IIoT Experimental Stand

[General description](#_gqag42wkz588)

[System overview](#_ukwpb2ykmpuf)

[Operating Principles](#_kuz7n6zaig1s)

[Pneumatic Stand](#_nc6v5xgj9063)

[Sensor System](#_gvbgl56ysjl3)

[Compute Area](#_8xn1dkd91z5b)

Connections

[PC and user functions](#_d76umn7dqntu)

[(!fun part!) ———How to use the stand————](#_n1mkqgj1yo3)

PME

Configuration

Code

[NodeRED](#_fq96x684fd33)

[Debug Node](#_etof08ee9xgx)

[Aventics Node](#_bnkl9icxw4rq)

[Aventics Node Modules](#_cs1j0g1w84t4)

[InfluxDB Integration](#_khycfyowurv1)

[Grafana](#_dgunyaxveer3)

[Basic graphing data](#_9h54vnq5vx5n)

[Basic tabling data](#_l92oebky2qhq)

[Components](#_grlx2x8p7x1q)

[Software used on the server (Tech Stack)](#_v26k6u5jfqfe)

[Photo gallery](#_1hos8lf12v4p)

[Pneumatic Stand](#_jcbjv5r2nc91)

[Sensors](#_6t2na4aqr37p)

[Microcontrollers](#_jjytmf9q6kz2)

[What is MQTT?](#_gimhronnq0oh)

[Role of Wi-Fi Micro Gateway for MQTT](#_z61t0lw5ow92)

# 

# 

# 

# General description

This is testing and research equipment made to gather data about pneumatic systems, and especially pneumatic pistons, capabilities, optimal parameters, limits, and the way normal wear affects its functionality.

It is meant to collect data, graph it, analyze it and perform predictive maintenance.

# System overview

# 

# Operating Principles

The RSTi-EP PACSystem controls the system and can be programmed with Proficy Machine Edition (PME). The EDGE controller collects and processes data, making it easy to access.

PACSystem connects to the distributor's inputs/outputs and sends data to the EDGE controller. Valves regulate the air flow, while sensors detect the piston position.

The AF2 IO-Link measures pressure/flow, filters air, and sends data to the BUS coupler. A cable connects the PACSystem to the BUS coupler.

