

# SISTEME DE CALCUL ÎN TIMP REAL

Laboratorul IV – Sisteme de calcul pe bază de micro – computer utilizate în  
Ingineria Electrică

Drd. Ing. Pintilie Lucian – Nicolae

e-mail: [Lucian.Pintilie@emd.utcluj.ro](mailto:Lucian.Pintilie@emd.utcluj.ro)



# SISTEME CU MICRO – COMPUTER ȘI UTILIZAREA ÎN INGINERIA ELECTRICĂ

# SISTEME CU MICRO – COMPUTER ȘI UTILIZAREA ÎN INGINERIA ELECTRICĂ

- Procesele și fenomenele din aria disciplinară a Ingineriei Electrice, (pe lângă partea de comandă și control), necesită de asemenea o largă varietate de alte operații logice / digitale, în scopul deservirii aplicațiilor precum:
  - ✓ Preluarea informațiilor de la alte sisteme (secundare) de calcul;
  - ✓ Schimbului de date între echipamente (comunicații);
  - ✓ Centralizarea informațiilor preluate;
  - ✓ Sincronizarea și coordonarea sarcinilor de lucru la nivel global;

# SISTEME CU MICRO – COMPUTER ȘI UTILIZAREA ÎN INGINERIA ELECTRICĂ

- În scopul deservirii altor operații digitale decât cele destinate comenzii și controlului, se utilizează MICRO-CALCULATOARELE sau MICRO-COMPUTERELE. Aceste sisteme de calcul prezintă:
  - ✓ Micro-procesor cu set redus de instrucțiuni (ex. ARM – RISC);
  - ✓ Sistem de operare (ex. Embedded Linux / Windows CE);
  - ✓ Interfețe de comunicare (ex. Ethernet, Wi-fi, Bluetooth, SPI, I2C);
  - ✓ Intrări și ieșiri digitale de uz general (eng. GPIO / Interrupt I/O);

# SISTEME CU MICRO – COMPUTER ȘI UTILIZAREA ÎN INGINERIA ELECTRICĂ

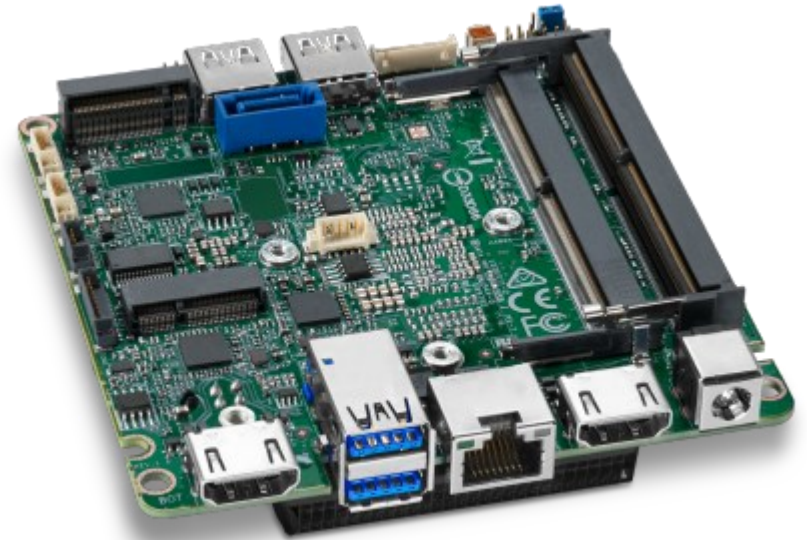
- Câteva exemple de astfel de sisteme de calcul utilizate în mod frecvent în Ingineria Electrică ar fi:
  - ✓ Intel Galileo / Edison / Atom / NUC;
  - ✓ Raspberry Pi;
  - ✓ Orange Pi;





