U.PORTO
FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

Comunicações Industriais

Industrial Communications

2022/2023

2nd lab assingment

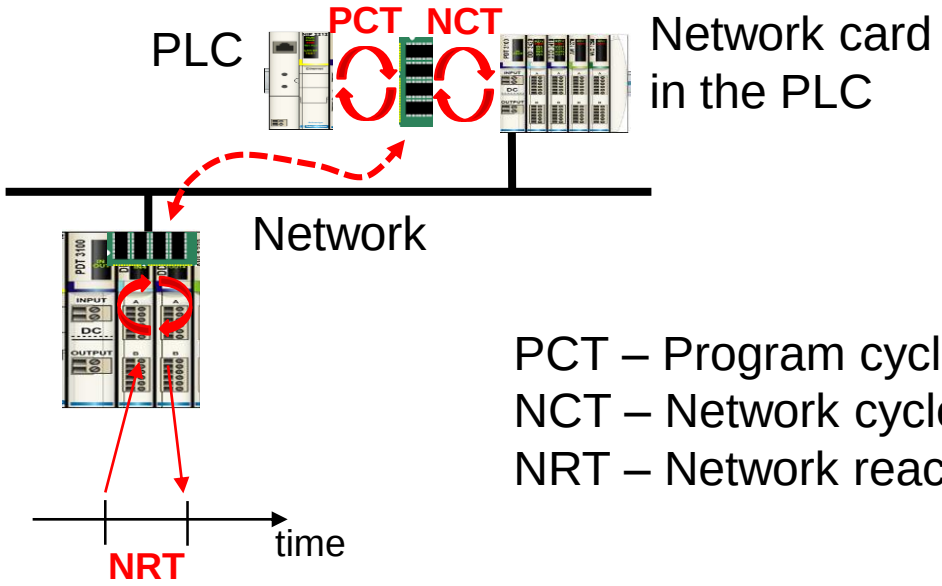
Network delay over fieldbuses

Learning objectives

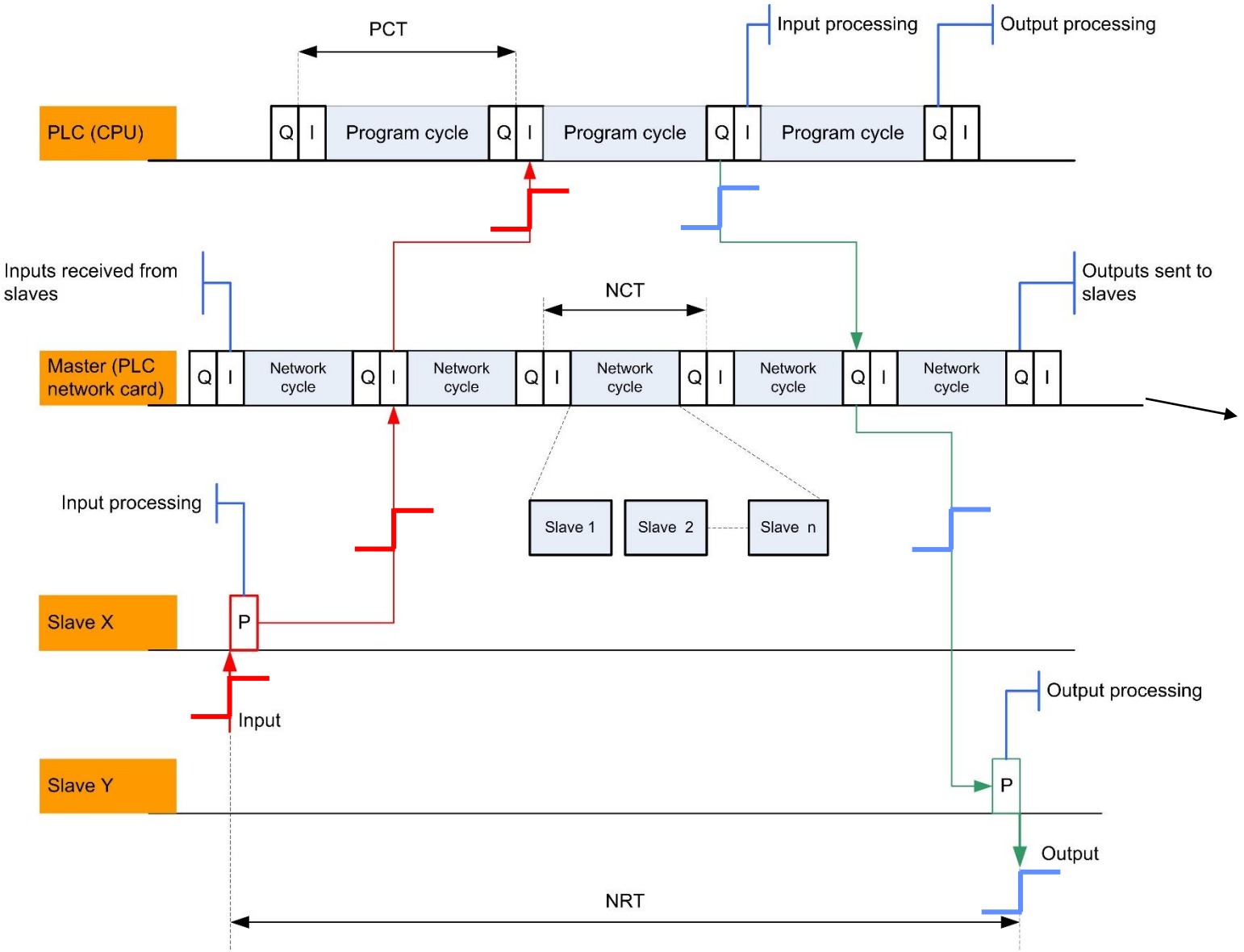
- Understand the use of industrial networks in practice and their configuration.
- Observe the typical cyclic behavior of industrial applications and its impact on the end-to-end delay of signals that travel over industrial networks.
- Acquire the capacity to use industrial networks.

General principles of cyclic operation

- **PLCs work in cycles**, reading/writing from/to memory
- **Network** cards in PLCs operate as communications **masters**
