lamaPLC LabView

Laboratory Virtual Instrument Engineering Workbench (LabVIEW) is a system design platform and development environment for a visual programming language developed by National Instruments.

The graphical language is named “**G**”; not to be confused with **G-code**. National Instruments, not LabVIEW, initially developed the G dataflow language. LabVIEW is commonly used for data acquisition, instrument control, and industrial automation on various operating systems (OSs), including macOS, other versions of Unix, Linux, and Microsoft Windows.

LabVIEW integrates the creation of user interfaces (termed front panels) into the development cycle. LabVIEW programs and subroutines are referred to as virtual instruments (VIs). Each VI has three components: a block diagram, a front panel, and a connector pane. The last is used to represent the VI in the block diagrams of others, calling VIs. The front panel is built using controls and indicators. Controls are inputs: they allow a user to supply information to the VI.

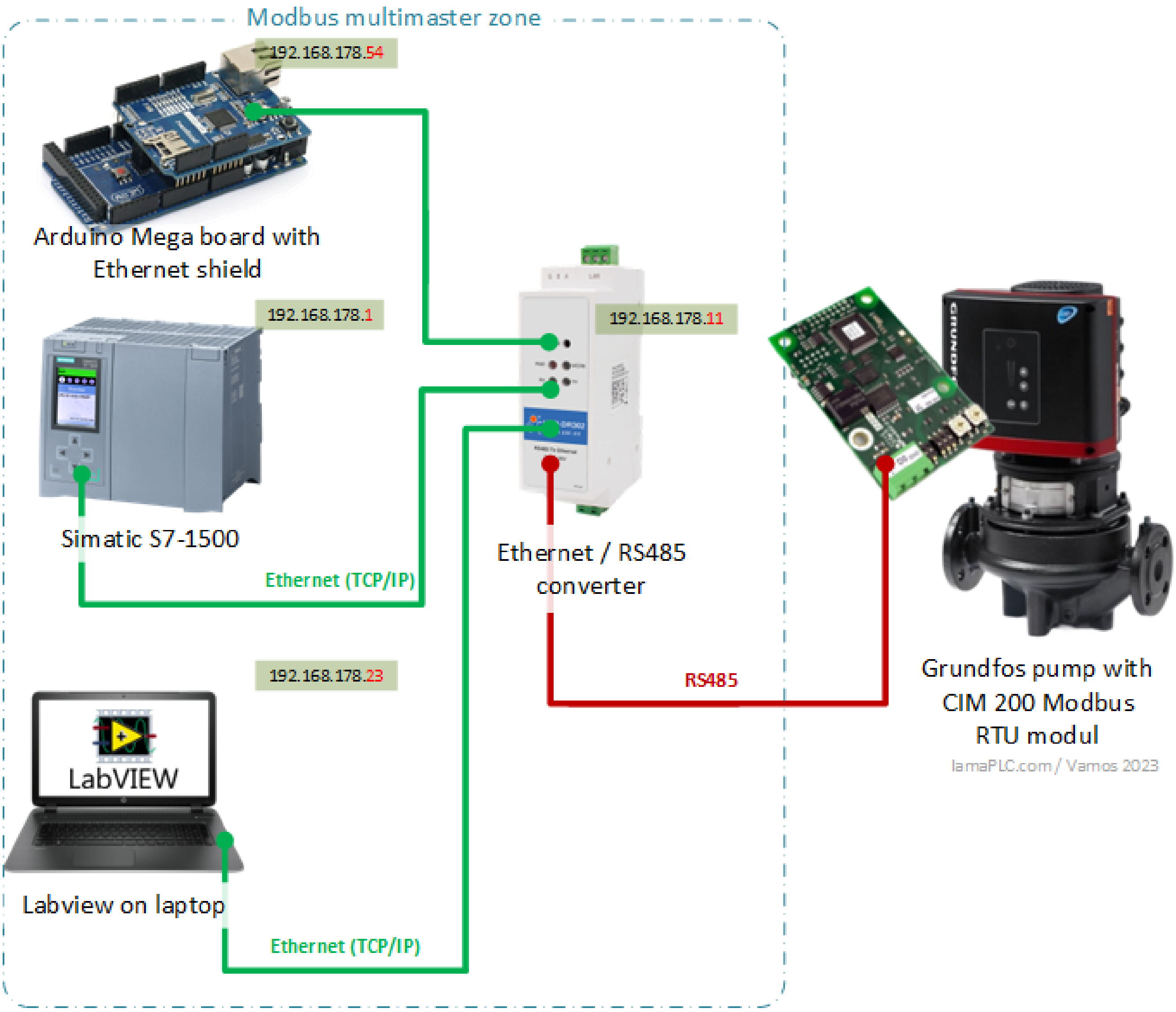
Indicators are outputs: they indicate, or display, the results based on the inputs given to the VI. The back panel, which is a block diagram, contains the graphical source code. All of the objects placed on the front panel will appear on the back panel as terminals. The back panel also includes structures and functions that perform operations on controls and supply data to indicators. The structures and functions are found on the Functions palette and can be placed on the back panel.

Collectively, controls, indicators, structures, and functions are referred to as nodes. Nodes are connected using wires; for example, two controls and an indicator can be wired to the addition function so that the indicator displays the sum of the two controls. Thus, a virtual instrument can be run as either a program, with the front panel serving as a user interface, or, when dropped as a node onto the block diagram, the front panel defines the inputs and outputs for the node through the connector pane. This implies each VI can be easily tested before being embedded as a subroutine into a larger program.

I personally use LabVIEW mainly for testing communication channels. It is often necessary to control sub-units with the Simatic PLC using Modbus communication. In many cases, it is not easy to set up the PLC correctly the first time, either; the sub-units can also hide surprises, in which case LabVIEW comes in handy. If I manage to control the sub-unit from here, it will also work from the PLC (in time). The example below shows the running of a Modbus test:

First Labview testprogram for Grundfos CIM 200 modul

In this case, the communication had to work between an S7-1500 PLC and a Grundfos CIM-200 module. In this case, Modbus multimaster communication is also possible due to the RS-485/Ethernet converter, allowing several masters to be connected simultaneously. Of course, if necessary, the pump can be controlled entirely from the laptop (as shown in the example):



This page has been accessed for: Today: 5, Until now: 10