

Modeling and Automation of Industrial Processes

Modelação e Automação de Processos Industriais / MAPI

Introduction to MAPI laboratory

Prof. Luis Custódio

Prof. José Gaspar

IST / DEEC / MAPI

Shift 1: Mon 09:30-11:00
Thu 10:00-11:30

Número	Grupo Nº	Nome
66325	1	Tomás Marques Videira Fonseca
96139	1	Afonso Miguel De Almeida Santos Klier
96329	1	Tiago Filipe Espadinha Simões
93143	2	Miguel Henrique Dos Santos Vicente Alves Nabais
96134	2	Afonso Bispo Certo
96945	3	Sebastião Barroso Dias Gonçalves Chaves
97381	3	João Mendes das Neves Martins
106903	4	Afonso Folgado Soares Amorim de Figueiredo
96297	4	Pedro Afonso Botelho Pires Lopes Dias

Shift 2: Wed 12:30-14:00
Thu 11:30-13:00

Número	Grupo Nº	Nome
86680	1	Ricardo Beja Belchior dos Santos
105737	1	João Lourenço Mestre Vitorino de Almeida e Paiva
104688	1	Rodrigues de Fátima Jacinto
90058	2	Duarte Mata da Silva Honrado
81595	2	António Martim Carneiro Portugal e Vasconcelos
93150	2	Paulo Luís Santos Cruz
107842	3	Arianna Esposito
105065	3	Carla Juan Cazalla
107886	3	Eirik Berg Wang
93197	4	Tomás Pericão Moreira Roque Pires
93777	4	João Afonso Pacheco de Sousa

Training Laboratory Work

Part A - Introduction to PLC Programming

Part B - Data Logging using the PLC Memory

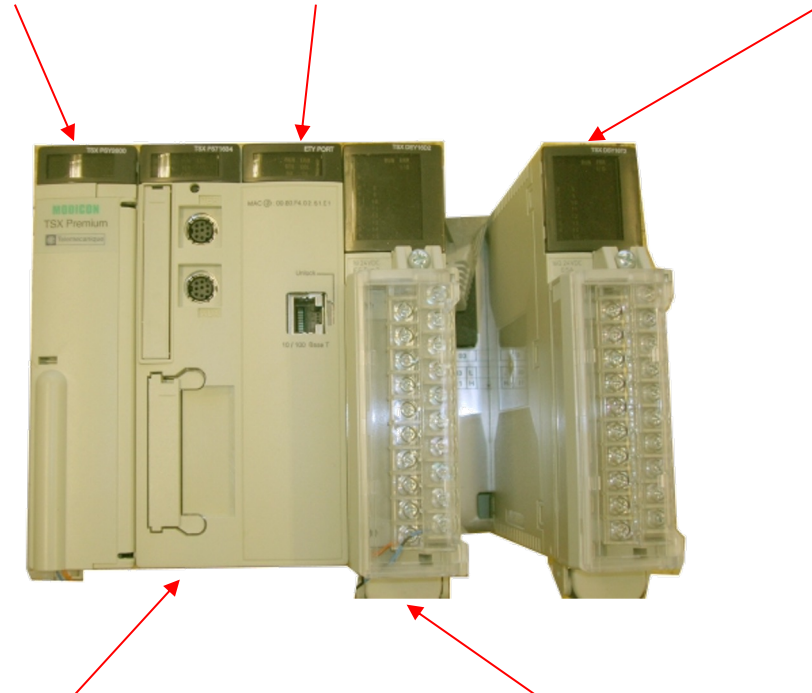
PLC TSX Premium P57 1634M



Power supply

Comm module

Output interface module



PLC processor (CPU)

Input interface module

The screenshot displays the Unity Pro XL software interface. The main window is titled "Unity Pro XL : <No name>*" and features a menu bar (File, Edit, View, Services, Tools, Build, PLC, Debug, Window, Help) and a toolbar. On the left, the "Project Browser" is visible, showing a "Structural view" of a project with folders for Configuration, Variables & FB instances, Motion, Communication, and Program. The "About Unity Pro XL ..." dialog box is open on the right, displaying the following information:

- Unity Pro XL V12.0 - 170404A
- Copyright (c) 2003-2017 Schneider Electric Industries SAS
- Registration informations:
 - Company: GlassWork
 - User name: Ju
 - Serial Number: 13345618595
 - Registration: Product is registered (permanent).
- Warning: This software is protected by copyright law and by international conventions. Any reproduction or distribution of the software in whole or in part, by any means is strictly prohibited.
- Integrity check: Perform self-test

The Schneider Electric logo is visible at the bottom right of the dialog box.

Unity PRO XL V11 V12 V13

toolbar

Editor window

The screenshot displays the Unity Pro XL software interface for Tank Management. The main window is titled "Unity Pro XL: TANK_MANAGEMENT" and features a menu bar (File, Edit, View, Utilities, Tools, Build, PLC, Debug, Window, ?) and a toolbar with various icons for file operations, simulation, and debugging. The interface is divided into several panes:

- Project Browser (Left):** Shows a hierarchical tree view of the project structure, including folders for Configuration, Derived Data Types, ALARM_DIA, Motor, Valve, Variables & FB instances, Communication, Routing Table, Program, Tasks (with MAST selected), Sections, Diagnostics, Simulation, application, SR Sections, Events, and Animation tables.
- Editor Window (Center):** Displays a graphical representation of a tank with a "Start Cycle" button and a "Tank high saf" indicator. Below it is a ladder logic diagram for "application: [MAST]" with rungs 8 through 14. Rung 8 shows "Open va..." and "Open va..." contacts leading to a "Valve" coil. Rung 9 shows "Close_v..." and "Close_v..." contacts leading to a "Valve" coil. Rung 10 shows "Lim_valve_opening" and "Lim_valve_closure" contacts leading to a "Valve" coil. Rung 11 shows "Lim_valve_opening" and "Lim_valve_closure" contacts leading to a "Valve" coil. Rung 12 shows "Lim_valve_opening" and "Lim_valve_closure" contacts leading to a "Valve" coil. Rung 13 shows "Lim_valve_opening" and "Lim_valve_closure" contacts leading to a "Valve" coil. Rung 14 shows "Lim_valve_opening" and "Lim_valve_closure" contacts leading to a "Valve" coil.
- Table (Right):** A table listing variables and their values. The table has columns for "Name" and "Value".

Name	Value
Stop	0
Valve_closure_cmd	0
Valve_opening_cmd	0
Valve_opening_error	0
Lim_valve_closure	0
Lim_valve_opening	1
Run	0
Tank_low_safety	0
Tank_high_safety	0
Tank_low_level	0
Tank_high_level	1
Motor_run_cmd	0
Tank_vol	9.2
Rate	0
Valve_flow	0.4
Pump_rate	0.4
Valve_closure_error	0
Contactors_return	1
Valve_closure_t	0s
Valve_opening_t	0s

The status bar at the bottom indicates "Process successful: 0 Error(s), 0 Warning(s)" and includes buttons for "Build Project", "Import/Export", "User Errors", and "Find/Replace". The status bar also shows "HMI Read/Write Mode", "EQUAL", "RUN", "NO UPLOAD INFO", "TCPIP:127.0.0.1", and "GENERAT" buttons.

