Modeling and Automation of Industrial Processes

Modelação e Automação de Processos Industriais / MAPI

PLC Network Security

http://www.isr.tecnico.ulisboa.pt/~jag/courses/mapi2223

Prof. José Gaspar, rev. 2022/2023

Some pointers to PLC security

Bibliography:

- **On PLC network security**
  Asem Ghaleb, Sami Zhioua, and Ahmad Almulhem
  International Journal of Critical Infrastructure Protection 22, 2018
  https://www.researchgate.net/publication/325700543_On_PLC_network_security

- **The real story of Stuxnet**
  D. Kushner
  IEEE Spectrum, 3(50), pp.48-53, 2013.
  https://spectrum.ieee.org/telecom/security/the-real-story-of-stuxnet

Online Videos:

STUXNET: The Virus that Almost Started WW3 - YouTube (~3min)
https://www.youtube.com/watch?v=7g0pi4J8auQ

American Blackout - National Geographic (Full Movie) Cyber Attack
https://www.youtube.com/watch?v=zNfJkMPTtWQ
YouTube (~1h30min) [link found broken in April 2021; Google gives alternatives for the title]

Cloud ***cyber-physical systems*** *are the natural evolution of* ***embedded systems*** *based on trends which may be observed in various domains [Jirkovsky18].*

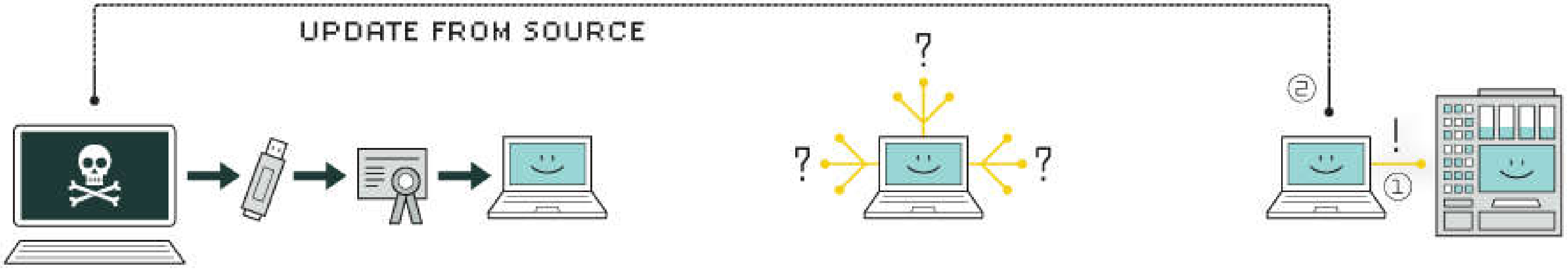
2010 - The **Stuxnet worm is detected**. It is the first worm known to attack SCADA (supervisory control and data acquisition) systems [Kushner18].

2011 - The Duqu worm is discovered. Unlike Stuxnet, to which it seems to be related, it was designed to **gather information** rather than to interfere with industrial operations.

[Jirkovsky18] OPC UA Realization Of Cloud Cyber-Physical System, V. Jirkovský, P. Kadera and M. Obitko, IEEE Int. Conf. on Industrial Informatics (INDIN) 2018

[Kushner18] The real story of Stuxnet, D. Kushner, IEEE Spectrum, 3(50), pp.48-53, 2013.

HOW STUXNET WORKED



1. infection

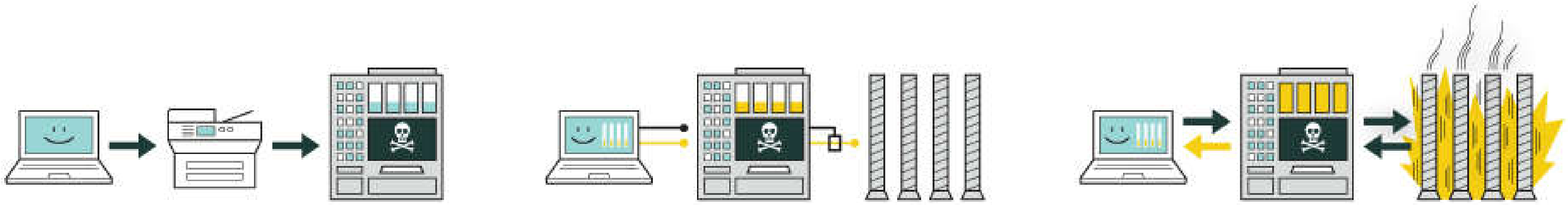
Stuxnet enters a system via a USB stick and proceeds to infect all machines running Microsoft Windows. By brandishing a digital certificate that seems to show that it comes from a reliable company, the worm is able to evade automated-detection systems.

