Laboratorio IoT

Prof. Paolo Napoletano
a.a. 2020/2021

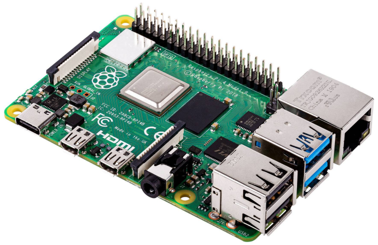
Oven Monitoring System

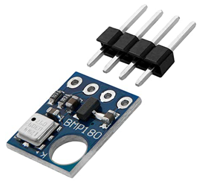
Manuel Zanaboni - 816105

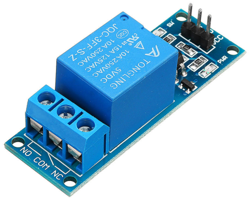
Materials

Description of the ingredients employed

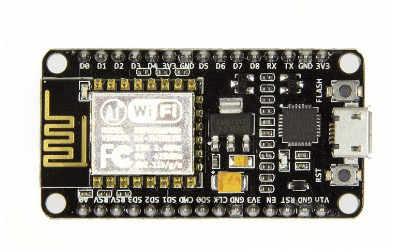
- RPi-oven-controller
  - RaspberryPi 4
  - n°1 10inches touchscreen monitor
  - n°4 MAX6675 thermocouples
  - n°2 BMP180 pressure sensors
  - n°12 5V relay

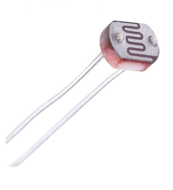


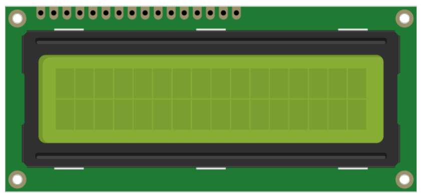




- Oven Monitoring System
  - ESP8266
  - n°1 liquidcrystal display
  - n°4 leds + 200Ω resistors
  - n°3 buttons
  - n°1 photoresistor + 10KΩ resistor

