

CHAPTER 2. Embedded Devices

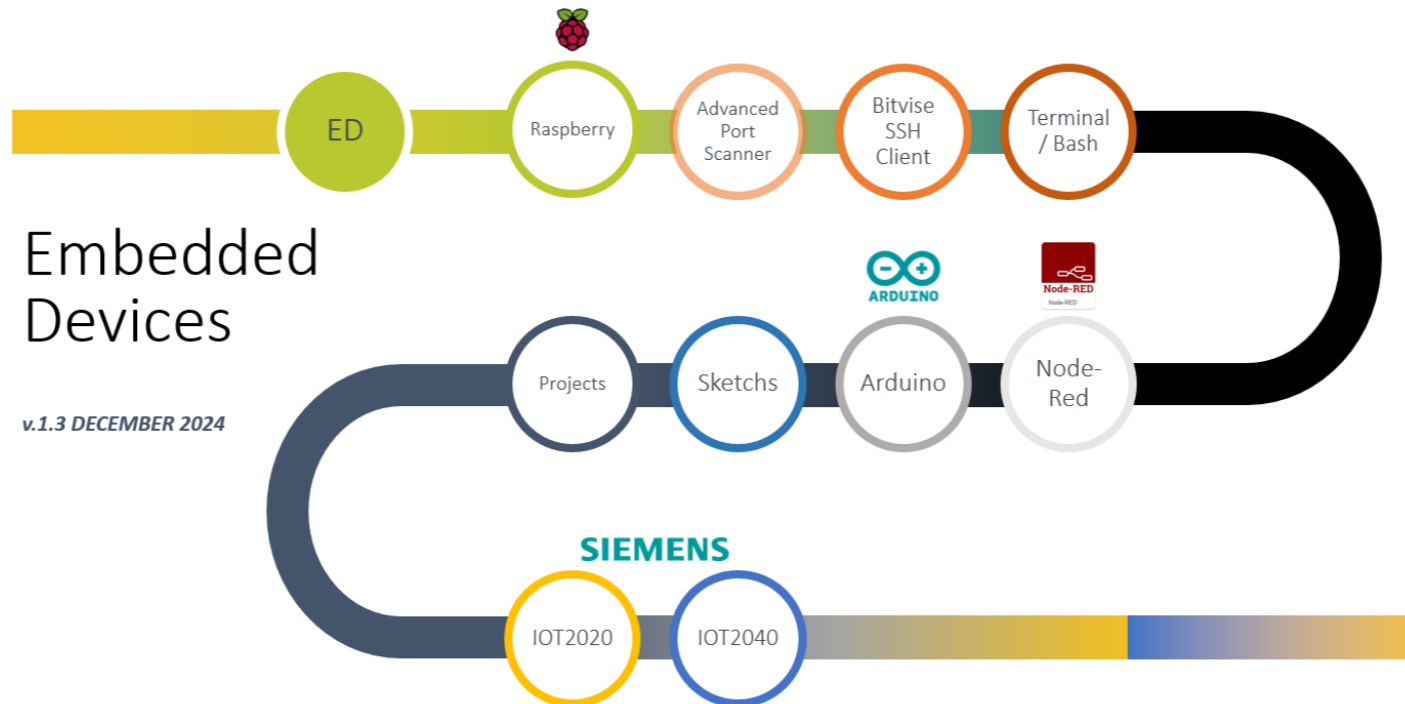
v.1.3 DECEMBER 2024



Ricardo Moraleda Gareta
[Director departamento
de software de GDO Software]