  - **Network masters** also **work in cycles**, reading/writing from/to slaves
- Communication happens through **memory to memory copies**



PCT – Program cycle time
NCT – Network cycle time
NRT – Network reaction time



The update cycles of inputs (I) and outputs (Q) may be separated and not back-to-back as in the figure

Objective

- Given the NRT (measurements) and the PCT and NCT (configuration)
  - Propose an **empirical model** for the **Maximum** and **Minimum NRT** as a function of **PCT** and **NCT**

NRTmax = f1 (PCT, NCT) **NRTmin = f2 (PCT, NCT)**

- If you have a **requirement** on **NRTmax**
  (for example, 40ms to switch off a motor after the activation of a detector)

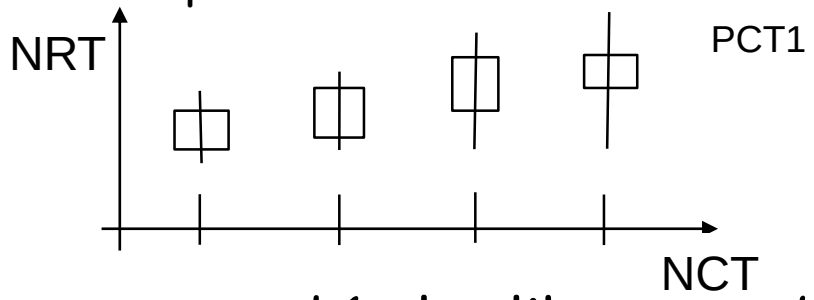
which PCT and NCT would you use?

