

Raspberry PICO and IoT for RPC chamber slow control

C.Combaret ,L.Mirabito

Institut de Physique des 2 Infinis de Lyon

June 18, 2024

Slow control of RPC chambers in Lyon

Controls and measures

Our RPC chambers operation requires:

- High Voltage (< 10 kV) provided by HV modules plugged in CAEN (SY1527) or Wiener crates
- Low Voltage (6 V) provided by Lambda Genesys or Zup power supplies
- Brooks Mass flowmeters for the 3 used gas (TFE, CO₂/IsoButane, SF₆)
- Bosch sensors (BMP283, BME280) for Pressure and Temperature monitoring
- Honeywell humidity sensor (HIH8000)
- Weighing machine (ADAM CPW+) for gas bottle monitoring

Previous framework

Readout

- HV: CAEN → Socket TCP/IP, Wiener → SNMP , from a linux PC
- LV: Zup and Genesys → RS232 on raspberry Pi
- Brooks flowmeters → Standalone Windows or micro-controller soft provided by resellers
- P,T: BMP283,BME280 → SPI/I2C on raspberry PI
- H,T: HIH8000 → I2C on raspberry PI
- Weighing machine ADAMS not read

Software

- Dedicated Linux daemon for each hardware, interfaced with REST Services for command (START,STOP,STATUS,COMMAND)
- One central Linux service for user interface (via python scripts), MongoDB storage and feeding of GRAFANA data point
- GRAFANA monitoring display

Motivations to change

Constraints

- Hardware
 - No watchdog if one readout is stuck.
 - No service configuration in case of power supply cut or network short failure for example
 - Some hardware not accessible remotely
- Software
 - Central configuration frozen, not really flexible in case of multiple tests
 - All controls are based on computer with OS (update, security)
 - Some software non-free

Solution

hardware → Use a micro-controller Raspberry PICO wherever it's possible

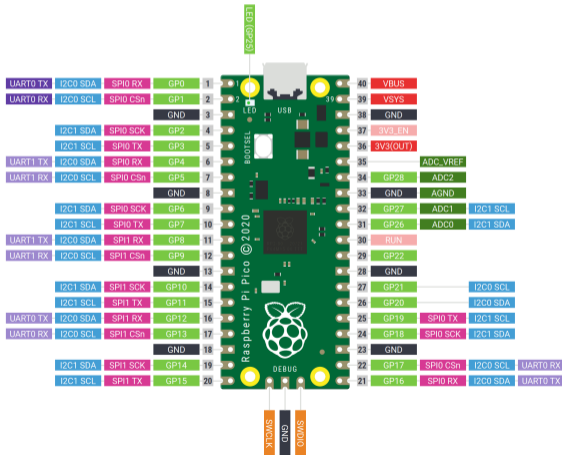
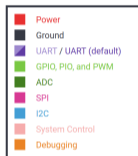
software Internet Of Things → MQTT

The Raspberry PICO

The Raspberry PICO is a micro-controller board developed by the Raspberry foundation:

- RP2040 double core at 133 MHz, Ram 264 Ko , 2 Mo of flash memory
- Low cost
- Programmable in C or python
- Large community of developers (hardware and software)

Connectivity



- 2 SPI, 2 I2C, 2 UART, GPIO, ADC, Timer
- μ RP2040 with embedded eeprom (0.2 Euros)
- USB, WiFi or Ethernet RJ45 TCP/UDP (wiznet W5500)