2. search

Stuxnet then checks whether a given machine is part of the targeted industrial control system made by Siemens. Such systems are deployed in Iran to run high-speed centrifuges that help to enrich nuclear fuel.

3. update

If the system isn't a target, Stuxnet does nothing; if it is, the worm attempts to access the Internet and download a more recent version of itself.



4. compromise

The worm then compromises the target system's logic controllers, exploiting "zero day" vulnerabilities-software weaknesses that haven't been identified by security experts.

5. control

In the beginning, Stuxnet spies on the operations of the targeted system. Then it uses the information it has gathered to take control of the centrifuges, making them spin themselves to failure.

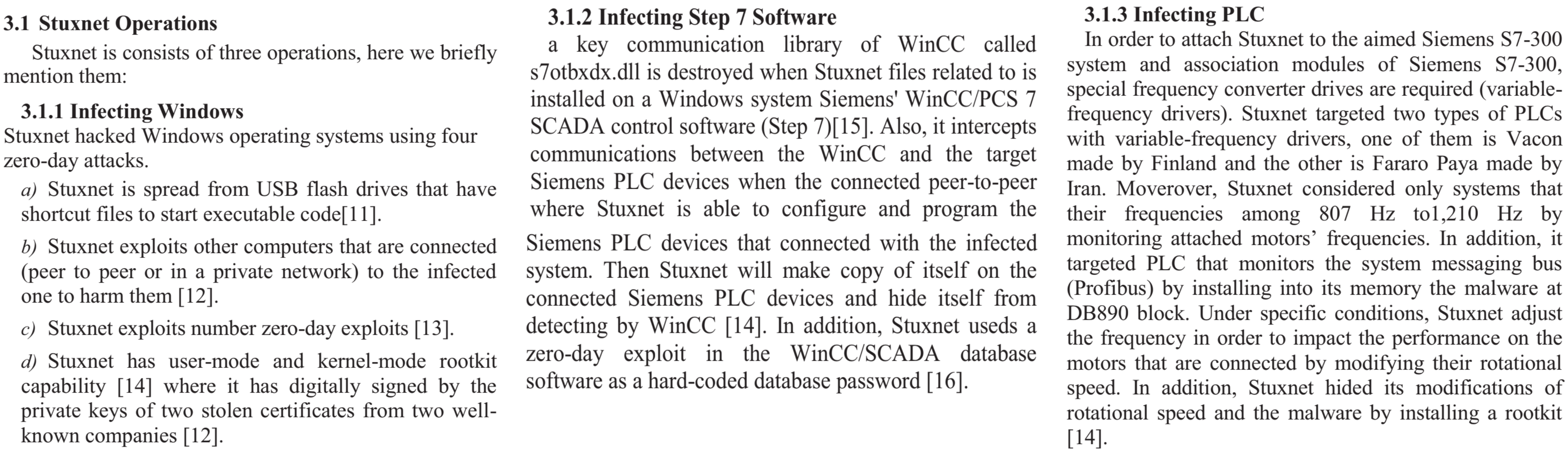
6. deceive and destroy

Meanwhile, it provides false feedback to outside controllers, ensuring that they won't know what's going wrong until it's too late to do anything about it.

https://spectrum.ieee.org/image/MjIyMTQzMg.jpeg

A 2018 publication:

S. Al-Rabiaah, "The “Stuxnet” Virus of 2010 As an Example of A “APT” and Its “Recent” Variances," *2018 21st Saudi Computer Society National Computer Conference (NCC)*, Riyadh, Saudi Arabia, 2018, pp. 1-5, doi: 10.1109/NCG.2018.8593143.