Experimental campaign

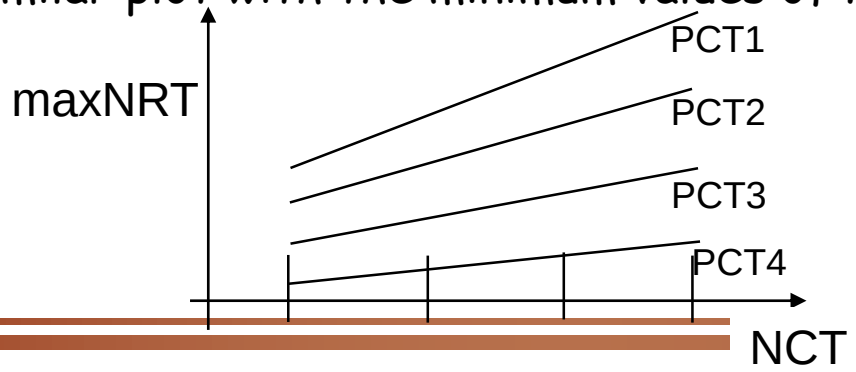
- We will use **2 networks**, ModbusTCP/Ethernet and CANopen/CAN
- For **each network**, configure the experimental setup with a (PCT,NCT) pair and obtain **~100 samples of NRT**.
- We suggest using (PCT,NCT) values that are reasonable in practice. In particular, consider the following **4 values** for each of the cycles, PCT and NCT: (*), **5ms, 20ms, 40ms and 60ms**.
  - * in CANopen, try also Async PDOs in place of the NCT
- Set **one value for PCT** and measure with the **four values of NCT** in sequence. Move to the **next value of PCT** and **repeat** all four values of NCT, etc.
- For **each measurement**, save all the NRT values in a **file** with a convenient name, e.g. "**PCTxxNCTxx**"
  - xx are the actual values used

Observing the results

- We recommend generating **4 plots**, one for each value of PCT
  - Use a **box-plot format**. In the Y axis represent the NRT measurements. In the X axis represent the 4 NCT values