## Pneumatic Stand

## 

Contents of the diagram

Compressor - AtlasCopco G2FF

Air preparation unit with a safety system

Valve 1 - Aventics 5/2 double solenoid valve, no detent

Valve 2 - Aventics 2x3/2 normally closed valve, no detent

Valve 3 - Aventics 5/3 double solenoid valve , closed center, no detent

Valve 4 - Aventics 5/2 single solenoid valve, spring return, no detent

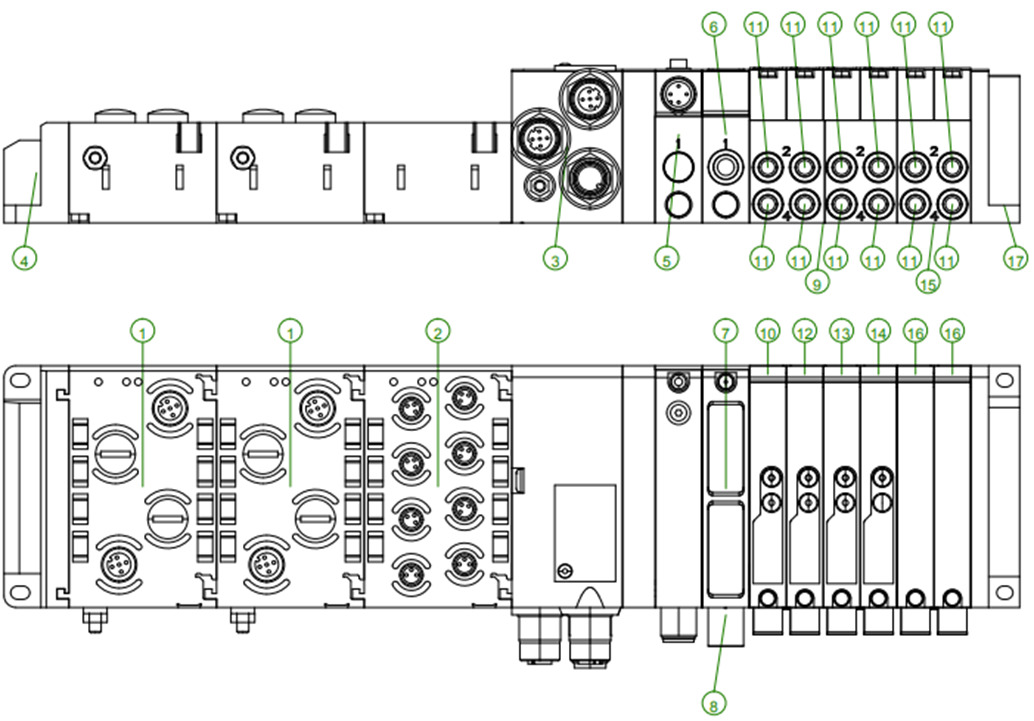
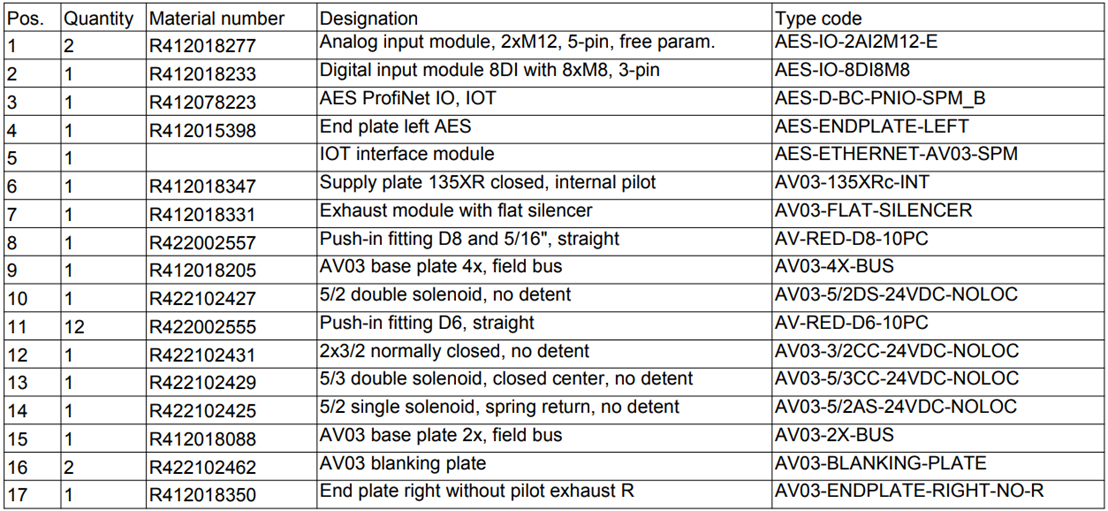
Valve 5 - Airtac 5/2 air actuated valve

Cylinder 1 - Aventics Ø63, Stroke: 500

Cylinder 2 - Aventics Ø32, Stroke: 500

Cylinder 3 - Aventics Ø63, Stroke: 300

Cylinder 4 - Aventics Ø50, Stroke: 300



The compressor adds pressurized air into the system. The air must be filtered and monitored for any irregularities in airflow.

Electronically controlled valves (valve 1 to valve 4) are used to control the air distribution to the cylinders.

An electric signal from the PACSystem to the BUS coupler tells the distributors on what position to be. The purpose of Valve 1 is to control the bigger Valve 5 which allows higher air flow to cylinder 1 for high speed tests, while the other valves (numbers 2 to 4) control the other cylinders directly, allowing for normal speed tests.

A BUS coupler gets information from the PACSystem and actuates the distributors.