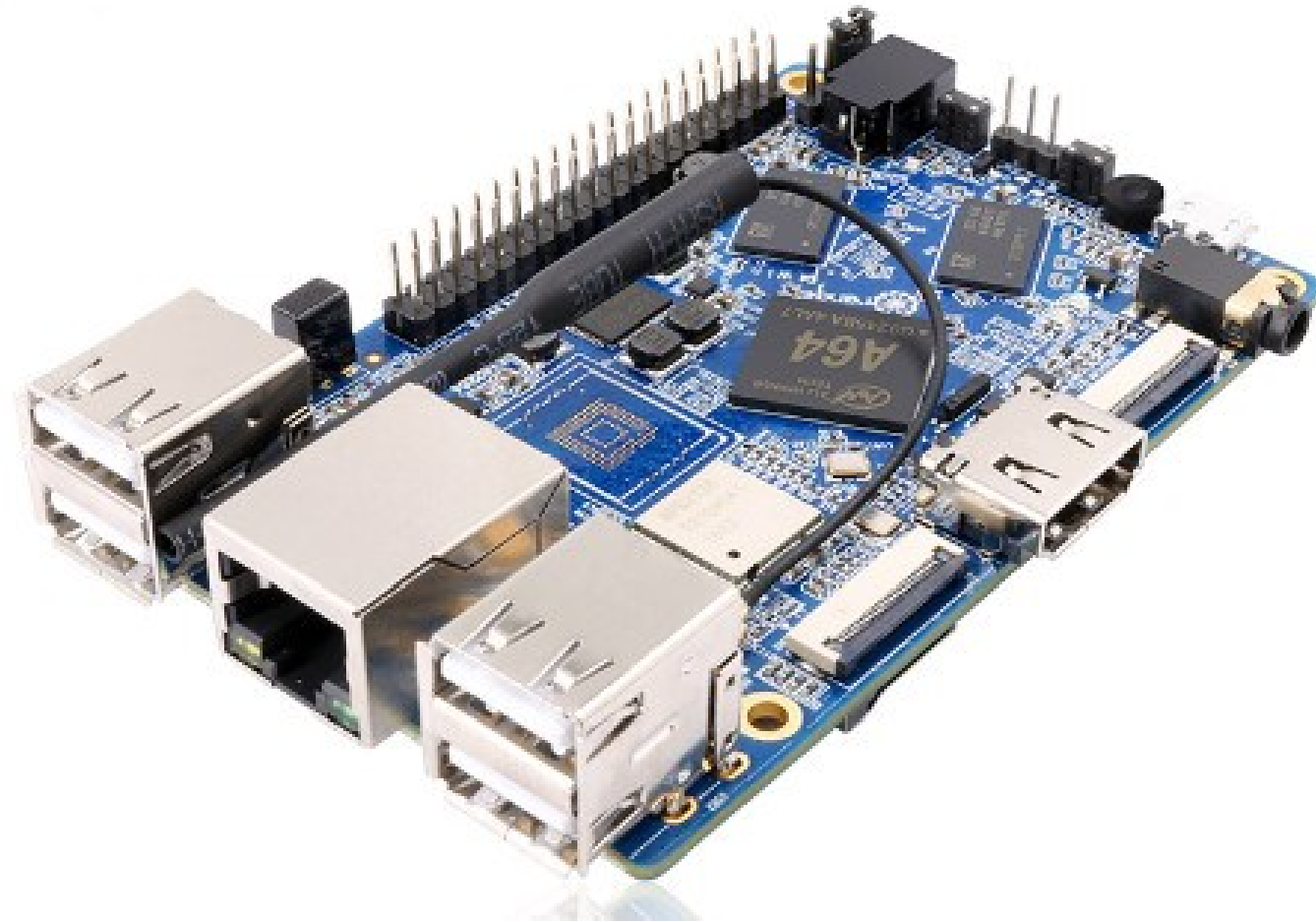
## Quad-Core Intel Atom x5

<https://hothardware.com/news/up-core-dev-board-boasts-quad-core-intel-atom-x5-grunt-to-challenge-raspberry-pi>