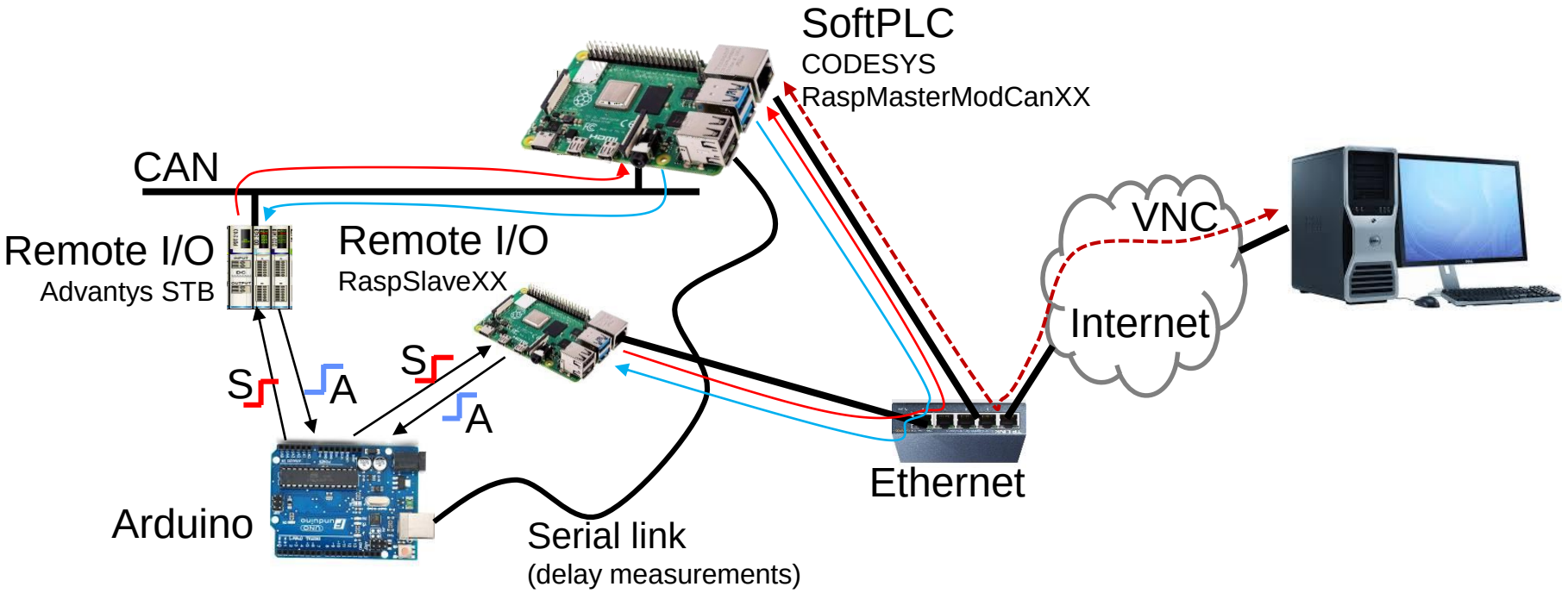


- Then we also recommend 1 plot like above but just with the **maximum values of NRT** (maxNRT). In this plot you can include four lines corresponding to the four values of PCT
  - And a similar plot with the minimum values of NRT (minNRT)