## Sensor System

A pair of magnetic sensors are placed at each end of the cylinder to monitor the position of the rod. Being a digital sensor it only detects if the rod is extended or retracted.

The airflow sensor is analog and transmits the current airflow going through it.

## Compute Area

The brain of the system is the EDGE Controller. It gathers, stores and processes the data from the stand. It is connected to the network and can be accessed by any PC connected to the same network.

Edge computing is a distributed computing framework that brings computation and data storage closer to the sources of data.

Using the Rxi2-LP industrial PC we can use the data forwarded from the PAC unit to store, process, graph and analyze information.

For wiring together the hardware devices and transferring the data we use Node-Red and for visualization and graphing out the data we use Grafana.

A Programmable Automation Controller (PAC) is an industrial computer control system that continuously monitors the state of input devices and makes decisions based upon a custom program to control the state of output devices.

In our case, the PACSystems\* RSTi-EP CPE115 PAC, takes inputs from sensors and based on programmed logic it switches the solenoid valves (outputs) in the system to allow air to flow, actuating the pistons. It can be programmed using Ladder Logic, Structured Text, C or Function Block Diagram.

PACs are digital computers that hold and execute embedded programs. The PAC, compared to the PLC, is better suited for more complex automation solutions dealing with advanced process control, motion control, visualization and much more.

The software used for programming is Proficy Machine Edition

## Connections

The *RSTi-EP PACSystem* controls the behavior of the system and can be programmed using *Proficy Machine Edition(PME).*

The *EDGE controller* is used to gather data, process it, graph it and make it easily accessible.

The *BUS coupler* is used to connect the *PACSystem* to the outputs and inputs of the system as well as transmitting the data to the *EDGE controller*.

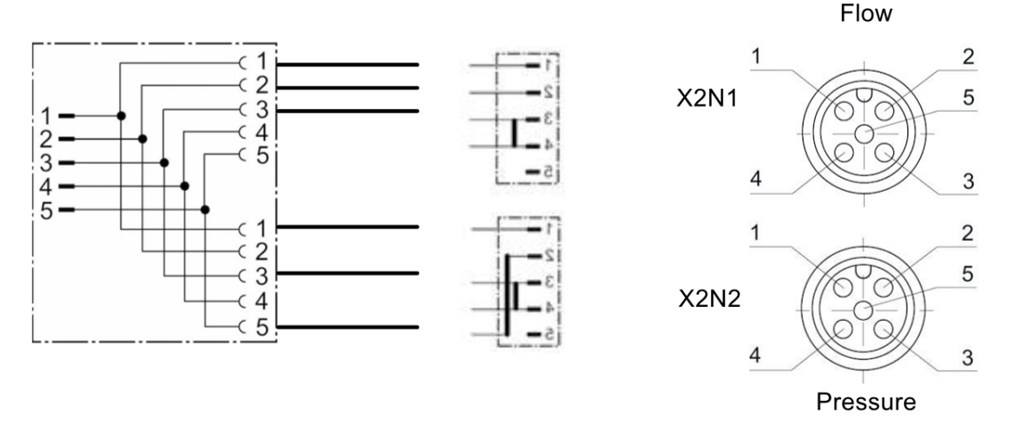
The *valves* control the flow of air, moving the *pistons*.

The *input*s are *sensors* connected to the extremities of the *pistons* for detection of the *rod's* position.

The *AF2 IO-Link* measures the pressure and flow , transmitting it to the *BUS coupler* and prepares the air by filtering it.

Connection between *PACSystem* and the *BUS coupler* is done via a M12-D to ethernet adaptor cable, from a LAN2 interface of the pac to the X7E1 port of the BUS and it makes connection between the programed logic and the inputs/outputs (sensors/valves).

Connection between AF2 is done via a M12-A splitter cable connected to the 2AI2M12-E I/O module for flow and pressure measurements.



Connection between the BUS coupler and the EDGE controller for analysis is done via a M12-A to USB cable.