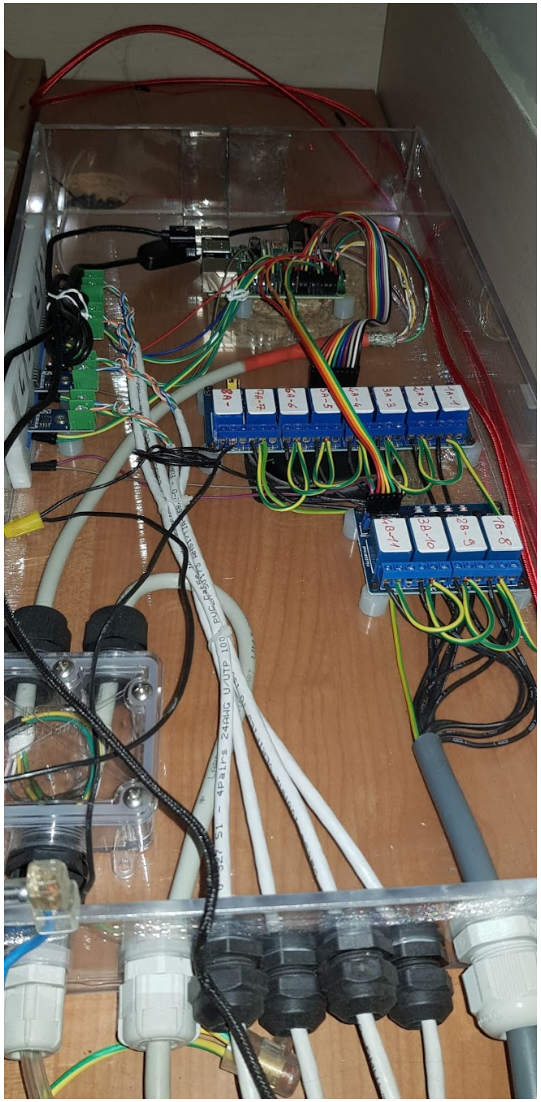






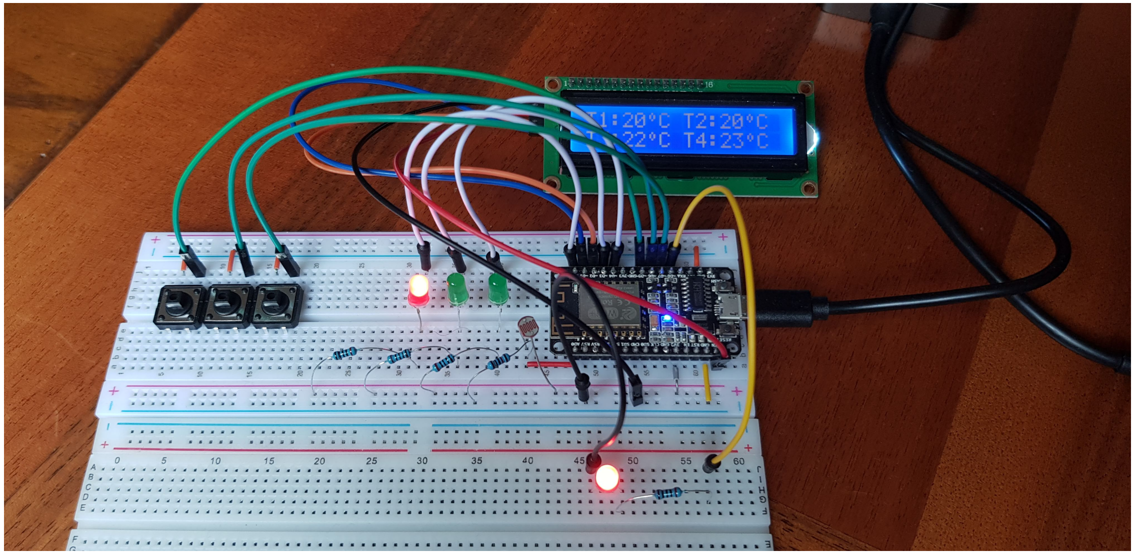
Materials

Description of the ingredients employed



- RPi-oven-controller

- Oven Monitoring System



Method

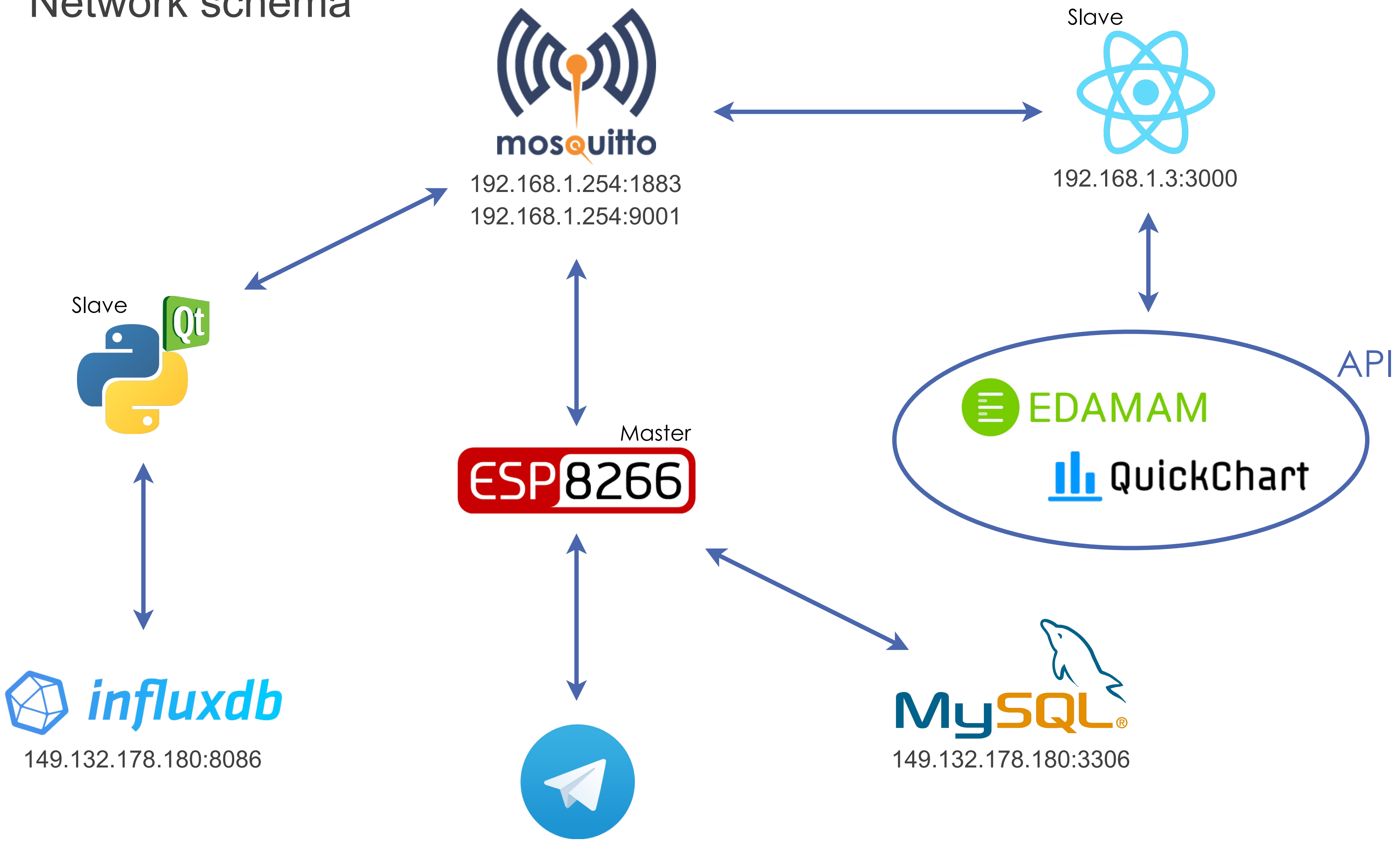
Description system: choices, parameters, use cases