Two models

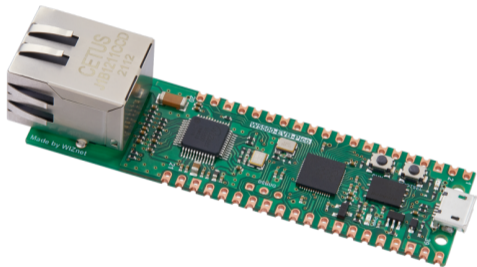


Figure: Ethernet 17 Euros

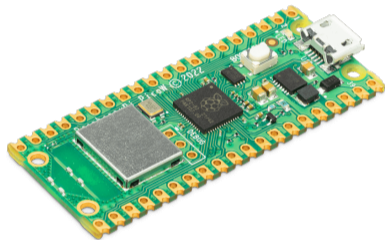


Figure: WiFi 8 Euros

Adaptation Board

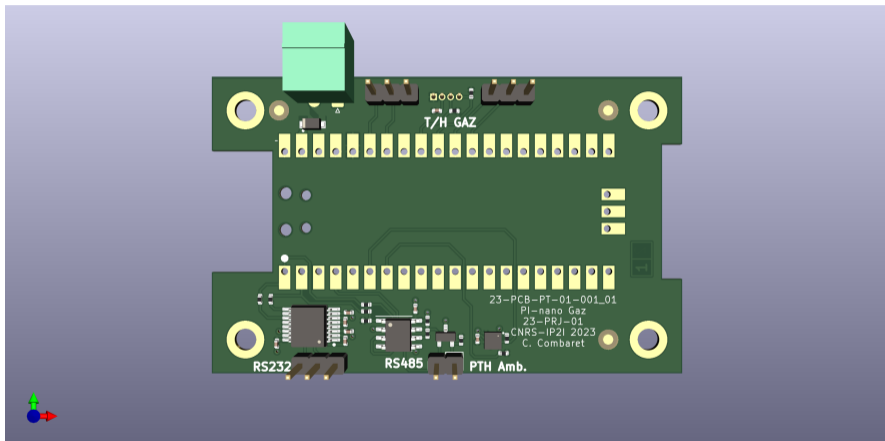


Figure: Versatile interface Board to our sensors. An additional LCD screen can be connected

Readout organization

hardware	Bus	Measures	device
Genesys / Zup	RS232	V,I	pico board
Brooks SLA5800	RS485	flow	pico board
BME280	SPI	P,T,H	pico board
HIH8000	I2C	H,T (Gas)	pico board
ADAMS CW+	RS232	Mass	pico board
Wiener HV	SNMP	V,I	Linux service on PC
SY1527 HV	TCP socket	V,I	Linux service on PC

P/T/H and LV

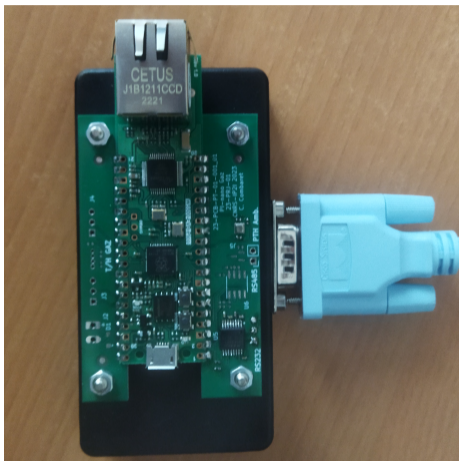


Figure: Adaptation board with an RS232 interface connected towards LV or weighing machine. An additional P/T/H BME280 sensor is connected to measure ambient pressure variation.

Gas relative humidity sensor

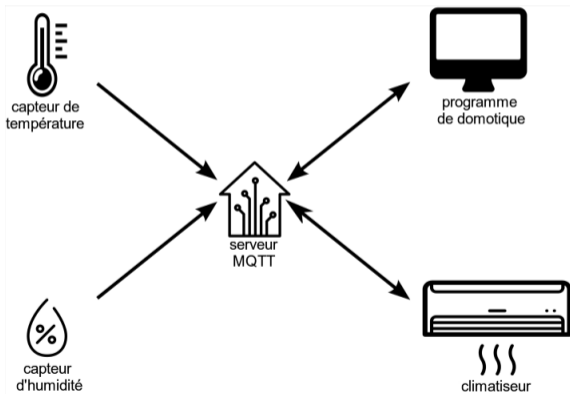
We used the HIH8000 sensor from Honeywell:



- The **HIH8000** is connected via a flat cable on top of the board.
- It is glued in a plastic tube inside the box.
- The box is then filled with epoxy to ensure the gas tightness.