The pneumatic cylinders are connected pneumatically via air tubes from the valves connected to the BUS coupler and electrically via magnetic proximity sensors mounted on the cylinders extremities that are connected to the 8DI8M8 digital inputs module.

Connection between a computer and the PACSystem is done by ETHERNET. After connection and addition of present IO-devices the project is ready for upload/download of any programmed logic.

## Inputs/Outputs

PACs have an open architecture and incorporate modular design, due to this the PAC cannot take inputs or outputs directly, it needs to have connected an I/O module. For this we are using the RSTi-EP PROFINET Network Adapter on the kart for its compact build and the more advanced AES bus coupler on the experimental stand.

HALL effect sensors are used to detect when the piston is at the extremities of the cylinder, so we know when to change the flow of air using the valves so that the piston keeps moving.

The AF2 flow sensor, IO-Link, is the first element in the circuit right after the FRL unit, it measures the flow and pressure in the system and returns the results as analog signals between 4 and 20 mA.

## PC and user functions

The user can program the whole system using the software Proficy Machine Edition.

Grafana is the main source of data analysis accessible by the user but NODE-Red can also be used to selectively view data gathered from the sensors or to see past operations.

Included in the monitoring suite, Emerson Aventics includes a piece of software for predictive maintenance.

# (!fun part!) ———How to use the stand————

The name of the game is FLEXIBILITY. It contains a lot of moving parts to play with and experiment.

## PME

### Configuration

1. Configure the PROFINET Controller

1. Set IP Address in the Inspector, this is recommended to be on a different subnet to the main network interface (eg controller LAN1 192.168.0.100, PROFINET LAN 192.168.1.100)

2. Set speed to 100Mbps (some devices don’t work at 1Gbps)

3. Configure IP Auto-Assign range (eg 192.168.1.101 to 192.168.1.199)

2. Configure the PROFINET IO Device under the PROFINET Controller

1. Right-click on the PROFINET Controller and select Add IO Device

2. Choose the correct device form the list of devices or import your own GSDML file

3. Assign a unique device number (1 to 255)

4. Give the device a unique device name (all lower case) (or make a note of the default if just using one device)

5. Note the IP address (auto-assigned from step 1.c above or assign your own IP address in the correct range)

6. Right-click on the IO device and select Change Module List then add modules as required

7. Double-click on the device and each of its modules and configure as required

3. Download the configuration to the controller

4. Configure the PROFINET IO device itself using PAC Machine Edition (some IO devices like RSTi-EP can also use their built-in web page to do this but PROFINET allows for the controller to send the configuration to the IO device when it powers up)

1. Connect the programming computer to the PROFINET Network (ie change Ethernet connections)

2. Change the programming computer’s IP address and subnet mask to match the PROFINET network

3. Start PAC Machine Edition in Administrator mode (right-click, run as administrator)

4. Open the project, then right-click on the PROFINET controller in the target and select “Launch Discovery Tool” to open the PROFINET DCP tool

5. Choose the correct network in the Connection pull-down (laptops often have more than one network connection) then Refresh Device List

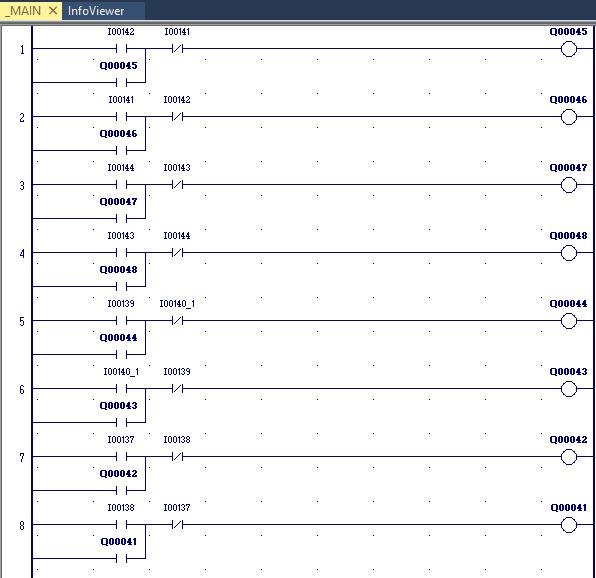
6. Double-click on a device to display its properties, then enter the correct device name and IP address configuration (click the identify device button to make its LEDs flash so you know you are configuring the correct device)