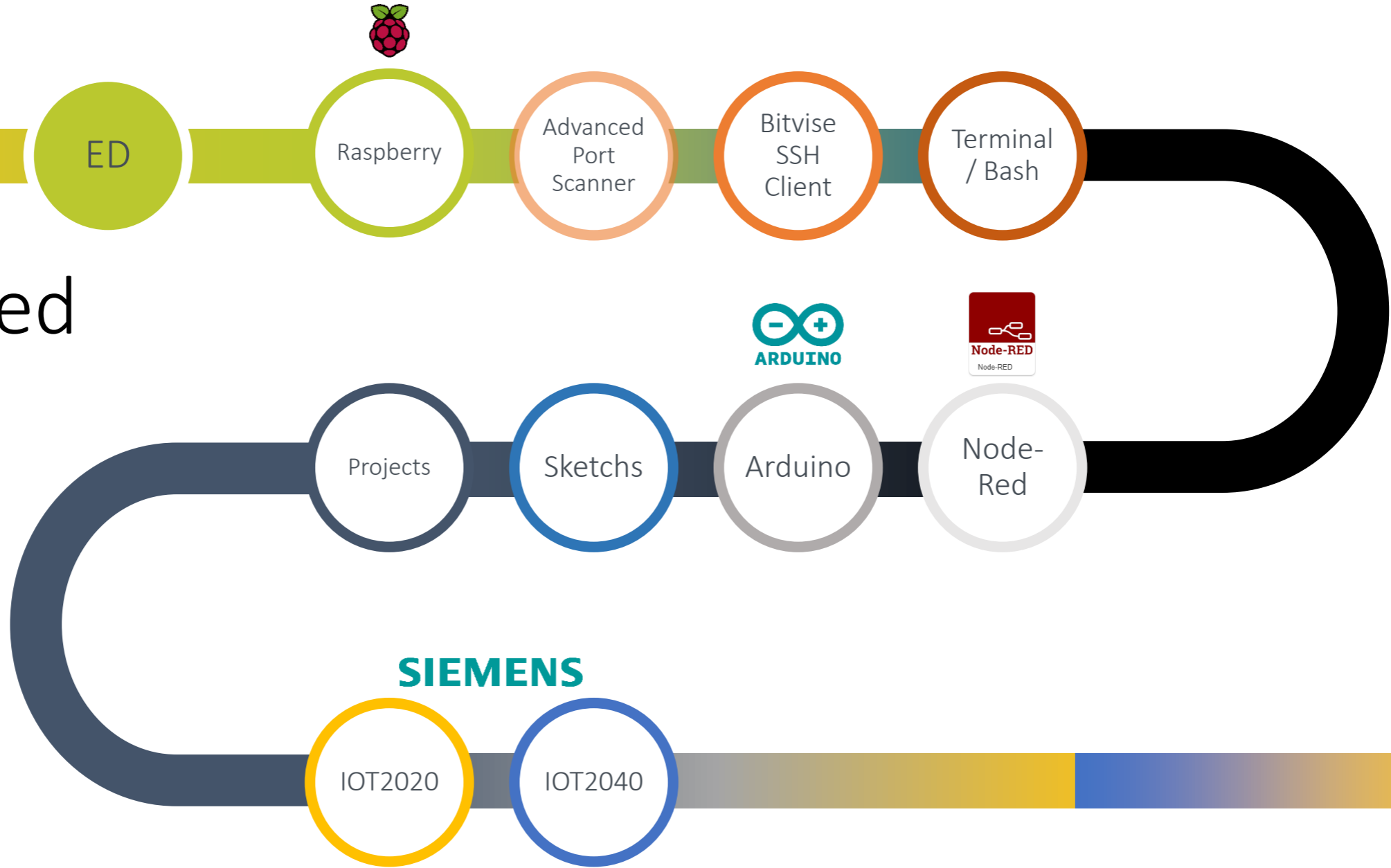


Melvin Francis Stephen
[Software Engineer
BE.services GmbH]



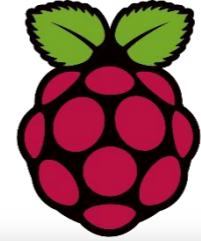
Embedded Devices

v.1.3 DECEMBER 2024





Raspberry



Raspberry Pi 4 B rev1.1

- With 64 GB microSD with Raspbian OS (based on Debian 11 "bullseye" – Linux)

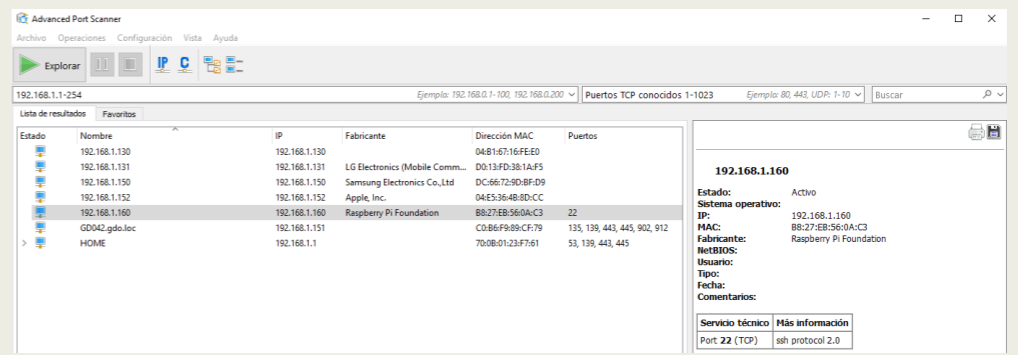
- WLAN/Bluetooth
- SoC
- 40 Pin GPIO Stiffleiste
- Gigabit-Ethernet
- MicroSD-Slot
- 2 x USB 3.0
- DSI Display
- 2 x USB 2.0
- USB-C Stromversorgung
- 2 x Micro HDMI
- CSI Kamera
- Audio/Video (analog)