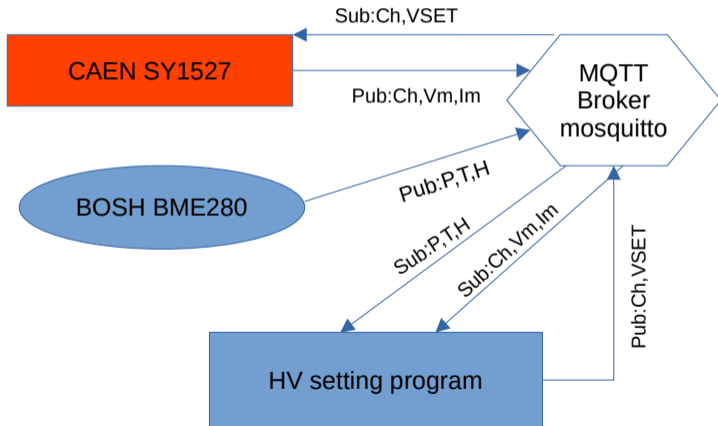
MQTT

Protocol of the Internet Of Things for home automation:



- The **broker** centralizes messages and redistributes them
- Sensors publish their data
- Air conditioner publish its status and subscribes to commands
- The home automation program subscribes to sensors data and publishes commands to the air conditioner

Application to HV control



Software framework

MQTT

- Library available in C,C++,python,JavaScript...
- Local Broker on linux → mosquitto deployed in docker

Code organization

- C++ for HV crates (TCPIP), python on micro-controllers
- Each client publish periodically its measurements and if appropriate, subscribes to commands (LVON,HVSET,FLOWSET....)
- One central linux daemon, subscribing to the status messages, is kept to save data in the MongoDB database

Programming environment

μ python

- Python implementation adapted to micro-controllers
- Libraries for each type of micro-controllers and buses (Ethernet, MQTT, SPI, I2C, UART, ADC...)
- Easy deployment and maintenance (Code copy in the Flash memory)

Commands

- Line command scripts : `mosquito_pub` `mosquitto_sub`
- Python script (Library Eclipse Paho)
- Web pages with MQTT JavaScript library

Monitoring

- Debug: MQTT explorer
- Visualization: plugin MQTT in GRAFANA

MQTT explorer

The screenshot displays the MQTT Explorer application window. The interface is divided into several sections:

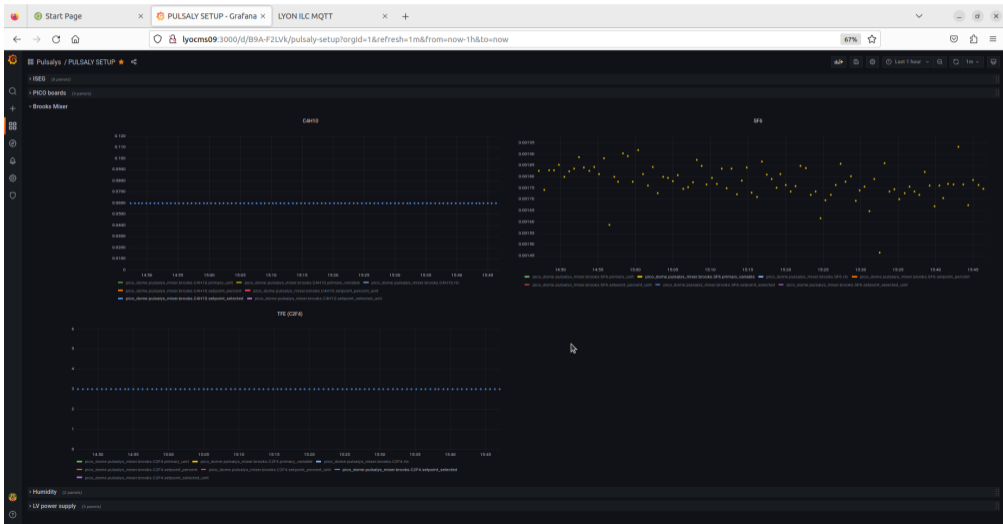
- Left Panel (Topic Tree):** A hierarchical tree view showing the MQTT topic structure. The selected path is `pico_dome / pulsalsys_mixer / brooks / SF6`. The tree includes topics like `test`, `pico_test`, `pico_dome`, `pulsalsys_inlet`, `pulsalsys_mixer`, `rp2040`, `brooks`, `GENSYS`, `pulsalsys_genesisys`, `dome_iseg`, `pulsalsys_outlet`, `cms_inlet`, `cms_outlet`, and `telescope_inlet`.
- Top Bar:** Contains the application name "MQTT Explorer", a search bar, a status indicator, and a "DISCONNECT" button.
- Right Panel (Message Details):** Shows the selected topic `pico_dome / pulsalsys_mixer / brooks / SF6`. It displays the message value as a JSON object:

```
{  "primary_unit": 138,  "primary_variable": 0.002069886,  "rtc": 1610416772,  "setpoint_selected_unit": 138,  "setpoint_selected": 0.06,  "setpoint_percent_unit": 57,  "setpoint_percent": 20}
```

