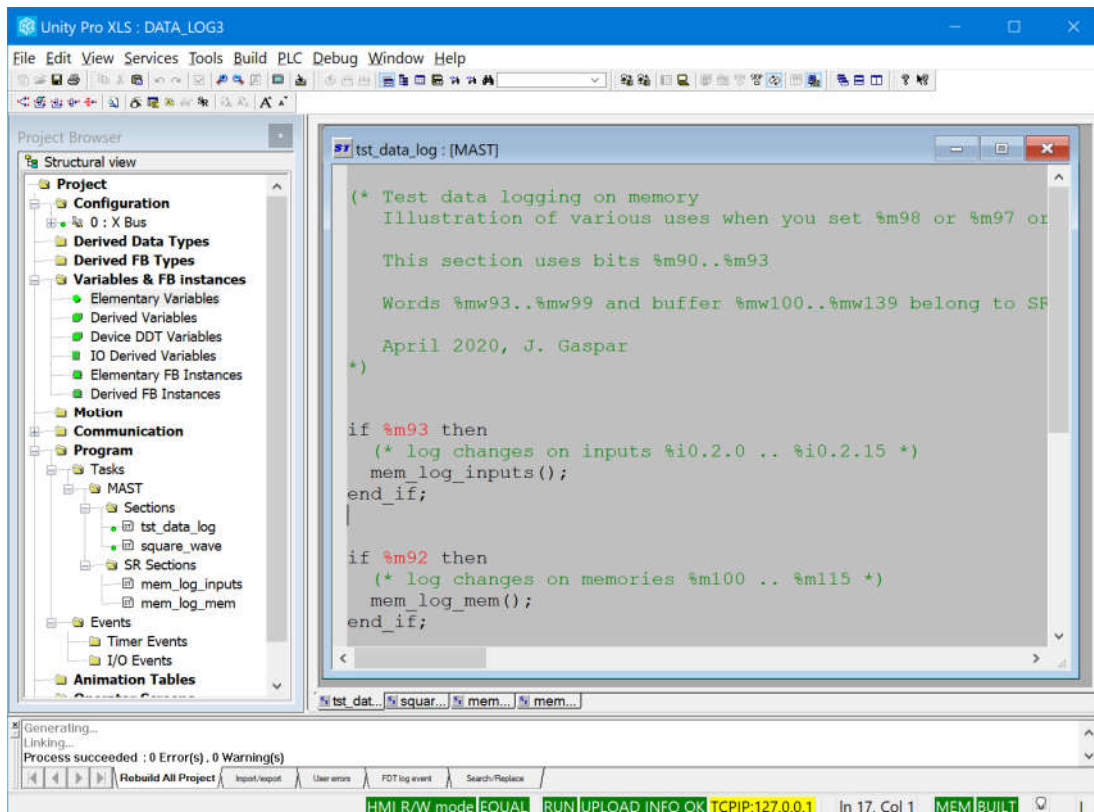

Fig.4: Project running, see `%m92` in red, which turns green with a right click and setting to 1.

B3. Adapt the project for your hardware, i.e. correct the PLC processor and IO modules if needed. Run the project using the real PLC. Try the data logger to monitor turning ON/OFF of two binary inputs, namely two switches of the terminal (see Fig.1 (a) or (b)). Use the two switches to make a sequence of states 00, 01, 11, 10, 00, 01, 11, 10, 00, each one taking about a couple of seconds. Make a data log (plot) showing each state of the sequence.