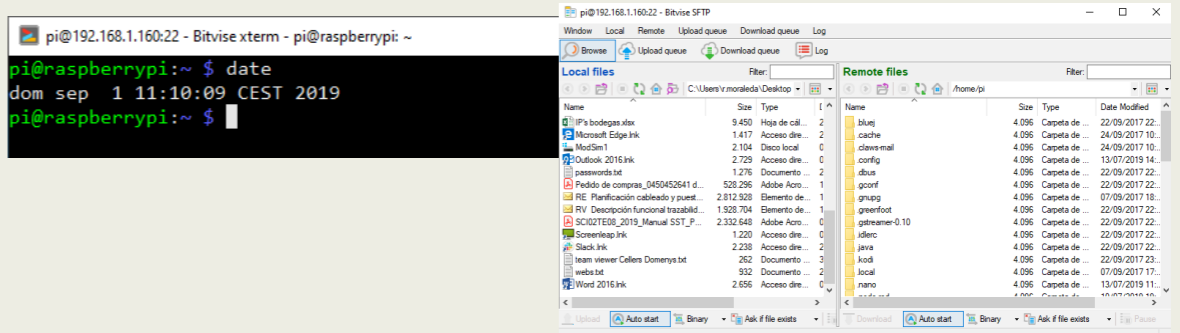
Its a mini PC !

PI 5 is now available in several modalities !! → <https://www.raspberrypi.org/>

With Advanced Port Scanner v.2.5.3581 PC software I detect the IP and ports

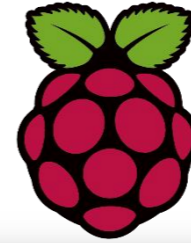


With Bitvise SSH Client PC software I connect via SSH (22) and SFTP (22) from the PC





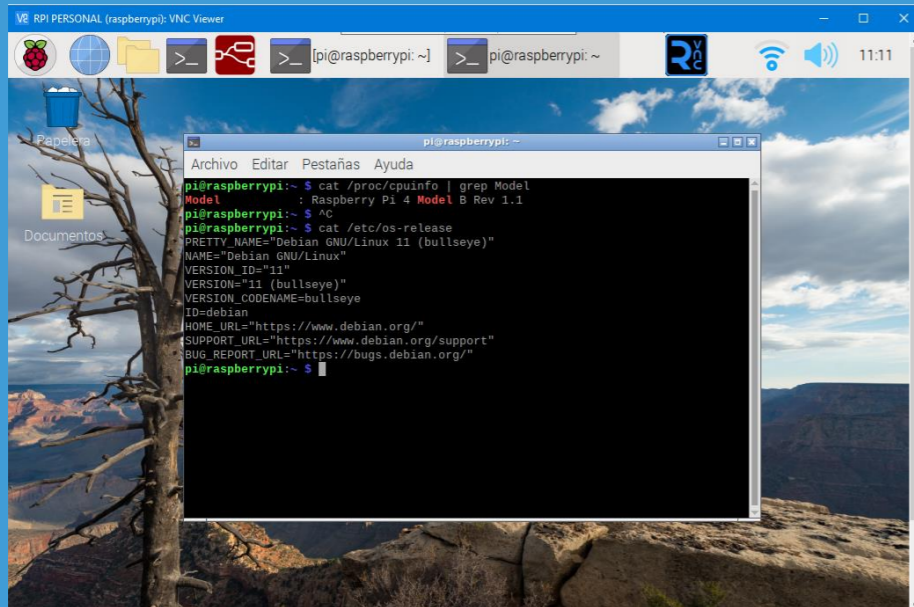
Raspberry



Raspberry PI 4 B rev1.1

To view the desktop-GUI remotely without having to be connected via HDMI directly, one solution is to connect via

<https://www.solveitc.com/tutoriales/article/4364-como-usar-comando-apt-linux/>



Terminal-Commands (Curiosities)

Sometimes it has happened to me that I periodically detect if a process is running and if not, restart it (for example). To do this a script (bash) and a scheduled task would be created.

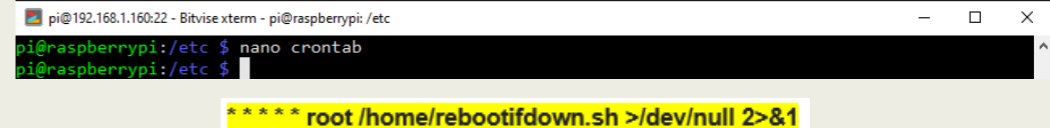
rebootifdown.sh

```
#!/bin/bash
#Scripts to start services if not running
ps -ef | grep node-red |grep -v grep
if [ $? != 0 ]
then
    reboot
fi
exit 0
```

1. Command ps -ef (list processes)
2. Grep node-red (check if the node-red process is present)
3. If not in the list → reboot

To make this text executable we need to do "chmod 755 rebootifdown.sh"

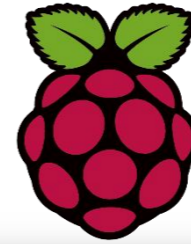
Add the following line to the crontab (en /etc):



Five asterisks at the beginning (* * * * *) means every 1 minute.