3.1 Stuxnet Operations

Stuxnet is consists of three operations, here we briefly mention them:

3.1.1 Infecting Windows

Stuxnet hacked Windows operating systems using four zero-day attacks.

a) Stuxnet is spread from USB flash drives that have shortcut files to start executable code[11].

b) Stuxnet exploits other computers that are connected (peer to peer or in a private network) to the infected one to harm them [12].

c) Stuxnet exploits number zero-day exploits [13].

d) Stuxnet has user-mode and kernel-mode rootkit capability [14] where it has digitally signed by the private keys of two stolen certificates from two well-known companies [12].

3.1.2 Infecting Step 7 Software

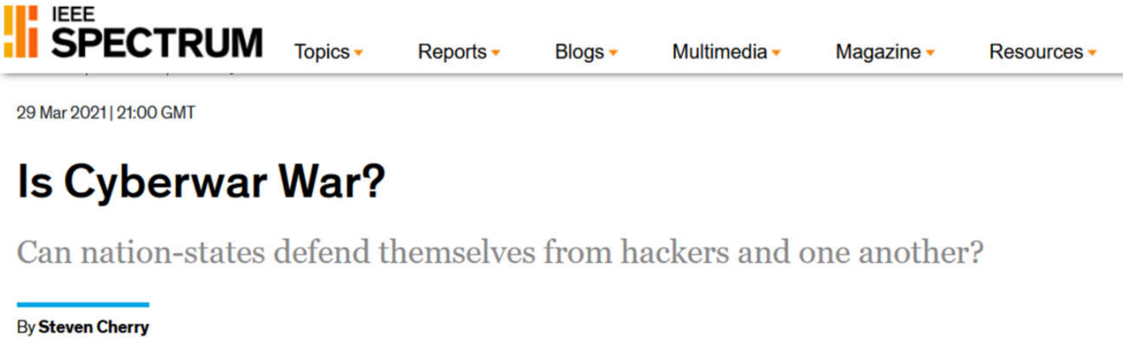
a key communication library of WinCC called s7otbxdx.dll is destroyed when Stuxnet files related to is installed on a Windows system Siemens' WinCC/PCS 7 SCADA control software (Step 7)[15]. Also, it intercepts communications between the WinCC and the target Siemens PLC devices when the connected peer-to-peer where Stuxnet is able to configure and program the Siemens PLC devices that connected with the infected system. Then Stuxnet will make copy of itself on the connected Siemens PLC devices and hide itself from detecting by WinCC [14]. In addition, Stuxnet useds a zero-day exploit in the WinCC/SCADA database software as a hard-coded database password [16].

3.1.3 Infecting PLC

In order to attach Stuxnet to the aimed Siemens S7-300 system and association modules of Siemens S7-300, special frequency converter drives are required (variable-frequency drivers). Stuxnet targeted two types of PLCs with variable-frequency drivers, one of them is Vacon made by Finland and the other is Fararo Paya made by Iran. Moverover, Stuxnet considered only systems that their frequencies among 807 Hz to1,210 Hz by monitoring attached motors’ frequencies. In addition, it targeted PLC that monitors the system messaging bus (Profibus) by installing into its memory the malware at DB890 block. Under specific conditions, Stuxnet adjust the frequency in order to impact the performance on the motors that are connected by modifying their rotational speed. In addition, Stuxnet hided its modifications of rotational speed and the malware by installing a rootkit [14].

More information:

https://spectrum.ieee.org/tag/Stuxnet



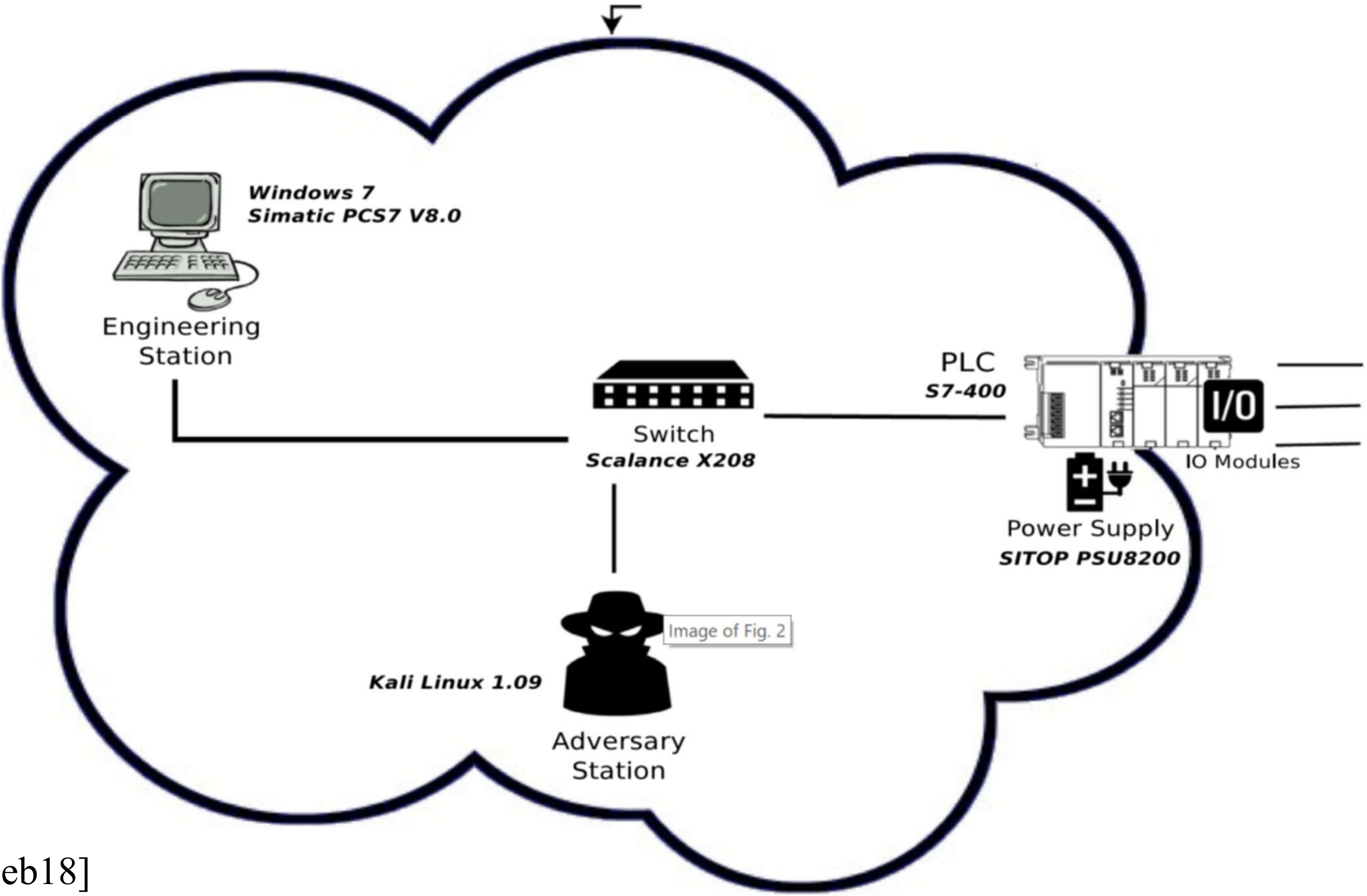
29 Mar 2021 | 21:00 GMT

Is Cyberwar War?

Can nation-states defend themselves from hackers and one another?