Project browser

Information window

IST / DEEC / MAPI

Group folder creation:

☞ Start MatLab 2022

☞ Run

```
>> login_mapi <group_number>
```

If you don't know your group number, just run

```
>> login_mapi
```

Find your group number in the list and enter it

This command creates your group folder with the following path (write it down):

C:\users2\mapi\<group_number>

Save all your files and projects in your group folder

IST / DEEC / MAPI

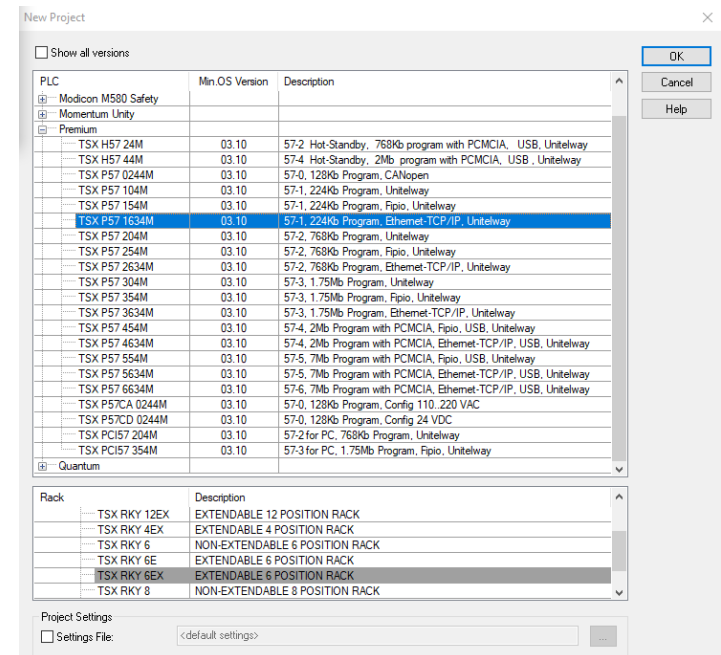
Configuration:

In the Lab there are PLCs with two different processors: 1634 or 2634

In the Unity software be sure that the right processor is selected
(**version 2.00** for both)

👉 File → New

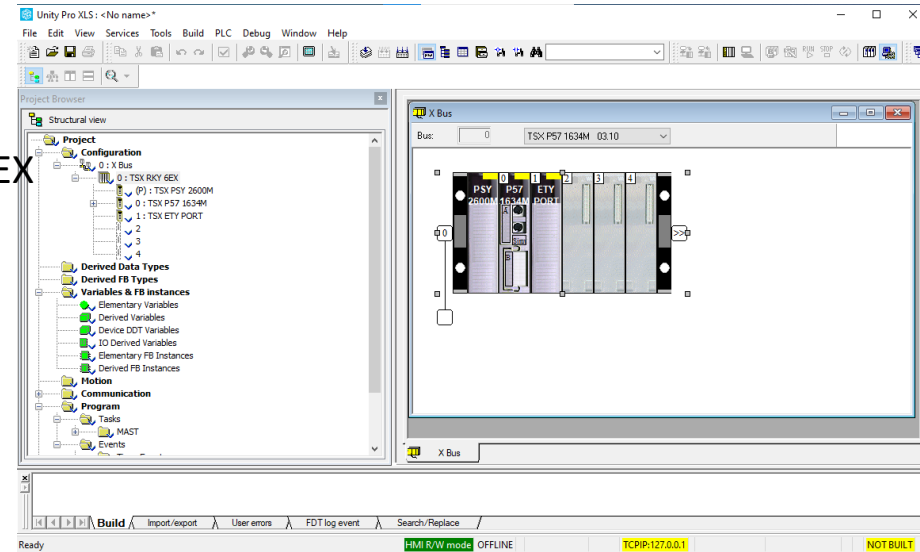
Premium TSX P57 1634M (or 2634)



IST / DEEC / MAPI

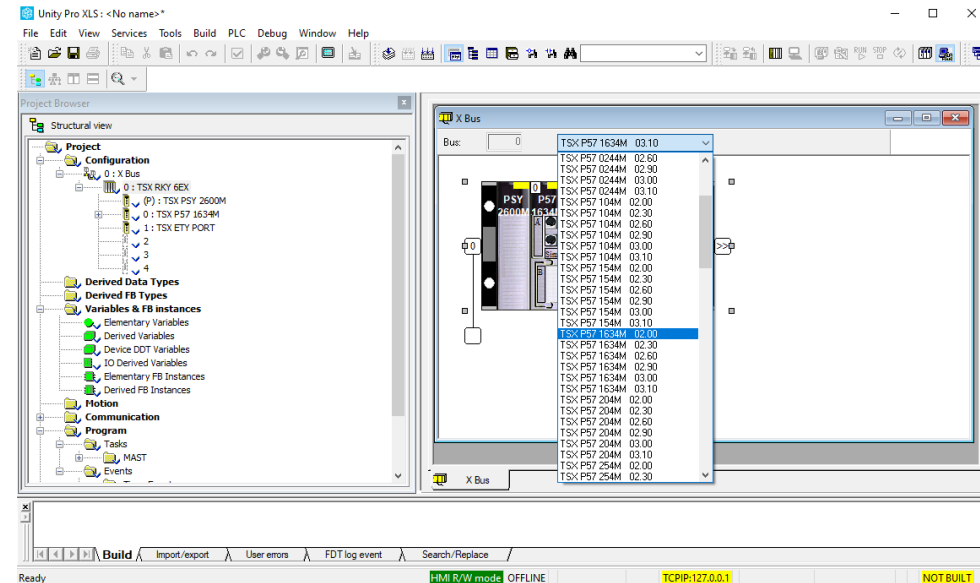
☞ (Project browser) click
doubleclick

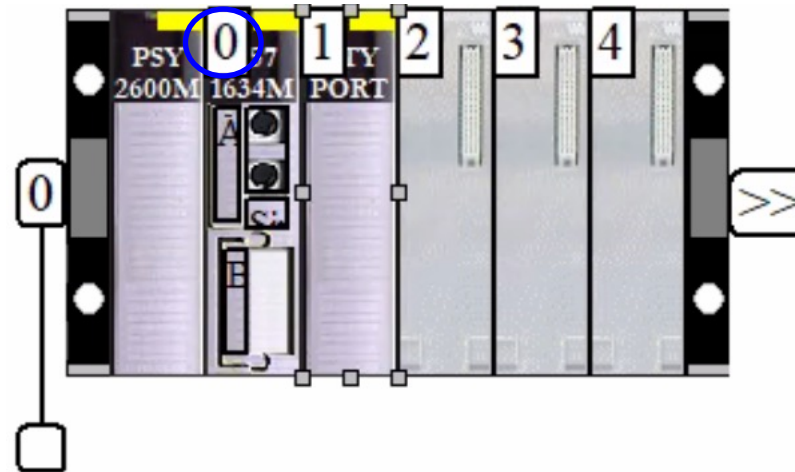
+ 0:X Bus
0:TSX Rky 6EX



downgrade to version 2.00:

or
TSX P57 1634M 02.00
2634M 02.00



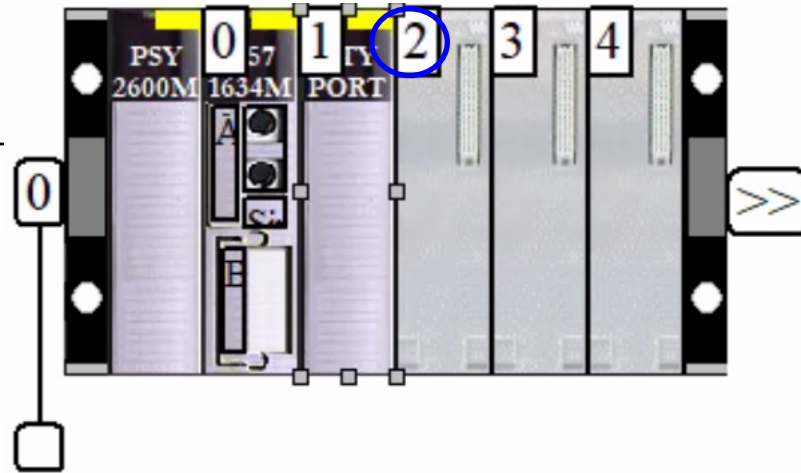


☞ doubleclick module 0

select tab I/O objects
[select Implicits %I and %Q]
[update grid with]

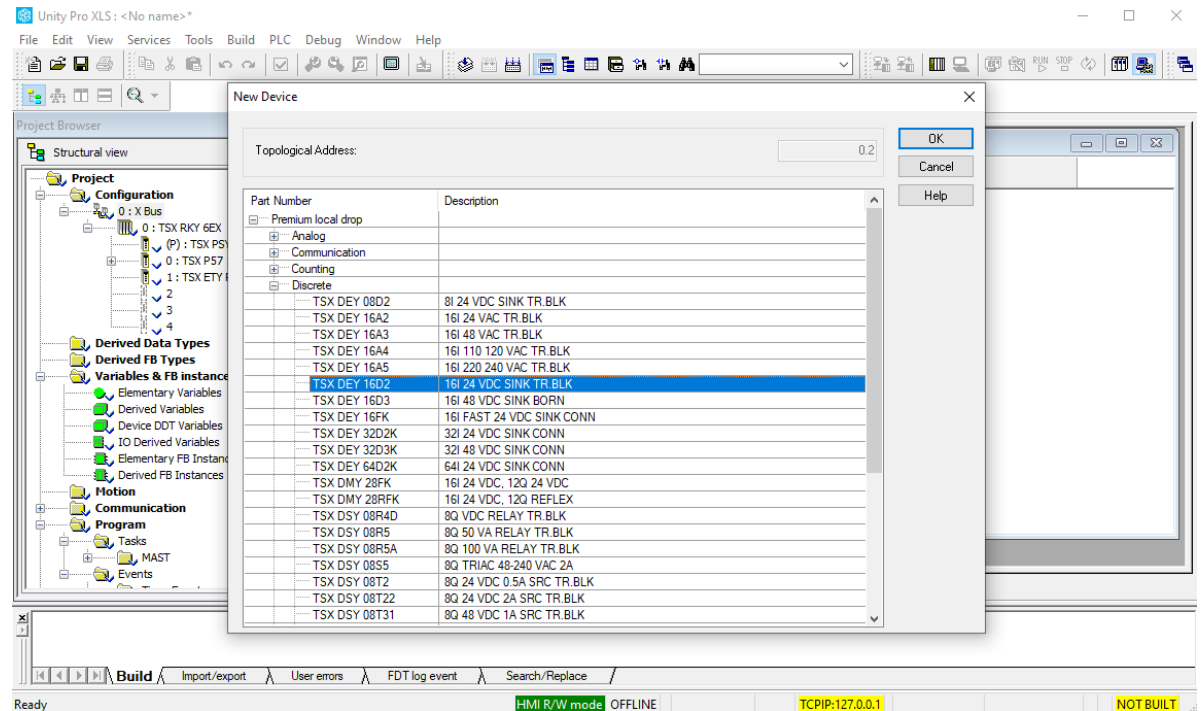
IST / DEEC / MAPI

Configuration:



doubleclick 2

select DEY 16D2 (discrete) or DMY 28FK (module 3)