## Intel NUC

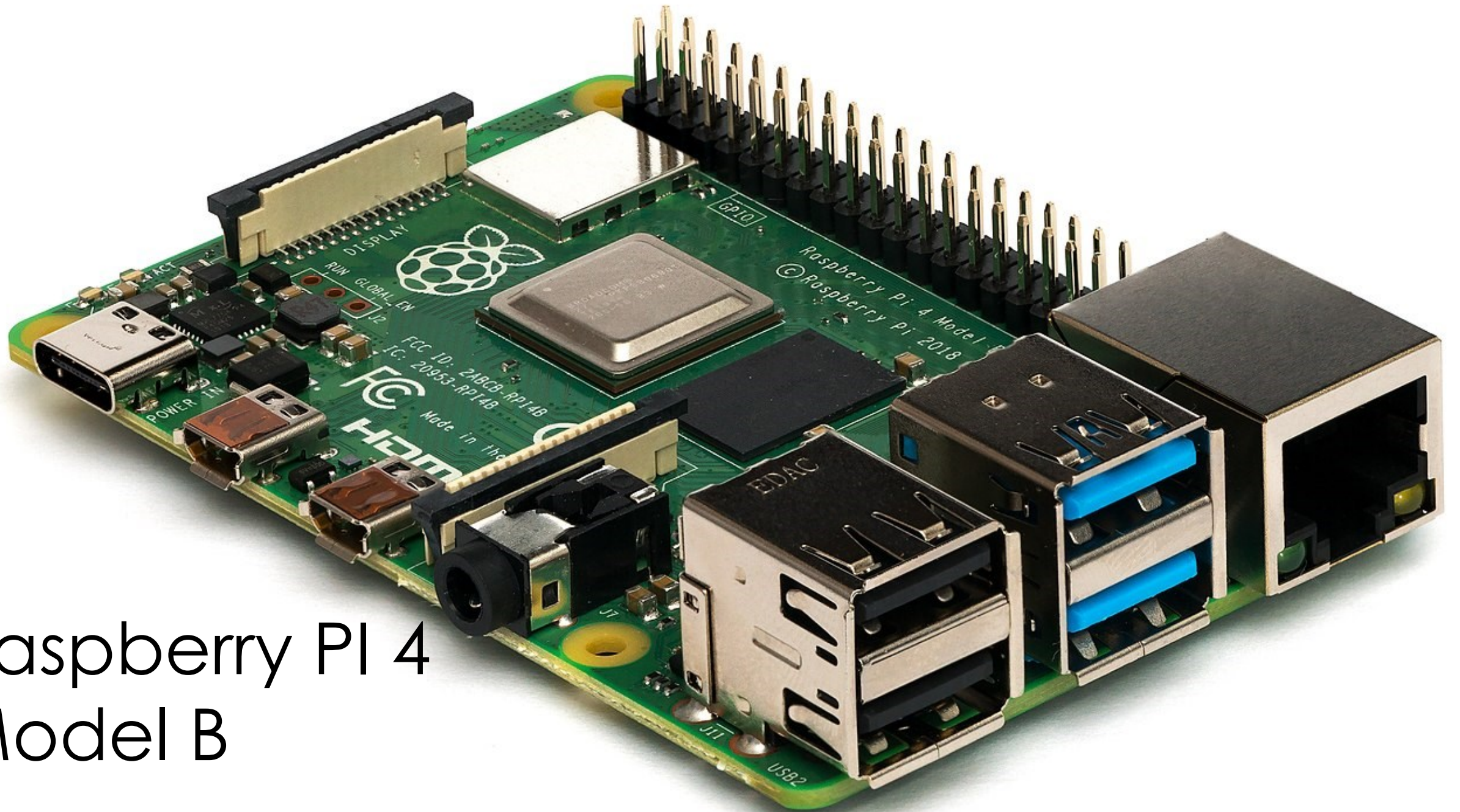
<https://www.intel.com/content/www/us/en/products/boards-kits/nuc/boards/nuc7i3dnbe.html>