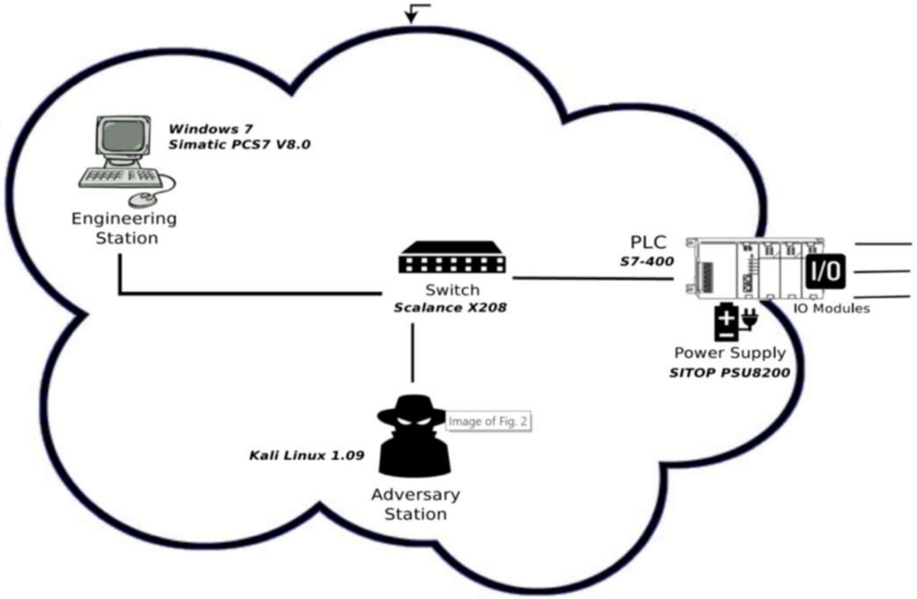
By **Steven Cherry**

PLC Network Security



[Ghaleb18]

PLC Network Security



Examples of attacks:

- **Replay attack** – three steps (i) make, or wait, the engineering station send one command (start, stop, etc) to the PLC, (ii) capture the packets, (iii) replay the captured packets at a later time. *(Likely accepting access from any IP.)*

- **Man in the middle attack** – Insert an attacker in between the engineering station and the PLC. Address Resolution Protocol (ARP) poisoned.

- **Stealth command modification attack** – Combination of the previous two methods. Interfering with sent commands by replaying other commands in a stealth way.

[Ghaleb18]

PLC Network Security – Challenges & Solutions

Industry 4.0 / Cyber Physical Systems protection

- **Firewalls** – Avoid field-buses direct access. Run traditional **field-buses just at the factory** floor. PLC programming (LD / IL / ST / SFC) **still to run as learned** in this course.

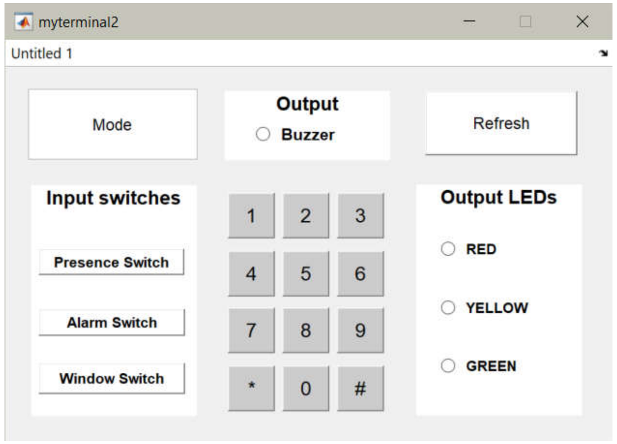
- **Hardware low level protection** – Separate local area networks (LANs) from the wide (global) area network (WAN). Use **dedicated hardware** and software to bridge the WAN to the LANs.

- **Industrial secure protocols** – Use knowledge, HW and SW from the telecommunications global market. **Open Platform Communications** (OPC), formerly OLE for Process Control (Object Linking and Embedding for …) already contains security features. OPC Unified Architecture (OPC UA), *extended to the field level may end the fieldbus wars [www19].*