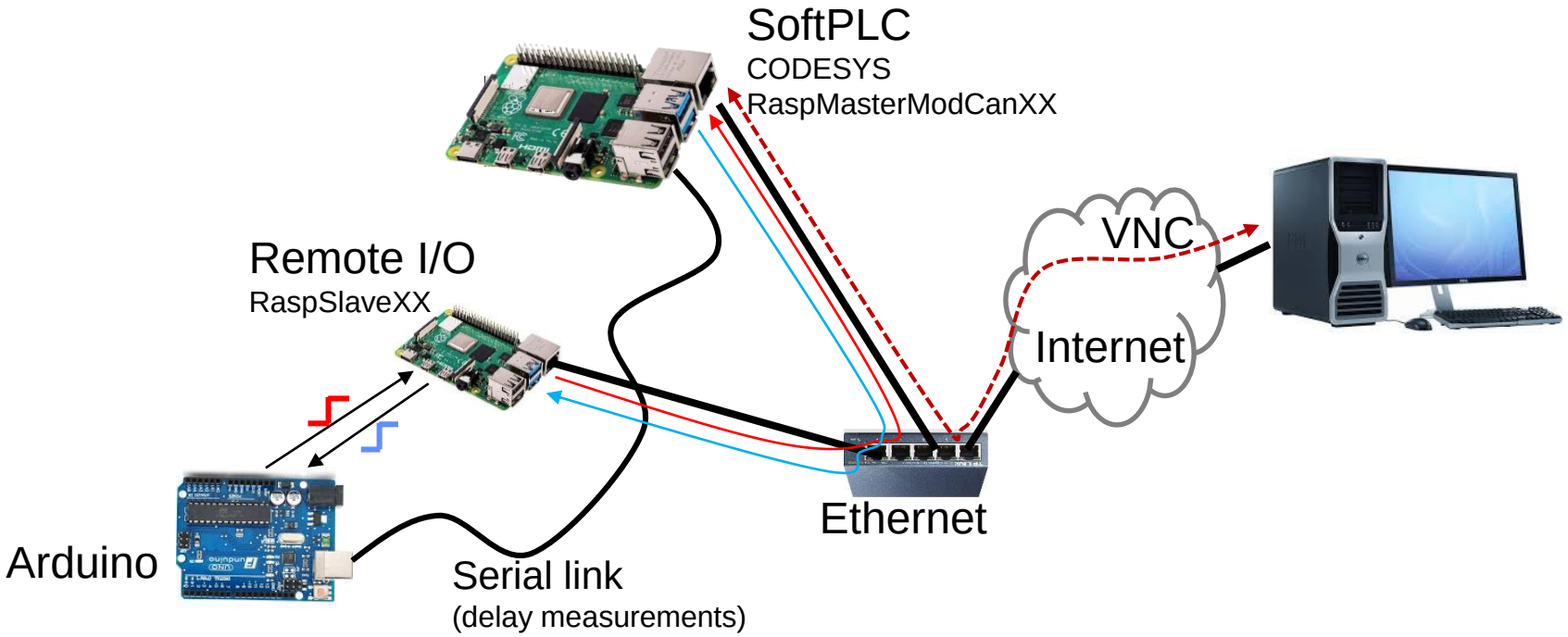
Experimental setup

- **SoftPLC** communicates with **Remote I/O** over **CAN or Ethernet**
- **Arduino** triggers event (0→1 transition) in Remote I/O **input (S)**
- **PLC reads** Remote I/O and **writes** the same signal on **an output (A)**
- **Arduino** measures the **time** between the transitions on **S and A**



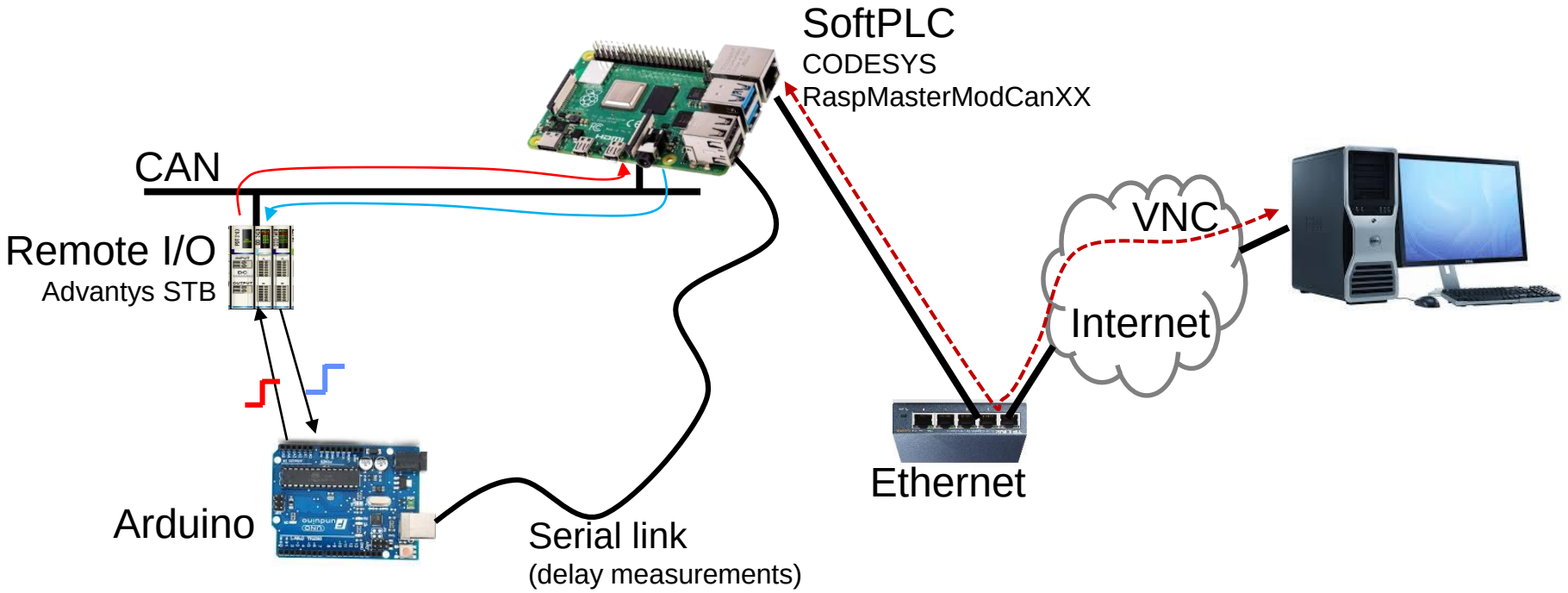
Experiments with ModbusTCP/Ethernet

- Use **VNC** to communicate with **RaspMasterModCanXX**
- Configure **Codesys** to use **ModbusTCP/Ethernet**
- Send **Arduino** (serial link) a command ('m') to trigger measurements with Modbus through the **RaspSlaveXX**