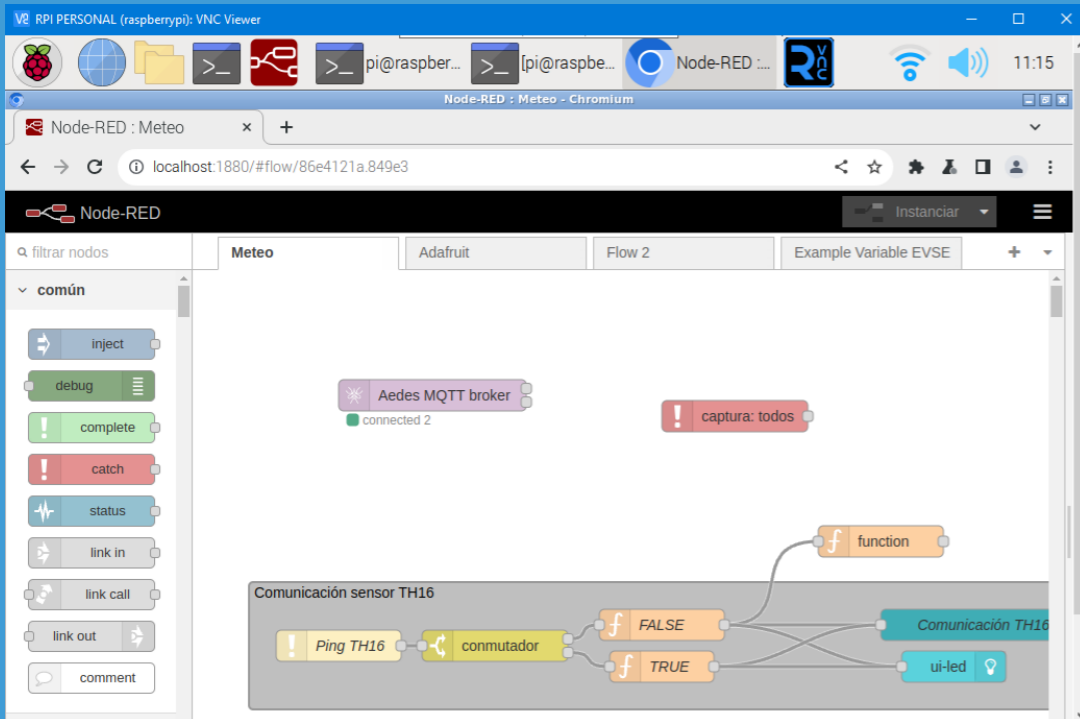


Raspberry

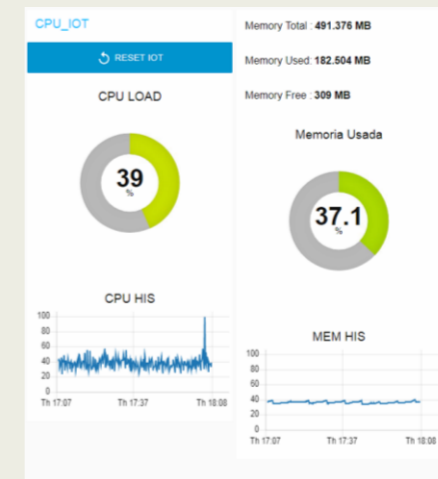
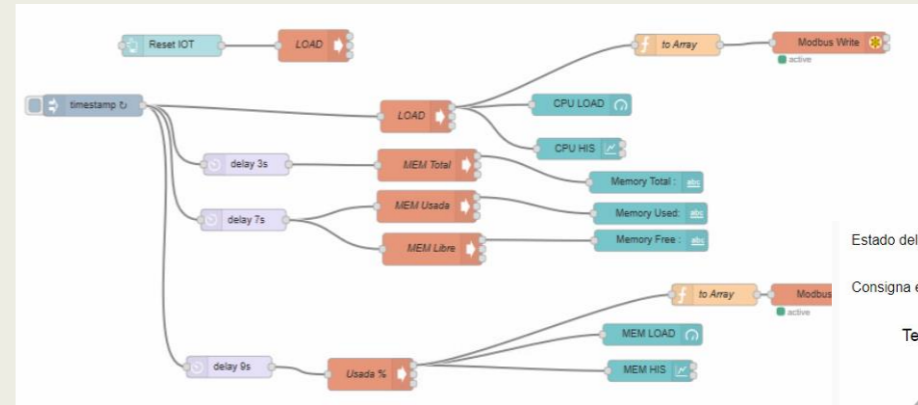


Node-red

We run Node-red on raspberry just as we did on PC or SIMATIC IOT 2020/2040 from Siemens.