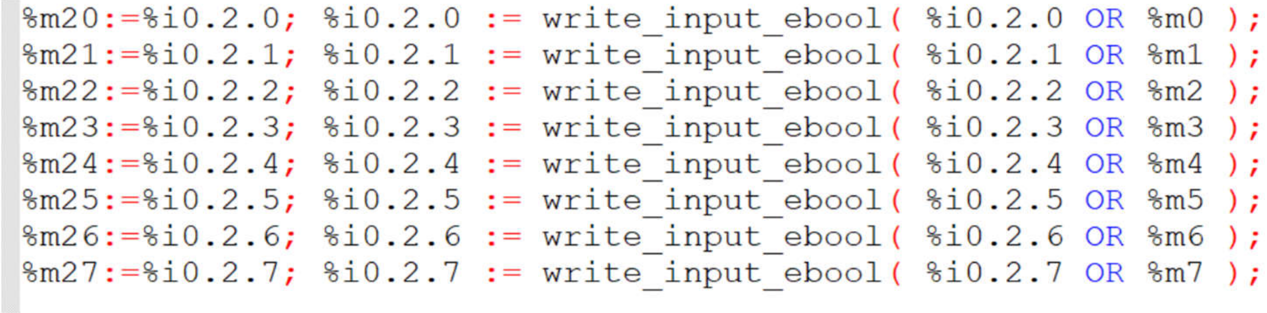
[www19] https://drivesncontrols.com/news/fullstory.php/aid/5851/Extending_OPC_UA_to_the_field_level__91will_end_fieldbus_wars_92.html (accessed Mar2019)

An example of IO overwriting with good purposes

Simultaneous real and simulated IO allows debugging keyboard reading without the real keyboard.

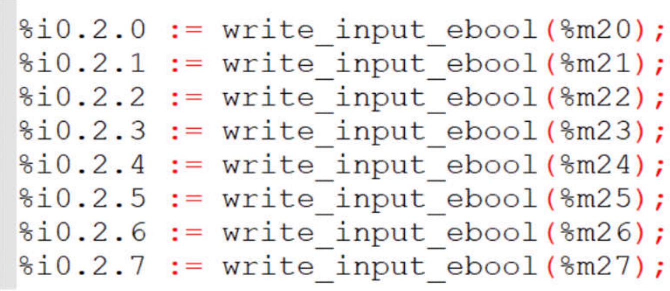


Init of scan cycle, use %m0..%m7 to *overwrite* real hardware inputs.

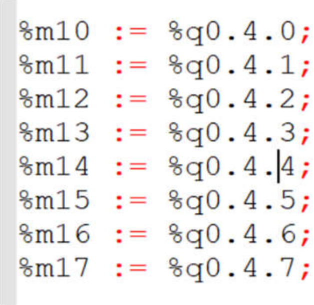


```
%m20:=%i0.2.0; %i0.2.0 := write_input_ebool( %i0.2.0 OR %m0 );
%m21:=%i0.2.1; %i0.2.1 := write_input_ebool( %i0.2.1 OR %m1 );
%m22:=%i0.2.2; %i0.2.2 := write_input_ebool( %i0.2.2 OR %m2 );
%m23:=%i0.2.3; %i0.2.3 := write_input_ebool( %i0.2.3 OR %m3 );
%m24:=%i0.2.4; %i0.2.4 := write_input_ebool( %i0.2.4 OR %m4 );
%m25:=%i0.2.5; %i0.2.5 := write_input_ebool( %i0.2.5 OR %m5 );
%m26:=%i0.2.6; %i0.2.6 := write_input_ebool( %i0.2.6 OR %m6 );
%m27:=%i0.2.7; %i0.2.7 := write_input_ebool( %i0.2.7 OR %m7 );
```

End of scan cycle, get back the real hardware inputs to use in the next scan cycle. Report outputs to %m10..%m17.



```
%i0.2.0 := write_input_ebool(%m20);
%i0.2.1 := write_input_ebool(%m21);
%i0.2.2 := write_input_ebool(%m22);
%i0.2.3 := write_input_ebool(%m23);
%i0.2.4 := write_input_ebool(%m24);
%i0.2.5 := write_input_ebool(%m25);
%i0.2.6 := write_input_ebool(%m26);
%i0.2.7 := write_input_ebool(%m27);
```



```
%m10 := %q0.4.0;
%m11 := %q0.4.1;
%m12 := %q0.4.2;
%m13 := %q0.4.3;
%m14 := %q0.4.4;
%m15 := %q0.4.5;
%m16 := %q0.4.6;
%m17 := %q0.4.7;
```

See the full example in the Unity Pro project:
http://www.isr.tecnico.ulisboa.pt/~jag/course_utils/plc_log/mix_io_show_strings.zip