Experiments with CANopen/CAN

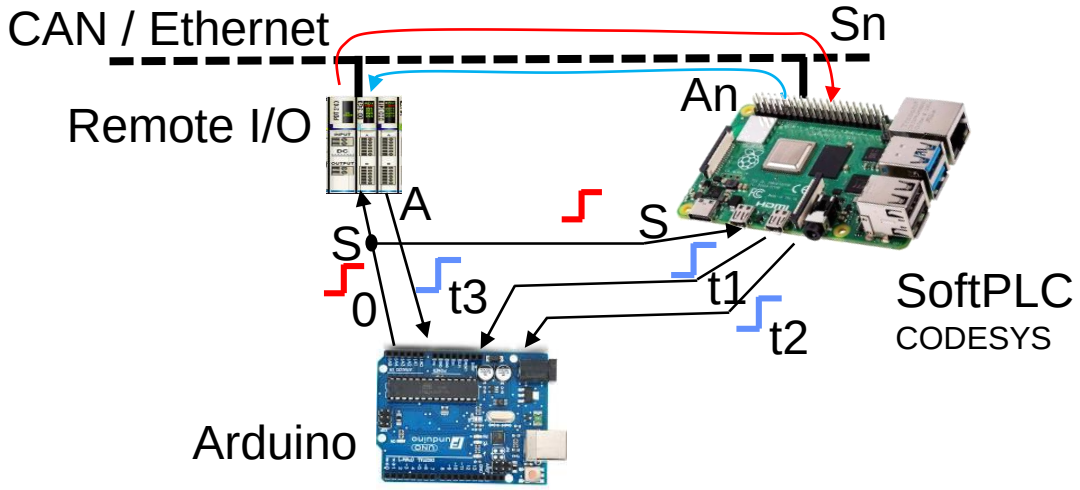
- Use **VNC** to communicate with **RaspMasterModCanXX**
- Configure **Codesys** to use **CANopen/CAN** (use the right **EDS**)
- Send **Arduino** (serial link) a command ('c') to trigger measurements with CANopen through the **Advantys STB**



Getting more timestamps within the NRT

To better understand the **propagation** of the signals through the network, we will add **two intermediated signals** captured in the master