. It also shows the QoS level (0), the timestamp (06/07/2023 1:32:26 PM), and a "History" section with 3 items.
- Bottom Panel (Graphs):** Two line graphs are displayed. The left graph, titled "pico_test/Bureau/bme", shows a pressure (P) value over time, with the y-axis ranging from 991.9 to 991.98. The right graph, titled "pico_test/Bureau/bme", shows a temperature (T) value over time, with the y-axis ranging from 26.79 to 26.85.

Figure: Open source debug tool for MQTT

GRAFANA Monitoring



Web pages with JavaScript interface

PICO boards MQTT interface

Data display is available in [GRAFANA](#)

Broker Brooks Low Voltage P/T/H Status HV control

Connection to broker

Hostname or IP Address and Port Number:

lyoilc07
8080

Location and system:

pico_dome

Connection: **Connect** Disconnect

```
Connecting to lyoilc07 on port 8080
Using the client id clientID-22
Subscribing to topic pico_dome/RUNNING
Message to topic pico_dome/LIST is sent
Topic:pico_dome/RUNNING Message : [{"devices":
["wiener"],"location":"pico_dome","subsystem":"dome_iseg"}
Topic:pico_dome/dome_iseg/wiener/INFOS Message : [{"commands":
["CLEARALARM","SET","OFF","ON","RAMPUP","RESET","STATUS","VSET"]}
Topic:pico_dome/RUNNING Message : [{"location":"pico_dome","subsystem":
"pulsalys_genesisys","devices":{"pp2040":"genesyys"}]}
Topic:pico_dome/pulsalys_genesisys/genesyys/INFOS Message : [{"id":"genesyys","cmds":
["SETON","SETADDRESS","SETOFF","SETVOLTAGE","SETCURRENT",
"SETREMOTE","CLEAR","VIEW","RESET","STATUS"],"rtc":1609459218}]
Topic:pico_dome/pulsalys_genesisys/pp2040/INFOS Message : [{"id":"pp2040","rtc":
```

PICO boards MQTT interface

Data display is available in [GRAFANA](#)

Broker Brooks Low Voltage P/T/H Status HV control

Brooks control

Brooks system pulsalys_mixer

Gas	Id	Max l/h	Flow set (l/h)	Flow read	New value (l/h)	view
C2F4	9057812	10	3.00000	0.09067	<input type="text"/>	<input type="button" value="Set"/> <input type="button" value="VIEW"/>
SF6	9057806	0.3	0.06000	0.00251	<input type="text"/>	<input type="button" value="Set"/> <input type="button" value="VIEW"/>
C4H10	9057816	1	0.06000	0.00062	<input type="text"/>	<input type="button" value="Set"/> <input type="button" value="VIEW"/>

Conclusion

Hardware

Using the Raspberry PICO with our adaptation board gave us a real Swiss knife to control and monitor most of our systems: Gas, LV, pressure, temperatures, humidity and mass.

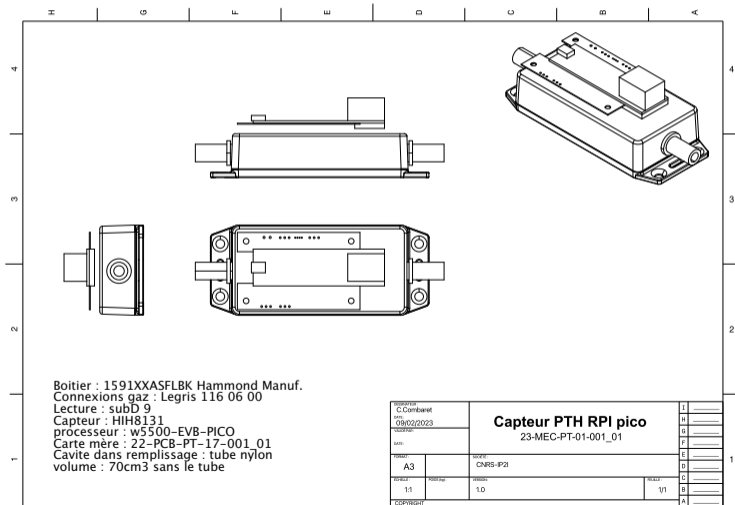
The cost of each system does not exceed a few tenth of Euros.