- Overall system composed by 3 subsystems:
  - Python desktop app, deployed on a Raspberry Pi, which has its internal logic. It requests configuration from ESP8266, sends sensors data and internal state, can be controlled from React.js frontend and telegram bot. It logs data into InfluxDB.
  - ESP8266 acts as master for configuration injection. Displays data through the LCD and alerts through LEDS. It logs network events (connect/disconnect, controls) into MySQL, and handles telegram connection.
  - Web application, developed under React.js, listens on port 3000 and makes the monitoring possible. It can also control python desktop app state and exploits Edamam and QuickChart API services to propose a pizza recipe.
- Network:
  - Network communication is managed through MQTT by a Mosquito instance deployed on the Raspberry Pi. React.js frontend performs HTTP GET requests.
- Telegram:
  - Telegram bot is now supported.

Method

Description system: choices, parameters, use cases

- Network schema



Method

Description system: choices, parameters, use cases

- MQTT topics

mosquitto

ESP8266	PUB	oven/config/rpi	QoS 2 - retain: false
		oven/config/rpi/influxdb	QoS 2 - retain: false
		oven/config/frontend	QoS 2 - retain: false
	SUB	oven/config	QoS 2
		oven/sensors	QoS 0
		oven/state	QoS 1
		oven/control	QoS 1
		oven/telegram	QoS 0
Qt Last-will: **oven/config** - “rpi-bye”	PUB	oven/config	QoS 2 - retain: false
		oven/sensors	QoS 0 - retain: false
		oven/state	QoS 1 - retain: true
	SUB	oven/config/rpi	QoS 2
		oven/config/rpi/influxdb	QoS 2
		oven/control	QoS 1
Last-will: **oven/config** - “frontend-bye”	PUB	oven/config	QoS 2 - retain: false
		oven/control	QoS 1 - retain: false
		oven/telegram	QoS 0 - retain: false
	SUB	oven/config/frontend	QoS 2
		oven/sensors	QoS 0
		oven/state	QoS 1

Method

Description system: choices, parameters, use cases

- Telegram Bot functionalities
  - /help : list all available commands
  - /burnerON : turn the burner ON (if it’s not already ON)
  - /burnerOFF : turn the burner OFF (if it’s not already OFF)
  - /resistanceON : turn the resistance ON (if it’s not already ON)
  - /resistanceOFF : turn the resistance OFF (if it’s not already OFF)
  - /state : display oven state (lists burner, resistance and stepping)

It also gets notified for frontend/rpi connections and disconnections.

Proposed pizza recipe is also forwarded to the bot chat.