# Orange Pi

<http://linuxgizmos.com/latest-orange-pi-offers-quad-a53-cores-and-2gb-of-ram/>





# Raspberry Pi 4 Model B

[https://en.wikipedia.org/wiki/Raspberry\\_Pi#/media/File:Raspberry\\_Pi\\_4\\_Model\\_B\\_-\\_Side.jpg](https://en.wikipedia.org/wiki/Raspberry_Pi#/media/File:Raspberry_Pi_4_Model_B_-_Side.jpg)



# METODE DE UTILIZARE ȘI LIMBAJE DE PROGRAMARE UTILIZATE

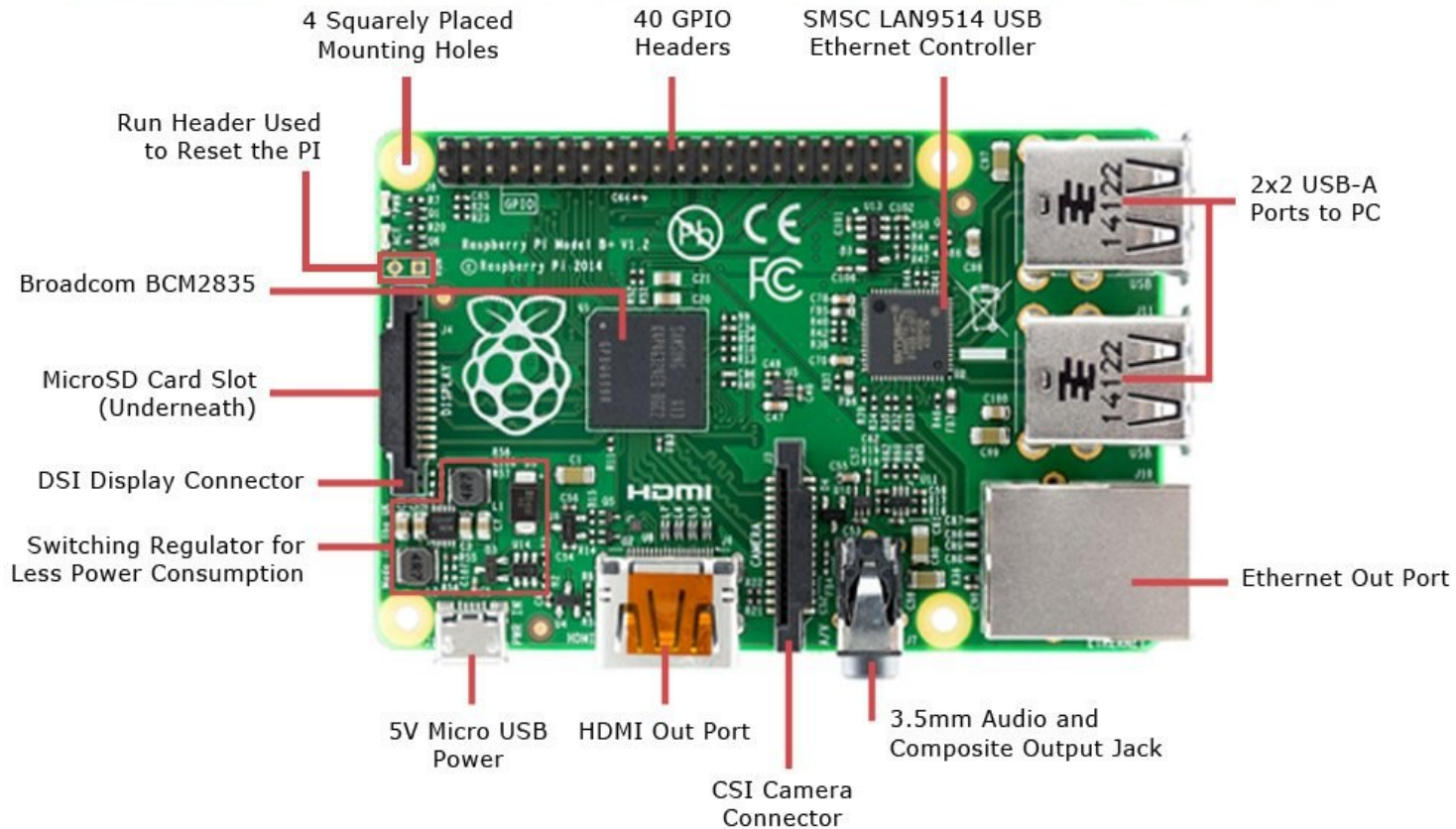
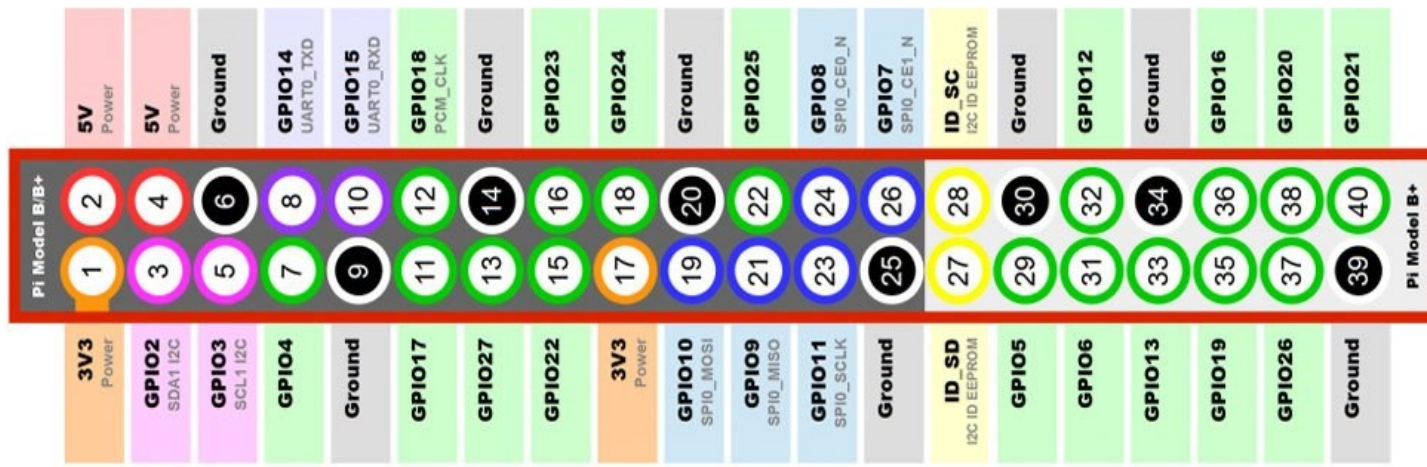
# METODE DE UTILIZARE ȘI LIMBAJE DE PROGRAMARE UTILIZATE