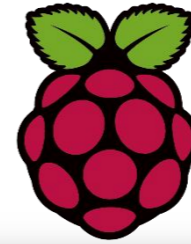


Flow examples (same as in Siemens PC or IOT).
Exportable in JSON format.



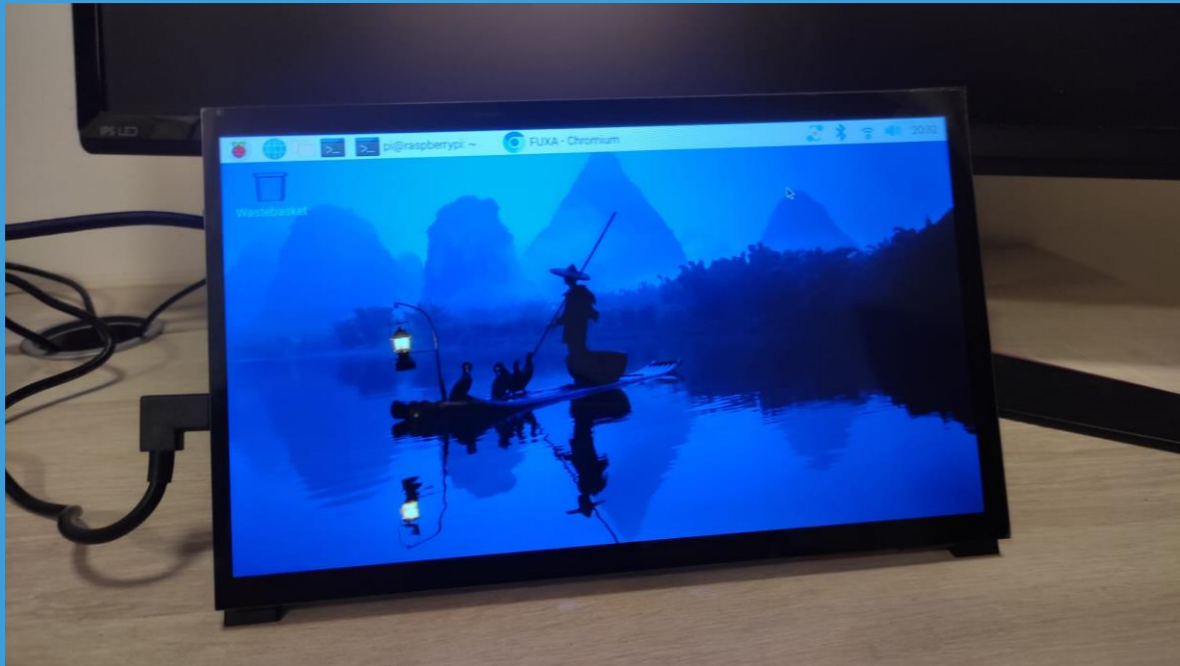


Raspberry



GeeekPi 10.1"

Capacitive Touchscreen for Raspberry Pi, HDMI Monitor IPS 1024X600 LCD Display with Dual-Speaker for Raspberry Pi 5/4B/3B+/3B, with Mount Hole to Install Raspberry Pi Board



GEEEKPI

- https://www.amazon.es/vdp/1cbbb4934fd44add9882ccf856934767?product=B0D9GGCB2Z&ref=cm_sw_em_r_ib_dt_tOcFrson53mic
- https://www.linkedin.com/posts/ricardo-moraleda-gareta-9421099_makers-geek-raspberrypi-activity-7269450771157430273--bxa?utm_source=share&utm_medium=member_desktop