Software

MQTT is a light, versatile, and flexible solution to manage slow control process. A large set of development and analysis tools and library are available, on a large variety of platforms and languages.

Our code is available on github together with the adaptation board Kicad project.

Backup Gas relative humidity measurement (mechanic)



Backup: Web pages with JavaScript interface

PICO boards MQTT interface

Data display is available in [GRAFANA](#)

Broker Brooks Low Voltage P/T/H Status HV control

Zup and Genesys Control

Lambda system pulsalys_genesys

V Set	I Set	V read	I read	Status	V Req.	I Req.	Set Max Current	ON	OFF	VIEW
5.042	39.800	0.002	0.000	0						

PICO boards MQTT interface

Data display is available in [GRAFANA](#)

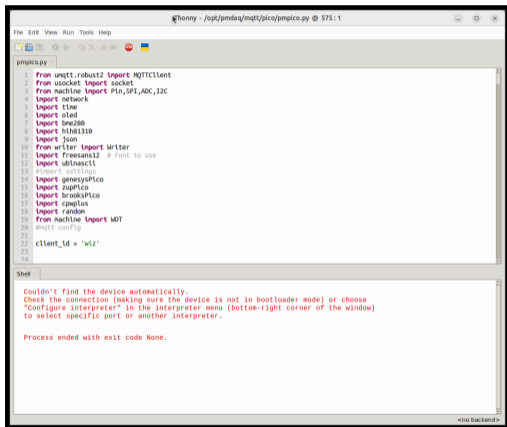
Broker Brooks Low Voltage P/T/H Status HV control

HV Monitoring

Wiener system dome_iseg

Channel	V Set (V)	I Set (uA)	Ramp Up	V read (V)	I read (uA)	Status	V Req. (V)	I Req. (uA)	Ramp req	Set Max Current	Set Ramp	ON	OFF	VIEW
0	6800.0	46.000	-47.0	-6800.1	-4.768	BITS: 80 00 80 outputConstantVoltage(16)								
1	6800.0	50.000	-47.0	-6800.2	-14.364	BITS: 80 00 80 outputConstantVoltage(16)								
2	6800.0	50.000	-47.0	-6800.1	-3.220	BITS: 80 00 80 outputConstantVoltage(16)								
3	6800.0	50.000	-47.0	-6800.2	-22.232	BITS: 80 00 80 outputConstantVoltage(16)								
4	6800.0	50.000	-47.0	-6800.1	-1.903	BITS: 80 00 80 outputConstantVoltage(16)								
5	6800.0	50.000	-47.0	-6800.1	-1.164	BITS: 80 00 80 outputConstantVoltage(16)								
6	6800.0	50.000	-47.0	-6800.2	-17.656	BITS: 80 00 80 outputConstantVoltage(16)								
7	6800.0	50.000	-47.0	-6800.1	-1.164	BITS: 80 00 80 outputConstantVoltage(16)								
8	6800.0	50.000	-47.0	-6800.2	-17.656	BITS: 80 00 80 outputConstantVoltage(16)								

Backup: Thonny IDE



```
thonny - /opt/pmdaq/mqtt/pico/pmpico.py @ 575:1
File Edit View Run Tools Help
pmpico.py
1 from umqtt.robust2 import MQTTClient
2 from usocket import socket
3 from machine import Pin,SPI,ADC,I2C
4 import network
5 import time
6 import oled
7 import bme280
8 import hts1310
9 import json
10 from writer import Writer
11 import fressans12 # Font to use
12 import ubinascii
13 #import settings
14 import geheysysPico
15 import zupPico
16 import brooksPico
17 import cpaplus
18 import random
19 from machine import I2C
20 import config
21
22 client_id = 'wiz'
23
24
Shell
Couldn't find the device automatically.
Check the connection (making sure the device is not in bootloader mode) or choose
"Configure interpreter" in the interpreter menu (bottom-right corner of the window)
to select specific port or another interpreter.

Process ended with exit code None.
<no backend>
```

- Python IDE
- Libraries for PICO and MQTT
- Debug of the micro-controller via USB

Figure: Python editor interfaced to raspberry pico