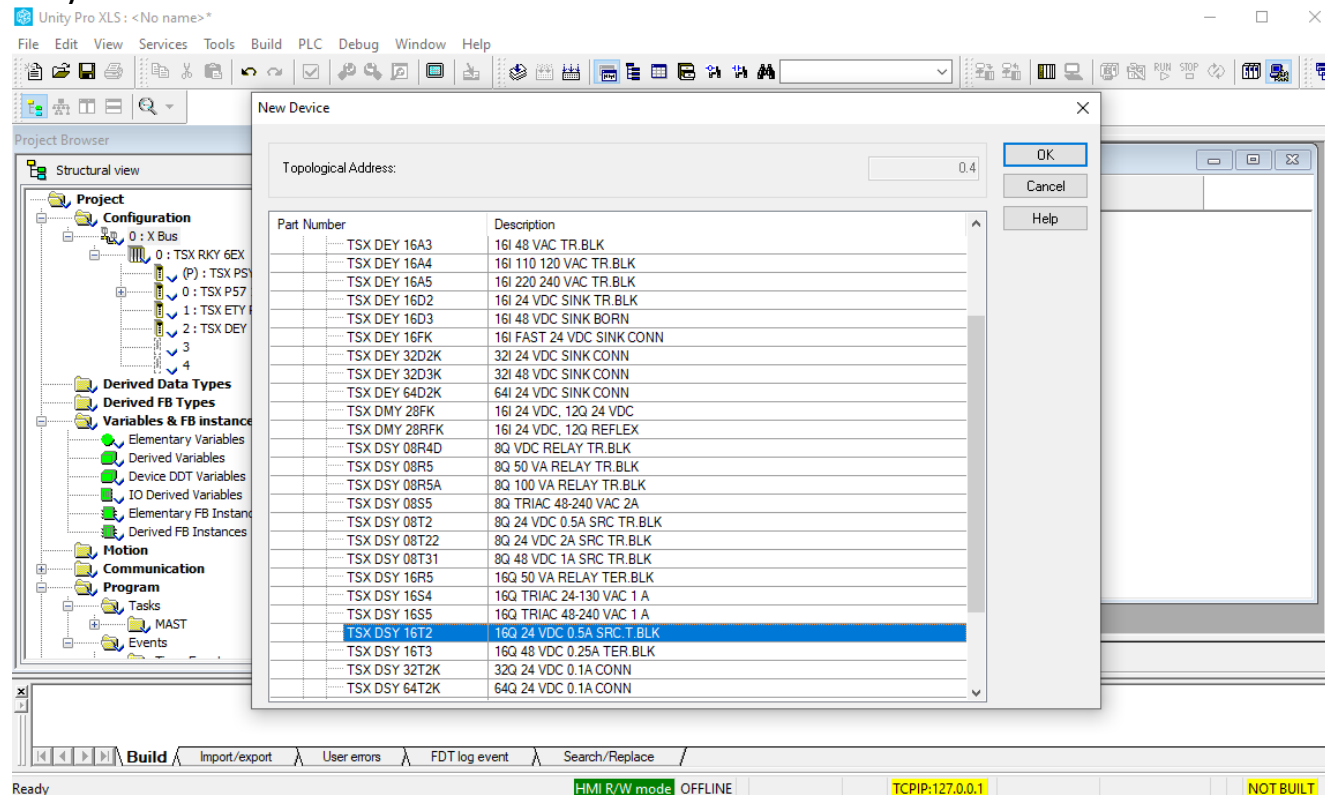
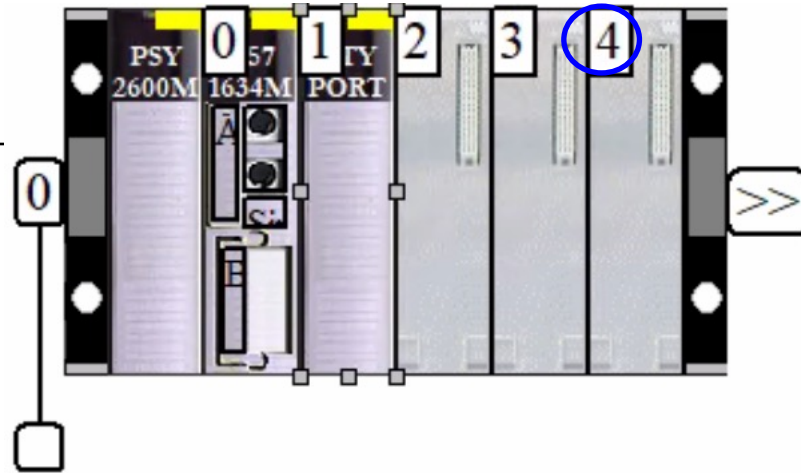


IST / DEEC / MAPI

Configuration:

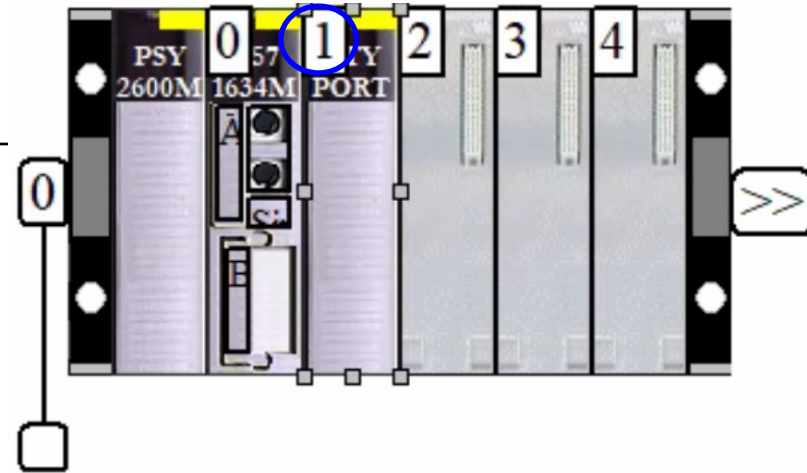
doubleclick **4**

select **DSY 16T2 (discrete) or DMY 08R5**



IST / DEEC / MAPI

Configuration:



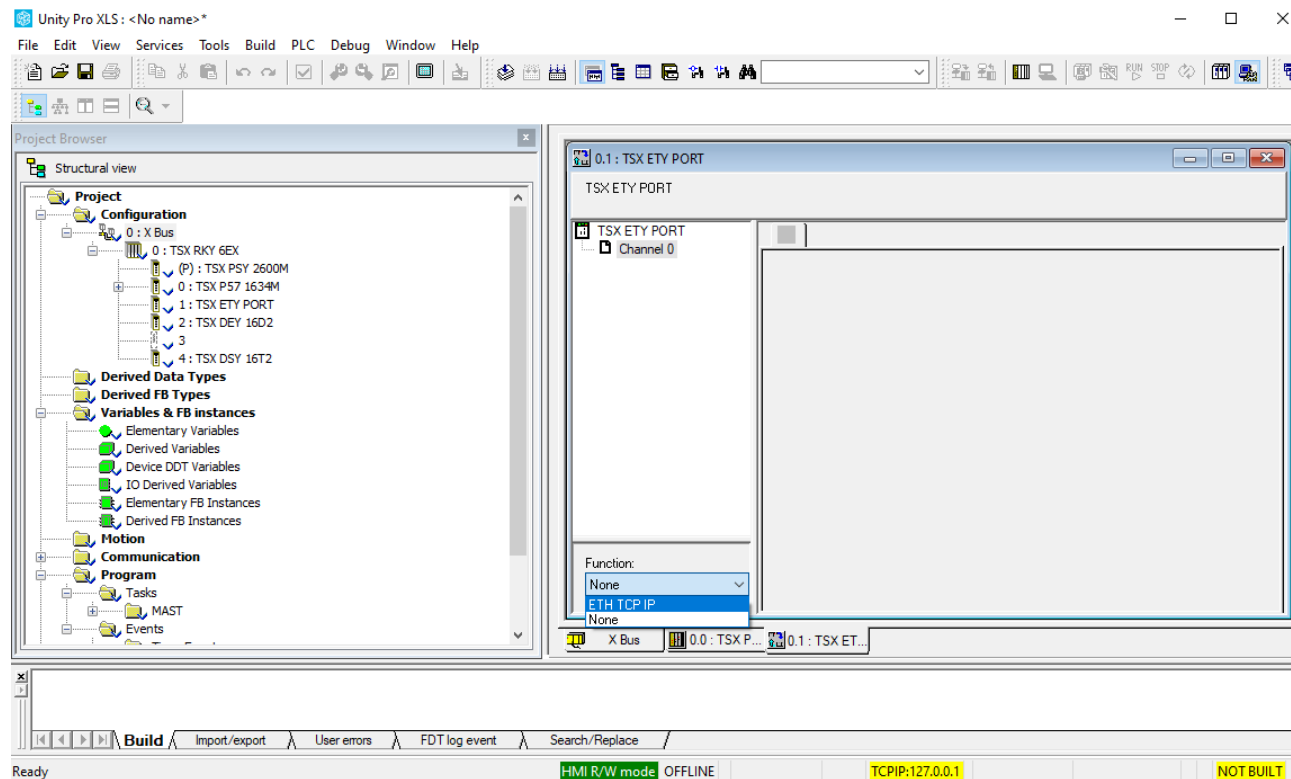
doubleclick



ETY Port & Channel 0:

Function "ETH TCP IP"

(close the window)



IST / DEEC / MAPI

Run:

👉 File → Save

[be sure to save in your folder c:\users2\mapi\]

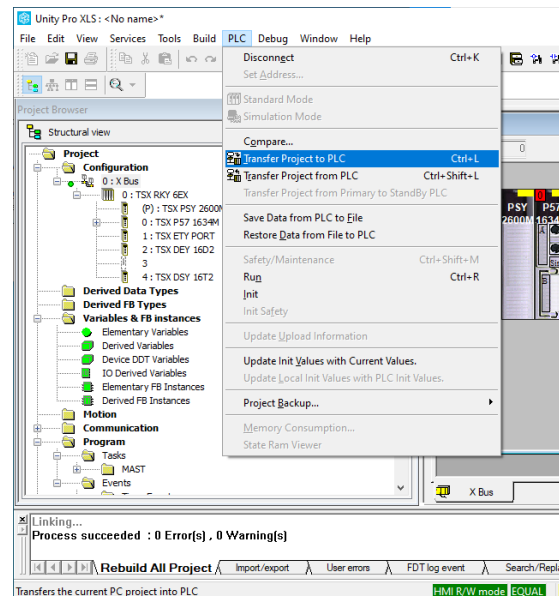
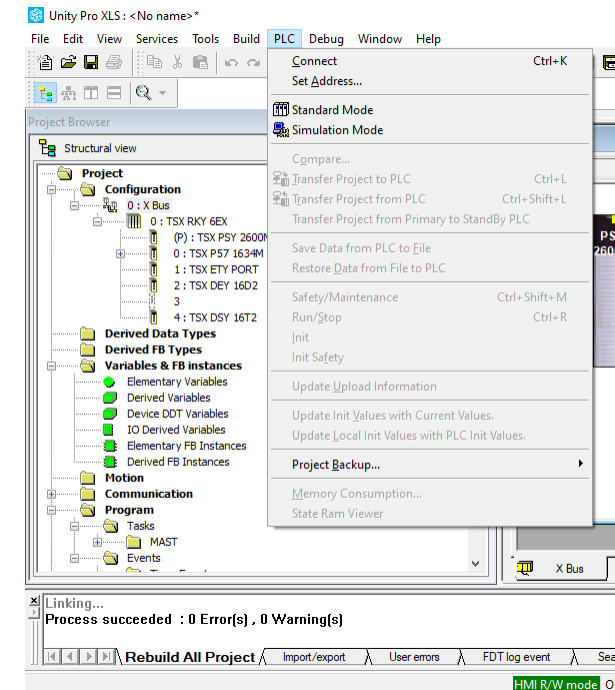
👉 PLC menu select Simulation Mode

👉 PLC → Connect

👉 PLC → Transfer project to PLC

(accept "Rebuild")

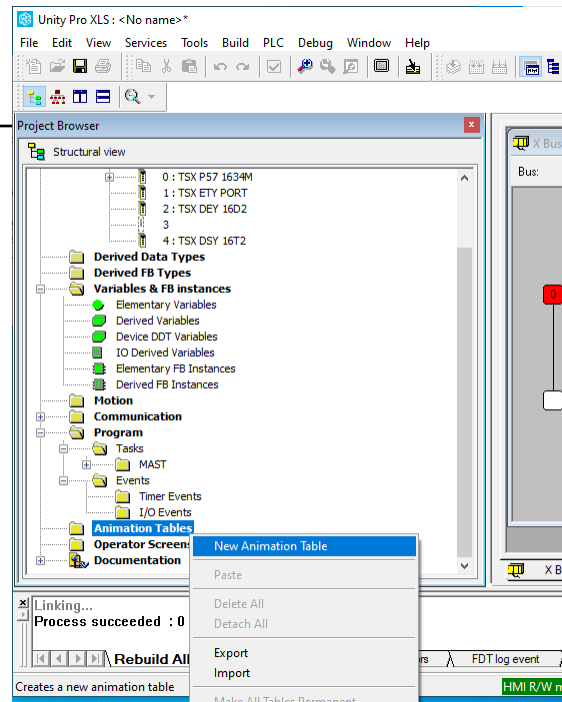
👉 PLC → Run



IST / DEEC / MAPI

Animation table:

☞ (Project browser) Animation Tables → New



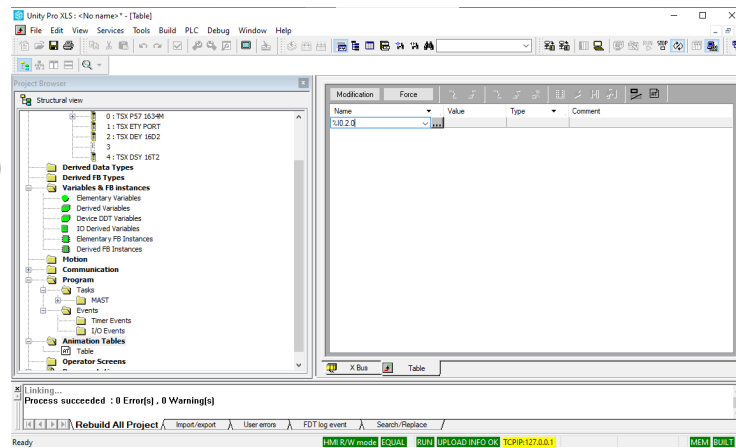
Name (insert)

%I0.2.0

%I0.2.1

%Q0.4.0

%Q0.4.1



DMY 28FK

%i0.3.0

%i0.3.1

%q0.3.16

%q0.3.17

IST / DEEC / MAPI

Animation table:



click Force



right click variable, option Force, Values: F0 F1 unforce

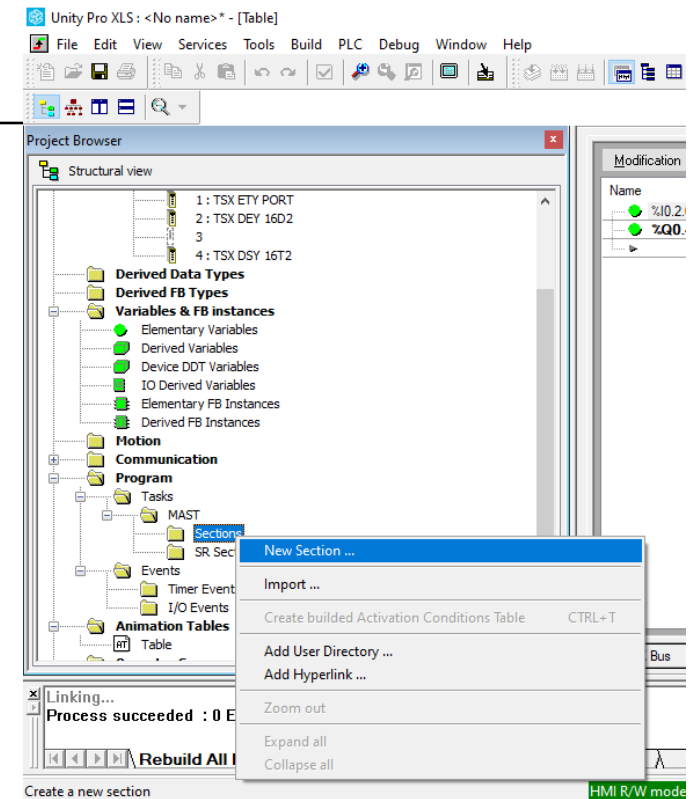
The screenshot displays the Unity Pro XLS software interface. The main window shows a table with columns for Name, Value, Type, and Comment. The table contains two rows: one for '%I0.2.0' with a value of '1' and another for '%Q0.4.0' with a value of '0'. A context menu is open over the '%Q0.4.0' row, showing options like Cut, Copy, Paste, Insert, Delete, Select All, Expand All, Collapse, Display Format, Set, Force, Customize Columns..., Initialize Search, Data Properties, and Properties. The 'Force' option is highlighted, and a sub-menu is visible with options: Force to 0, Force to 1, and Unforce. The Project Browser on the left shows a tree view of the project structure, including Derived Data Types, Derived FB Types, Variables & FB Instances, Motion, Communication, Program, Tasks, Events, Timer Events, I/O Events, Animation Tables, and Operator Screens. The status bar at the bottom indicates 'Ready' and 'HMI R/W mode EQUAL RUN UPLOAD INFO OK TCP/IP:127.0.0.1 MEM BUILT'.

Name	Value	Type	Comment
%I0.2.0	1		
%Q0.4.0	0		

IST / DEEC / MAPI

First Ladder program:

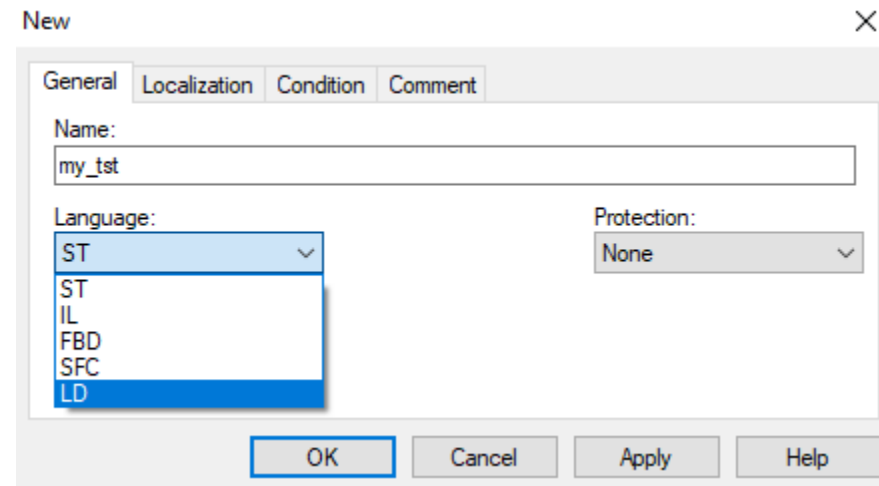
👉 Program → Tasks → MAST → Sections (right click)



“New section”

name: my_tst

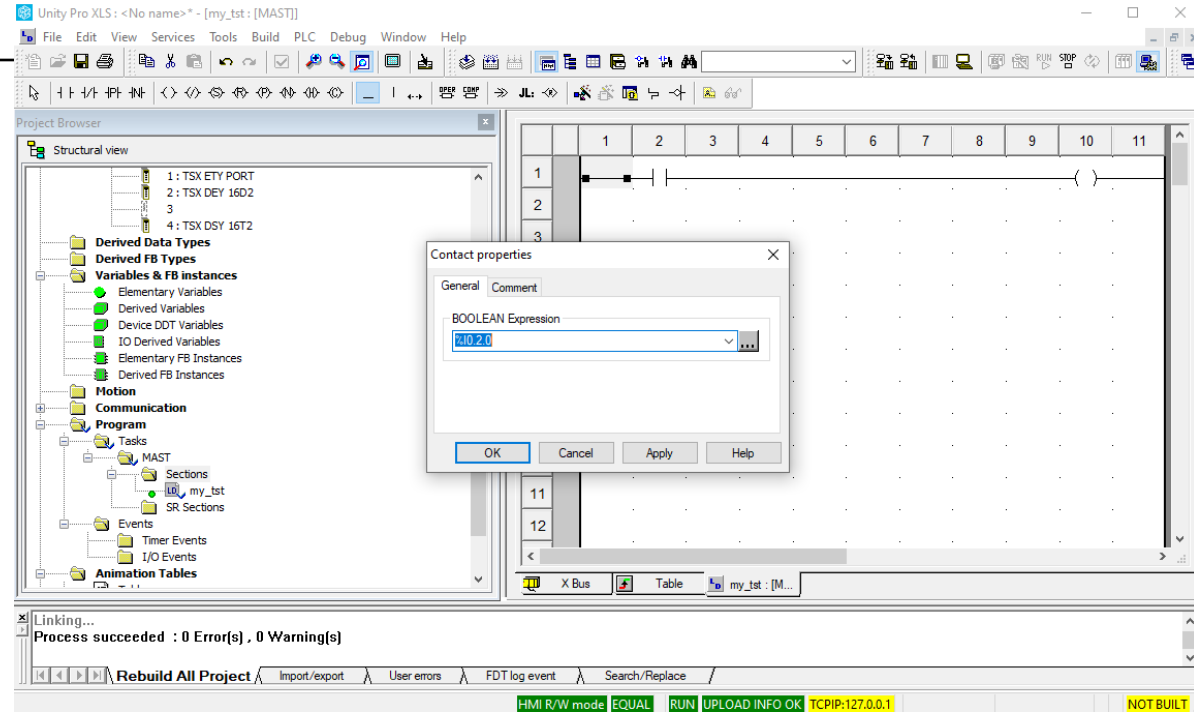
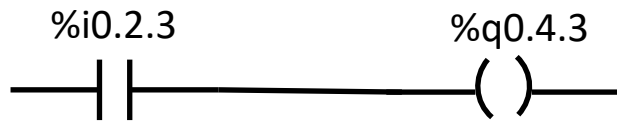
language: LD



IST / DEEC / MAPI

First Ladder program:

☞ (Editor window) enter



(add to the animation table %i0.2.3 and %q0.4.3)

☞ PLC → Connect (if not connected)