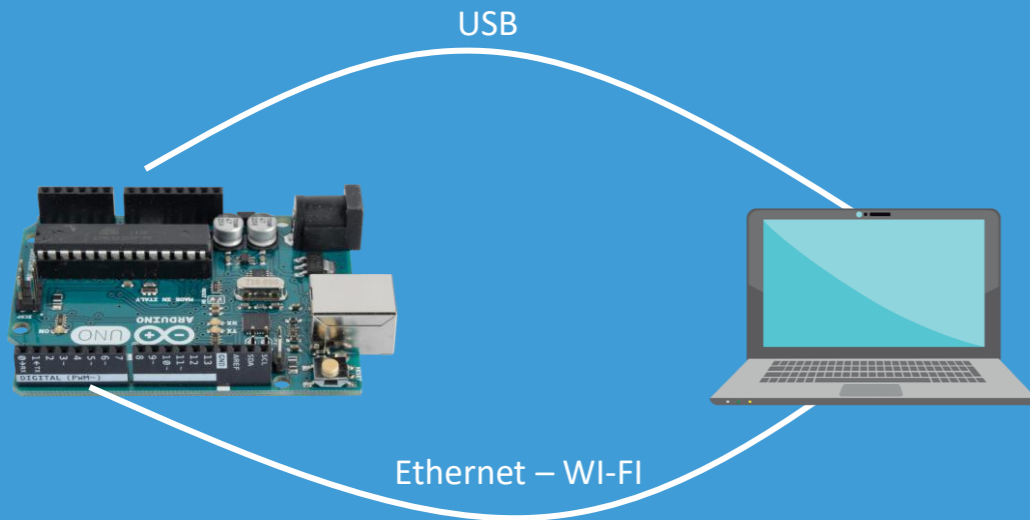
Arduino



Arduino 2009 / UNO

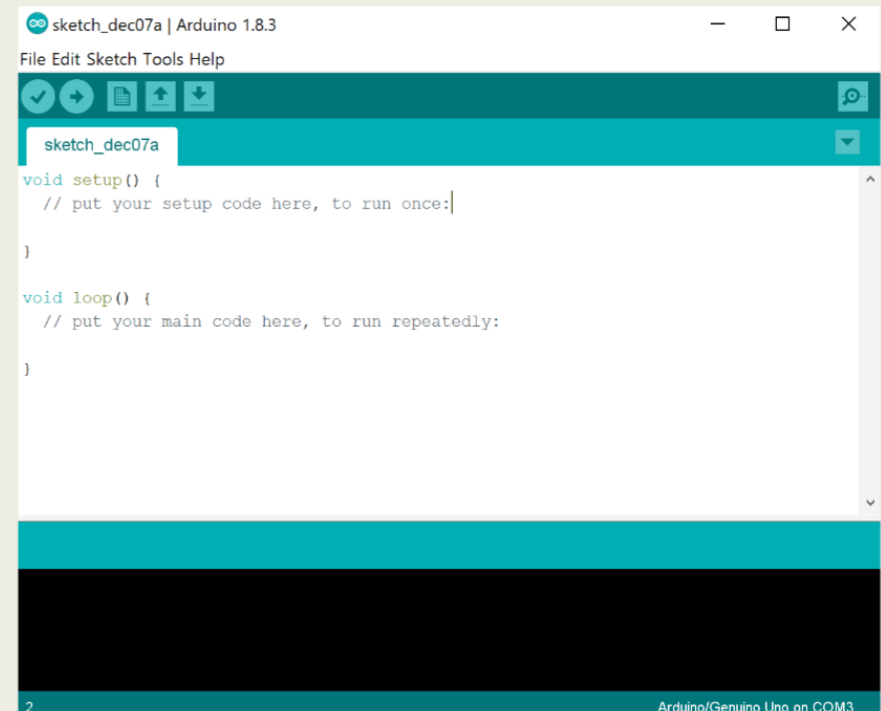
- Basic but powerful microcontroller. It has analog inputs and outputs.

<https://www.arduino.cc/en/software>



Allows shields (stackable backpacks) with different functionalities: Ethernet, Wi-Fi, GSM / 3G, I/O, etc.