5. Once the application is downloaded and the PROFINET device is configured then you should be able to monitor inputs and control outputs (you will need to connect the programmer to the controller and change the IP address back to the correct IP address)

## Code

Ladder logic is a programming language that represents a program by a graphical diagram based on the circuit diagrams of relay logic hardware. Ladder logic is used to develop software for programmable logic controllers (PLCs) used in industrial control applications.

Ladder is intuitive, easy to learn with efficient representation of discrete logic but being so simplified it has its downsides.



The ladder code consists of blocks that control the extension or retraction of pistons, when a sensor is activated at one of the extremities of the cylinder a coil is turned on that changes the position and the respective valve and so the direction of the piston.

## NodeRED

Node-RED is a visual programming tool that allows users to create and connect hardware devices, APIs, and online services. It uses a web-based flow editor to drag and drop nodes, which represent different functionality, to create automated workflows. It simplifies the process of building Internet of Things (IoT) applications by providing a user-friendly interface for connecting and controlling devices.

### Debug Node

The debug node is a built-in Node-RED node that allows users to view the message passing through the nodes in real-time. It is useful for troubleshooting and understanding the flow of data through a flow.

Users can view the contents of the message, filter messages by type or topic, and output messages to the console or sidebar. It is easy to use by simply dragging and dropping it into the flow and connecting it to other nodes.

The green square on the right means it will debug. Pressing it will stop the node from debugging without deleting it.

Debug On:



Debug Off:



### Function Node

This node uses Javascript to transform data or use any other js function.

#### Air Pressure Translation Function

This function is used to translate the raw data from the sensor into Bars.

*msg.payload = msg.payload / 10000 \* 5.38*

*return msg;*

### Aventics Node

If the node doesn’t connect to the sensors:

1. It needs approximately 30-60s to connect
2. In the node properties panel it was selected as an unavailable module.
3. The program just needs to be redeployed just because.
4. The second coupler needs to be selected for the system to gather data from the sensor.

The node looks like this:

### 

### Aventics Node Modules

1. 01 Valve driver 4 valves - Controls the pneumatic distributors.
2. 03 IO-Module dig. (9DI8M8) - Digital sensors module. It is connected to 8 magnetic sensors each situated at the end of every piston on the stand. Each address of the module is a different sensor output. In the software they are numbered from 1 to 8 but on the plastic module they are labeled from 0 to 7.
3. 07 IO-Module ana. (2AI2M12-E) - Analog sensors module. It is connected to an airflow sensor situated at the beginning of the pneumatic system right after the air from the compressor is filtered and lubricated. Address 1 is pressure (needs this formula to translate it).

Parameters:

* Topic - Cosmetic name of node for internal use only. It doesn’t affect other functionality.
* Function - Determine how the data should be pulled from the sensor.
* From - Select which sensor it pulls data from. Each address is a different sensor in the selected module.
* Poll interval - The minimum delay between each reading. The sensor sends data each time it changes states. This will only delay the next reading. For x seconds it will not send any data after some data was sent.

Possible errors:

1. Aventics Node doesn’t always reconnect to the sensor: Go to the edge controllers container dashboard (Portainer) and restart mqtt-internal-ipc.

### InfluxDB Integration

The simple InfluxDB node takes whatever data it is given and along with a tag it is sent to the database along with a time stamp. [See chapter “Basic graphing data” for more details.](#_9h54vnq5vx5n)



Select the second Server for data to be logged.