☞ PLC → Transfer project to PLC

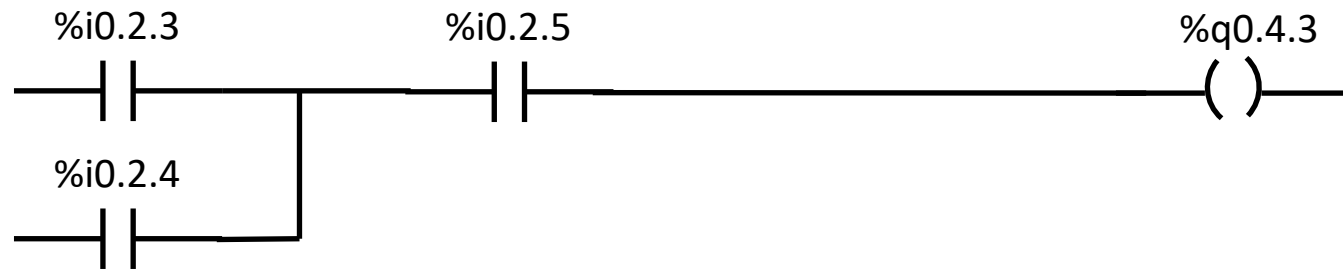
☞ PLC → Run

☞ Animation table - (right click) %i02.3

Force to 1

new Ladder program:

☞ (Editor window) enter



(test it)

What is the logic table?

IST / DEEC / MAPI

Lab hardware:

- PLC TSX Premium P57 in two different configurations: 1634 and 2634
- PC
- connection between PLC and PC uses a RS485 cable linked to the PC serial port
- Lab Panel with:
 - a 2-position switch (On/Off): On → bit 2 = 1
 - a 3-position switch (left/middle/right): left → bit 0 = 1, right → bit 1 = 1
 - a 12-key 4x3 keyboard (see more next)

- a buzzer (output bit 0 = 1)
- a red led (output bit 1 = 1)
- a yellow led (output bit 2 = 1)
- a green led (output bit 3 = 1)

- Power supply for the I/O modules



IST / DEEC / MAPI

Setup with one input and one output modules



IST / DEEC / MAPI

PLC testing:

- 👉 connect PLC (standard mode)
- 👉 connect power supply for the I/O modules
- 👉 change the program to



👉 PLC menu select Standard Mode

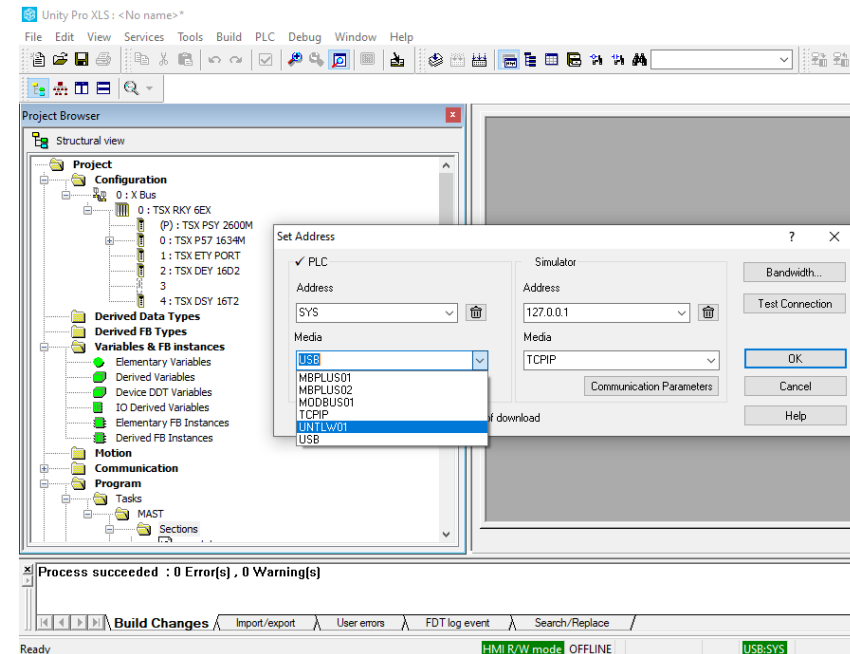
👉 PLC → Set address:

Media → UNTLW01
Address → SYS

👉 PLC → Connect

👉 PLC → Transfer project to PLC

👉 PLC → Run

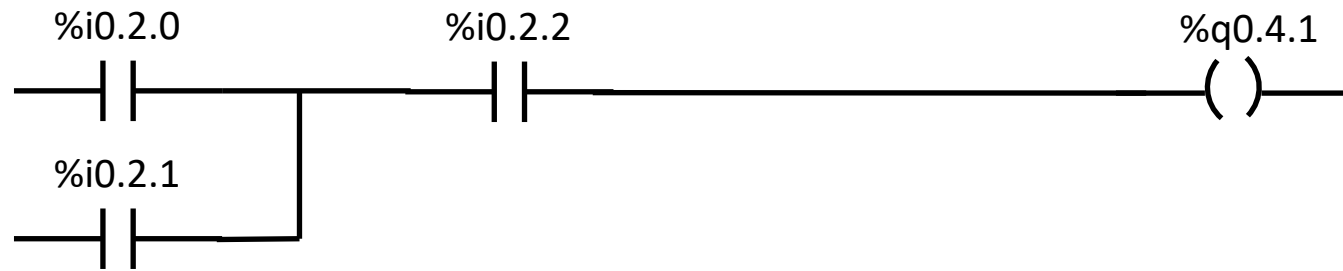


(guess what this program does)

IST / DEEC / MAPI

PLC testing:

☞ change the program to

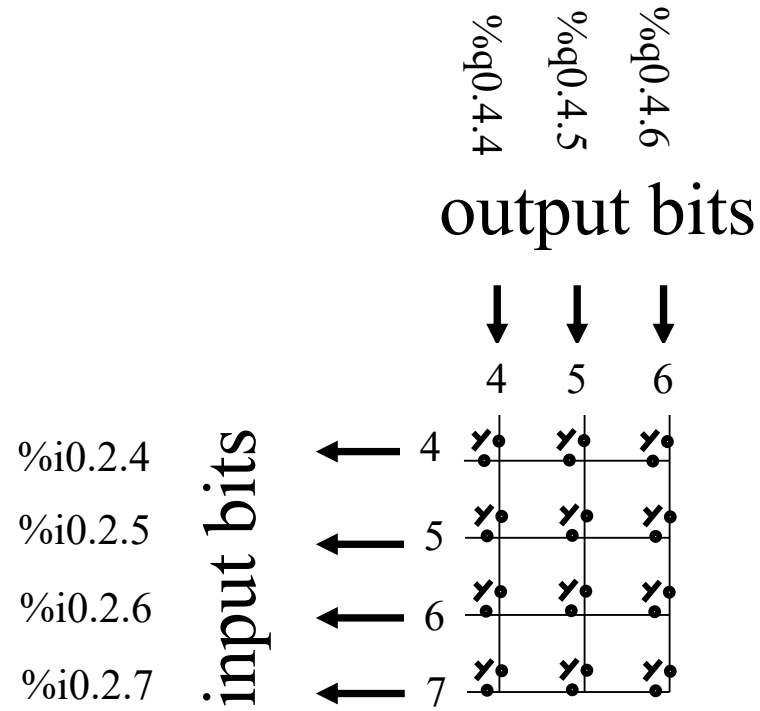
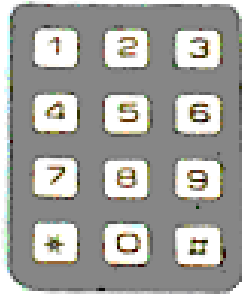