IDE development environment for PC





IOT 2020/40

SIEMENS



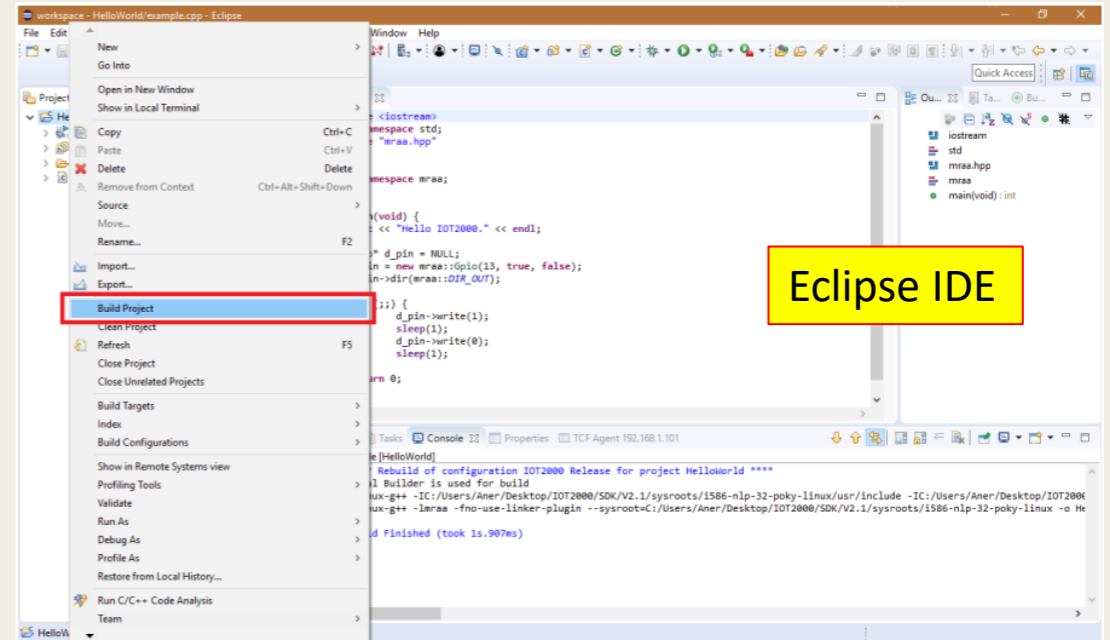
SIMATIC IOT 2020/40

- For **industrial** environments (IIOT)
 - DIN Rail mounting
- With Intel Galileo motherboard and compatible with Arduino shields.

	SIMATIC IOT 2000	ARDUINO	RASPERRY PI
Designed for Industrial field	Yes	No	No
Compatibility with Operating Systems	Yes	No	Yes
Compatibility with a multitude of protocols	Yes	No	Yes
Connectivity	Yes	Yes (requires Arduino Shield)	Yes
Information Capacity	Yes	No	Yes
Processing Capacity	Yes	No	Yes
RAM Memory	Yes	No	Yes
Analog input pins	Yes	Yes	No
Analog output pins	Yes	Yes	No

Yocto Linux distro operating system
 Programmable through various environments:

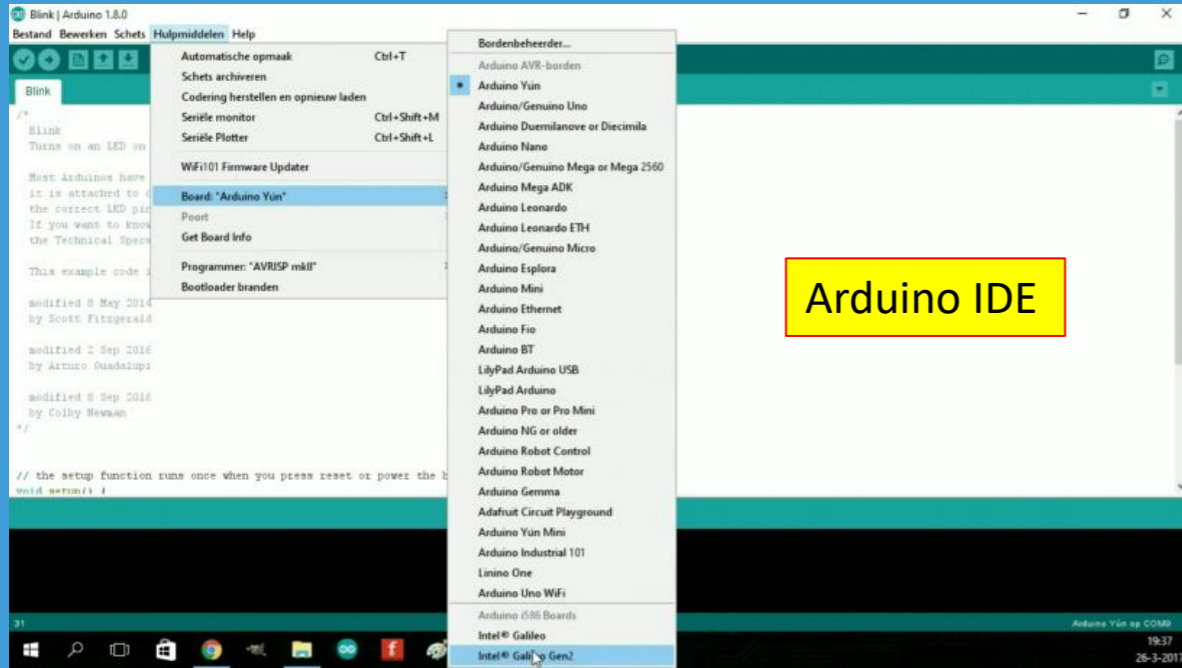
- Eclipse IDE (C o C++)
- Arduino IDE (C)
- Node-RED (JavaScript)