Each time a new piece of data is added to the database it will automatically get a timestamp when it was logged.

Parameters:

* Measurement - Name of the date stored and used for identification in SQL.

### What if NodeRED breaks

#### Aventics Node doesn’t remain connected.

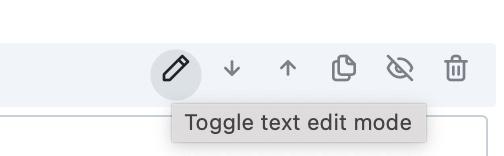
## Grafana

It is used to graph and monitor the data stored in InfluxDB.

### Basic graphing data

In the dashboard, click on the top bar of a graph and select edit.

In the lower half of the screen there will be an area to change the rules for the graph. Click on the pencil on the right side and delete the default code.



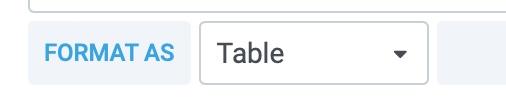
Replace it with:

*SELECT \* FROM “DataTypeToGraph”*

This will graph the data added to the database by a single InfluxDB node in Node-RED using the tag it was defined with.

### Basic tabling data

In a new dashboard, under the sql code, from the dropdown select “Table”. After that select Table view from the tab on the right called Format. This should table each entry in the database instead of graphing it.



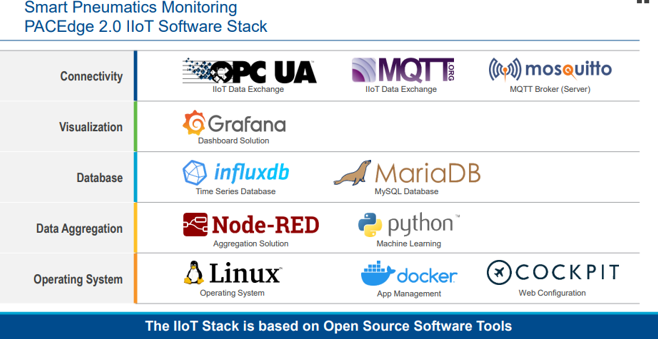
Use this SQL code to show only the last value recorded:

*SELECT last(value) FROM “Data”*

# Components

## Software used on the server (Tech Stack)

1. InfluxDB is the software where all the data is stored. It can be used with InfluxQL, which is a programming language similar to SQL for data processing.
2. Grafana is a plugin used to visualize and process the data collected on InfluxDB, as well as from other sources.
3. Node-RED is a visual programming tool that enables users to easily create event-driven applications by wiring together devices, APIs, and online services in a browser-based flow editor.



# Photo gallery

## 

## Pneumatic Stand

* Pneumatic Cylinder - 1
* Air filter and lubricant - 3
* Pneumatic 5/2 distributor - 5
* Valve System and IOT interface module- 10

## Sensors

* Magnetic proximity sensors at the end of each cylinder. Their purpose is to transmit the relative position of the piston rod in the cylinder. - 2
* Analog air flow monitor - 3
* A two-channel vibration sensor is used. The entire device contains a box with the brain and its power supply through 6 AA batteries. Two wires with one sensor attached at the end of each come out of it. - 9
* I/O modules, 2 analogic, 1 digital - 12

## Microcontrollers

* Wi-Fi Micro Gateway for MQTT Wireless receiver and transmitter box - 6
* PACedge - 7
* Edge Controller - 8

## What is MQTT?

MQTT (Message Queuing Telemetry Transport) is a networking protocol between sensors, devices, and a central server. Sensors can communicate through MQTT using Ethernet or Wi-Fi. MQTT is based on the idea of "publish and subscribe". Sensors can publish information, and other devices can subscribe to the sensors.

## Role of Wi-Fi Micro Gateway for MQTT

The Wi-Fi Micro Gateway for MQTT is an intermediate device between the server and the sensors. The gateway retrieves information from the sensors via MQTT and then transmits the information to the server.