☞ PLC → Transfer project to PLC

☞ PLC → Run

(guess what this program does)

Lab keyboard:



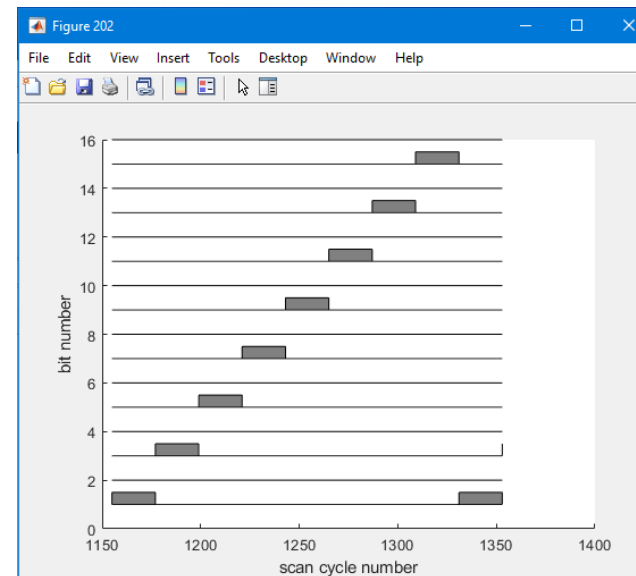
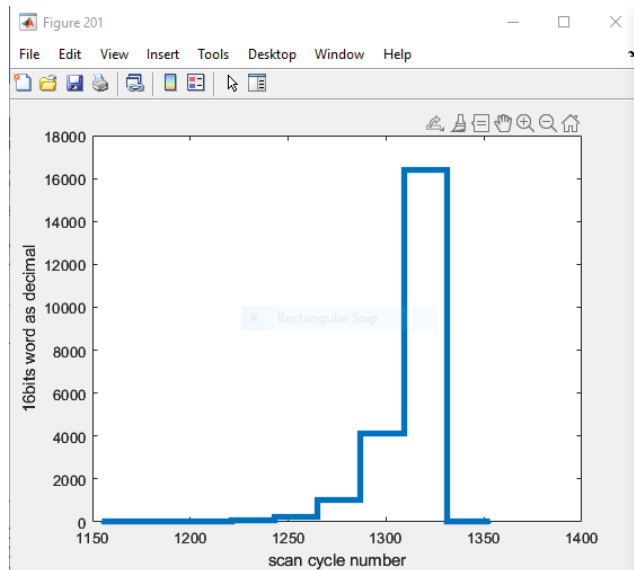
In order to identify that a key has been pressed, first we have to energize each one of the keyboard columns through the output bits 4 to 6, and then read the input bits 4 to 7

For instance, if $\%oq0.4.4$ is set to 1 (and both $\%oq0.4.5$ and $\%oq0.4.6$ are zero) and $\%i0.2.6$ is 1, then key “7” is pressed

Data Logger

Procedure to test the program:

- download the zip file [data_log_up13.zip](http://users.isr.ist.utl.pt/~jag/course_utils/plc_log/data_log_up13.zip) by running `>> get_files_logger` using Matlab (also available at http://users.isr.ist.utl.pt/~jag/course_utils/plc_log/data_log_up13.zip)
- Decompress the downloaded zip file to your group folder (extract, not double click)
- the zip file contains already one log file "**data_log2_tmp.dtx**"
- Using Matlab change dir to your group folder and run: `>> mem_dump_load_tst` and observe the produced plots



Data Logger (cont)

Procedure to test the program:

- Using Unity Pro, open the file **data_log2.stu**, transfer the code to the PLC simulator, and run it.
- Double-click on section 'tst_data_log' and try to interpret the code

- Set variable %m92 to 1 to start the log process
- Check value of variable %mw96
- Wait until %mw96 = 20
- using Unity Pro, export the memory to a file, PLC -> **"Save data from PLC to file"**. Save data into the file **"data_log2_tmp.dtx"** (overwrite it) in your group folder.

```
if %m93 then
  (* log changes on inputs %i0.2.0 .. %i0.2.15 *)
  mem_log_inputs();
end_if;

if %m92 then
  (* log changes on memories %m100 .. %m115 *)
  mem_log_mem();
end_if;

(* test buffer full and place this info in %m91 *)
if %mw96 >= 2*MOD(%MW99,1000) then
  %m91 := 1;
else
  %m91 := 0;
end_if;

(* delete logged data *)
if %m90 then
  %mw97 := 0;
  (* reset will be done in the next call to mem_log_mem() *)
  %m90 := 0;
end_if;
```

Data Logger (cont)

Procedure to test the program:

- Within Matlab run again: >> **mem_dump_load_tst** and observe the new plots
- Reset the log memory by setting the adequate variable
- Start a new log process and observe the changes in the produced plots

Data Logger (cont)

Procedure to test the program:

- Double-click on section 'square_wave' and try to interpret the code
- Set variable %m97 to 1
- Reset the log memory
- Start a new log process and observe the changes in the produced plots

```
(* Use one timer for all square waves *)
TON_0 (IN := NOT(%M99) (*BOOL*),
      PT := t#0.5s (*TIME*),
      Q => %M99 (*BOOL*));

(* Update the output whenever TON_0 shows a timeout *)
IF %M99 THEN

  (* output a 1Hz square wave to %q0.4.1 *)
  %q0.4.1 :=NOT(%q0.4.1);

  (* %m97=0 make square waves, %m97=1 pulse even bits *)
  (* if you want, output all zeros by setting %m98 *)
  if %m98 then
    %mw90 := 0;
  end_if;
  WORD_TO_BIT( int_to_word(%mw90),
    %m100, %m101, %m102, %m103,
    %m104, %m105, %m106, %m107, |
    %m108, %m109, %m110, %m111,
    %m112, %m113, %m114, %m115 );
  if NOT(%m97) then
    (* update counter to make square waves in memory %m100..%m115 *)
    %mw90 := %mw90 +1;
  else
    (* pulse even bits in memory %m100..%m115 *)
    if %mw90=0 then %mw90:=1;
    else %mw90:= ROL(%mw90,2);
    end_if;
  end_if;
END_IF;
```

Data Logger (cont)

- Solve item B3 using the PLC

Use the two switches in Lab panel 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.