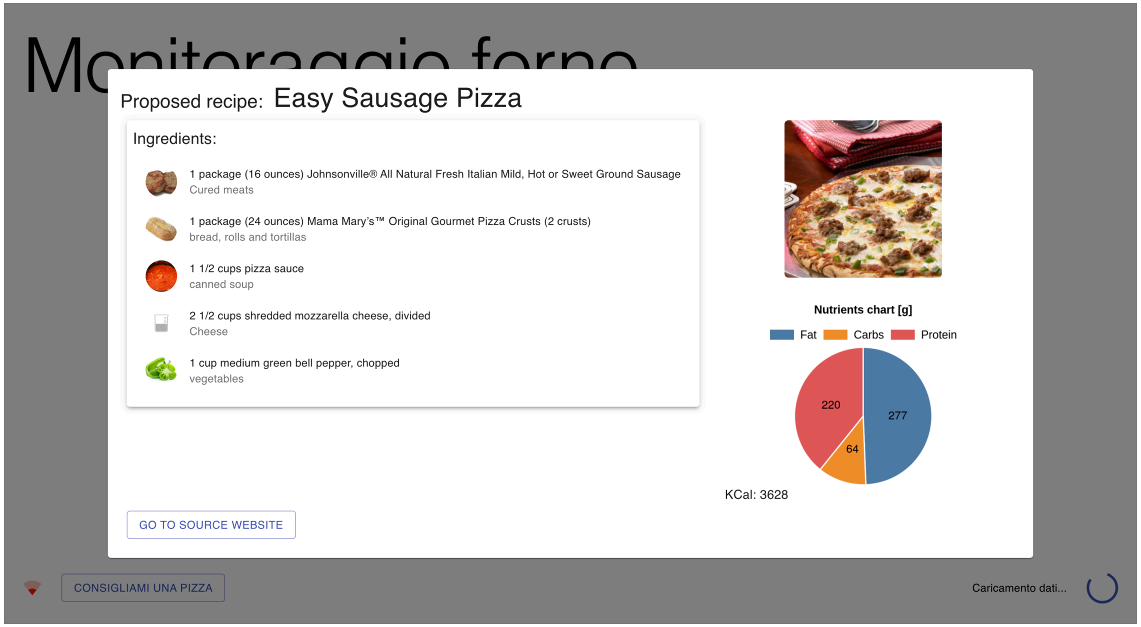
Method

Description system: choices, parameters, use cases

- API

EDAMAM — Food API endpoint, provides recipe search, nutrition analysis and so on… In this project it has been use to retrieve and propose PIZZA recipes.

QuickChart — Open source Chart Image API, exploits Chart.js library to generate chart images, PDFs and QR codes.



Monitoraggio forno

Proposed recipe: Easy Sausage Pizza

Ingredients:

- 1 package (16 ounces) Johnsonville® All Natural Fresh Italian Mild, Hot or Sweet Ground Sausage
  Cured meats
- 1 package (24 ounces) Mama Mary's™ Original Gourmet Pizza Crusts (2 crusts)
  bread, rolls and tortillas
- 1 1/2 cups pizza sauce
  canned soup
- 2 1/2 cups shredded mozzarella cheese, divided
  Cheese
- 1 cup medium green bell pepper, chopped
  vegetables

KCal: 3628

GO TO SOURCE WEBSITE

CONSIGLIAMI UNA PIZZA

Caricamento dati...

Method

Description system: choices, parameters, use cases

- Power consumption

Use case: system works for **4 hours/week**

Capacity of the battery: **2400mAh**

Regular power consumption: **120mA**

Total hours available = Capacity / Current

20 hours = 2400mAh / 120mA

↕

5 weeks

Modem-sleep power consumption: **15mA**

Sleep-interval: *100ms*	Work-interval: 1*0ms*
900 ms @ 15mA	3240 sec @ 15mA
100 ms @ 120mA	360 sec @ 120mA

1mA * 3600sec = 1mAh

15mA * 3240sec + 120mA * 360sec =

48600mA sec + 43200mA sec =

91800mA sec / 3600 sec = **25.5mA**

~94 hours = 2400mAh / 25.5mA

↕

23,5 weeks

Final remarks

Results, Discussion, conclusion

Results:

- Oven Monitoring System is accessible from every device connected to the local network.
- A “physical” monitoring system is available in the place where the ESP8266 will be positioned.
- MQTT protocol provide integral system state consistency.
- Sensors data is logged into InfluxDB, network events are logged into MySQL.
- Network communication is now bidirectional.
- Additional Telegram “interface” provided.

Next steps:

- Extend React.js frontend functionalities.
- Extend Telegram bot functionalities.

Final remarks

Results, Discussion, conclusion

DEMO TIME