IOT 2020/40

SIEMENS



Arduino IDE

The board must be Intel Galileo or Intel Galileo 2nd generation. Sketches are programmed in C language

```

int led = 13;
// the setup routine runs once when you press reset:
void setup() {
  // initialize the digital pin as an output.
  pinMode(led, OUTPUT);
}

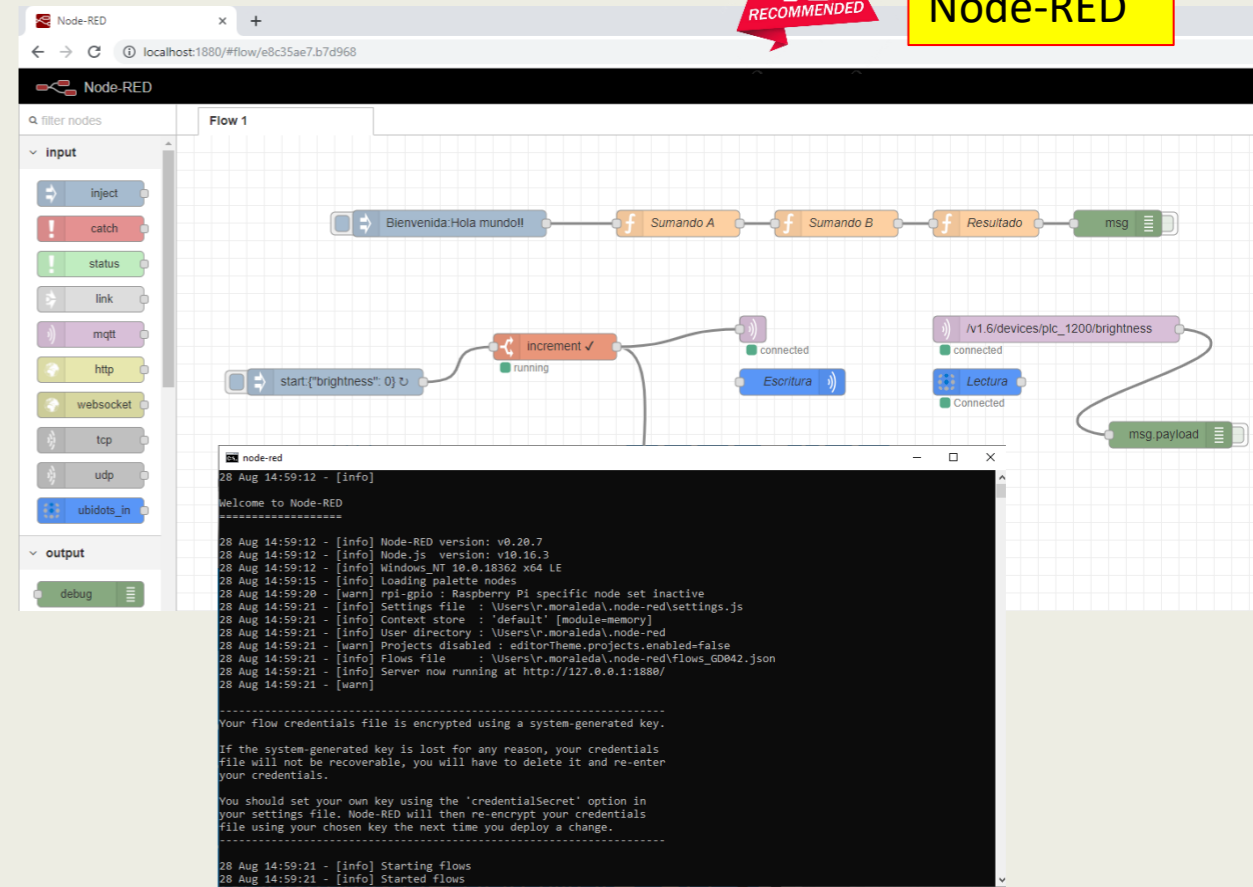
// the loop routine runs over and over again forever:
void loop() {
  digitalWrite(led, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000);             // wait for a second
  digitalWrite(led, LOW); // turn the LED off by making the voltage LOW
  delay(1000);             // wait for a second
}

```

variable declaration section

setup section

loop section



Node-RED

Web environment with Chrome or other browsers and very visual. Flows are programmed with nodes + JavaScript

EMBEDDED DEVICES

v.1.3 DECEMBER 2024



<https://www.linkedin.com/in/ricardo-moraleda-gareta-9421099>

<https://www.linkedin.com/company/gdo-electric1996/>

<https://www.linkedin.com/in/melvin-francis/>