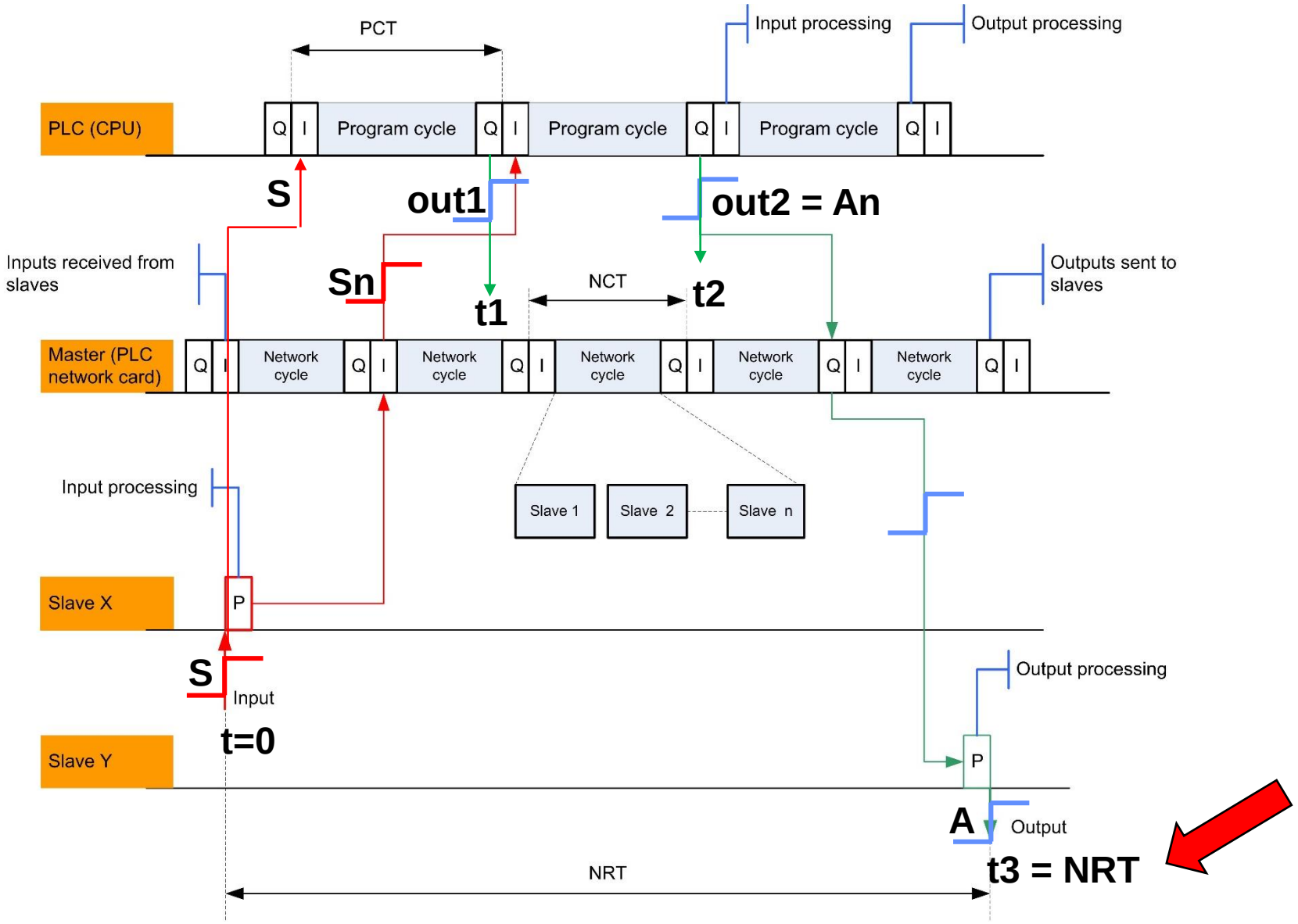
- Input signal **read directly** by the master (S→out1/t1)
- Input signal **read by the Master** through the network (Sn→out2/t2)
- Intermediate signals are captured and **timestamped** by the Arduino. Each Arduino reading contains **three** timestamps **<t1,t2,t3>**



Code for the Master

(* Master reads S directly*)
t1 → out1:=S; (*no network *)
(* Master reads S through the network *)
t2 → out2:=Sn;
(* Master writes A through the network *)
t3 → An:=Sn; (* = NRT *)

(see next slide)


PCT
Input processing
Output processing
PLC (CPU)
Q
I
Program cycle
S
out1
out2 = An
Inputs received from slaves
Sn
NCT
t1
t2
Outputs sent to slaves
Master (PLC network card)
Network cycle
Input processing
Slave 1
Slave 2
Slave n
Slave X
P
S
Input
t=0
Output processing
Slave Y
A
Output
t3 = NRT
NRT

General procedure

- **Configure** network (ModbusTCP/CANopen) in the SoftPLC (**Codesys**)
- **Verify** network operation from SoftPLC (access to Remote I/O)
- Write **SoftPLC program**, download it and initiate operation
- Carry out multiple **measurements** sending Arduino the desired command and saving the **delay measurements** in the RaspMasterModCanXX
- Transfer the measurements to the **remote PC** (e.g. using email)

- Once all measurements are done, do the plots and try to come up with the **empirical models** corresponding to **f1** and **f2** in slide 5

- Analyse the process and propose corresponding **analytical models**