- În majoritatea cazurilor, micro-computerul deserveste următoarele roluri funcționale în echipamentele dedicate:
  - ✓ Server de aplicație dedicat (ex. instrumentație SCADA);
  - ✓ Server de comunicație (ex. rutarea diverselor protocoale);
  - ✓ Centralizator de informații (ex. server de baze de date SQL);
  - ✓ Server WEB pentru interfațarea proceselor de la distanță;
  - ✓ Interfață hardware de interacțiune (touchscreen + interfață);

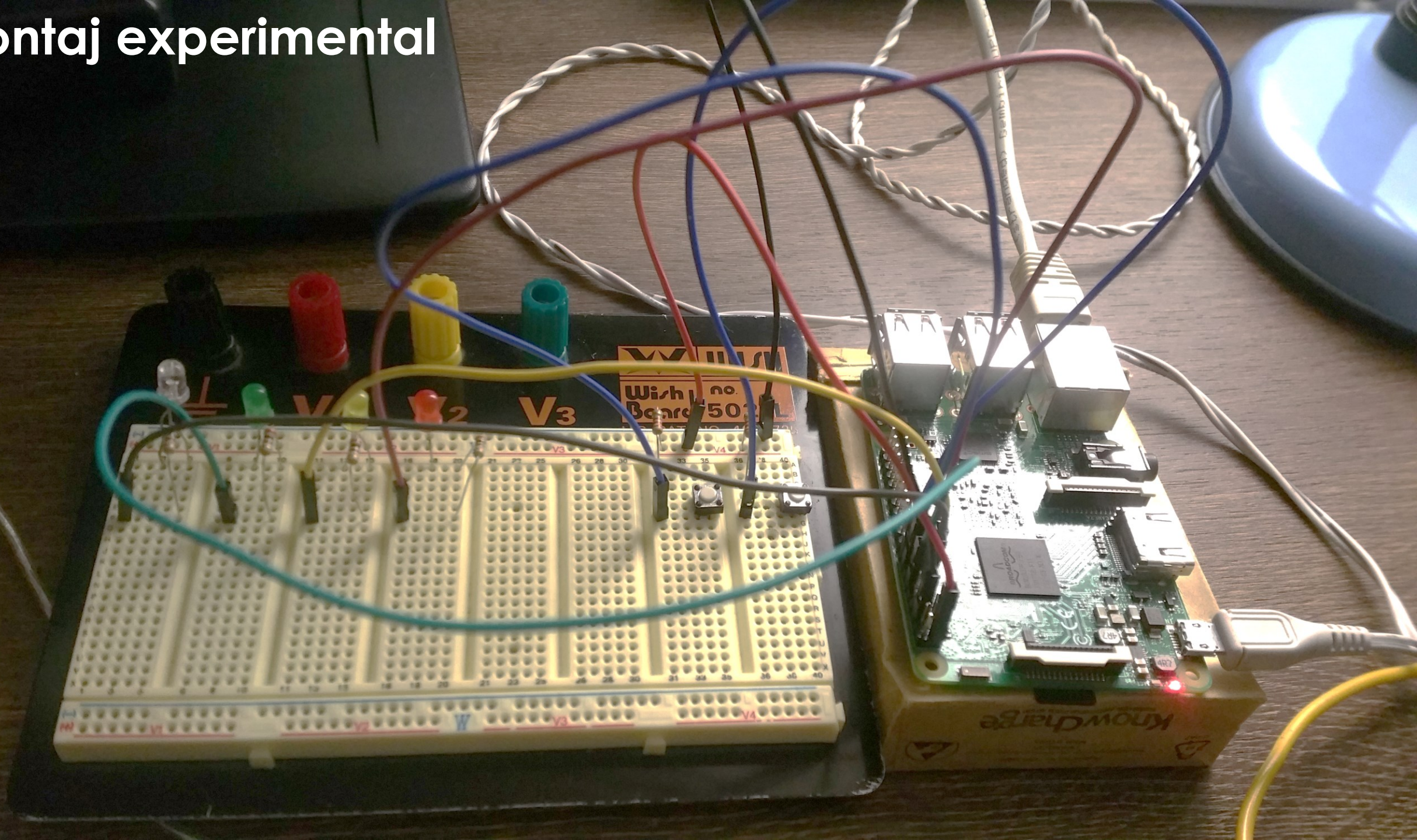
# METODE DE UTILIZARE ȘI LIMBAJE DE PROGRAMARE UTILIZATE

- Aproape în toate aplicațiile menționate anterior, NU ESTE ABSOLUT NECESARĂ UTILIZAREA INTERFEȚEI GRAFICE. Astfel, în marea majoritate a aplicațiilor se preferă sistemele de operare de tip UNIX / LINUX, care pot funcționa independent de interfața grafică, în schimb sistemul de calcul poate fi accesat de la distanță printr-o consolă de comandă (ex. SSH) sau printr-o interfață WEB (ex. Node-RED). Există deci următoarele metode de programare și gestionare:
  - ✓ Programarea în mod direct prin limbaj Python;
  - ✓ Programarea în mod indirect din mediul Matlab – Simulink;
  - ✓ Programarea în mod indirect cu ajutorul platformei WEB Node-Red;

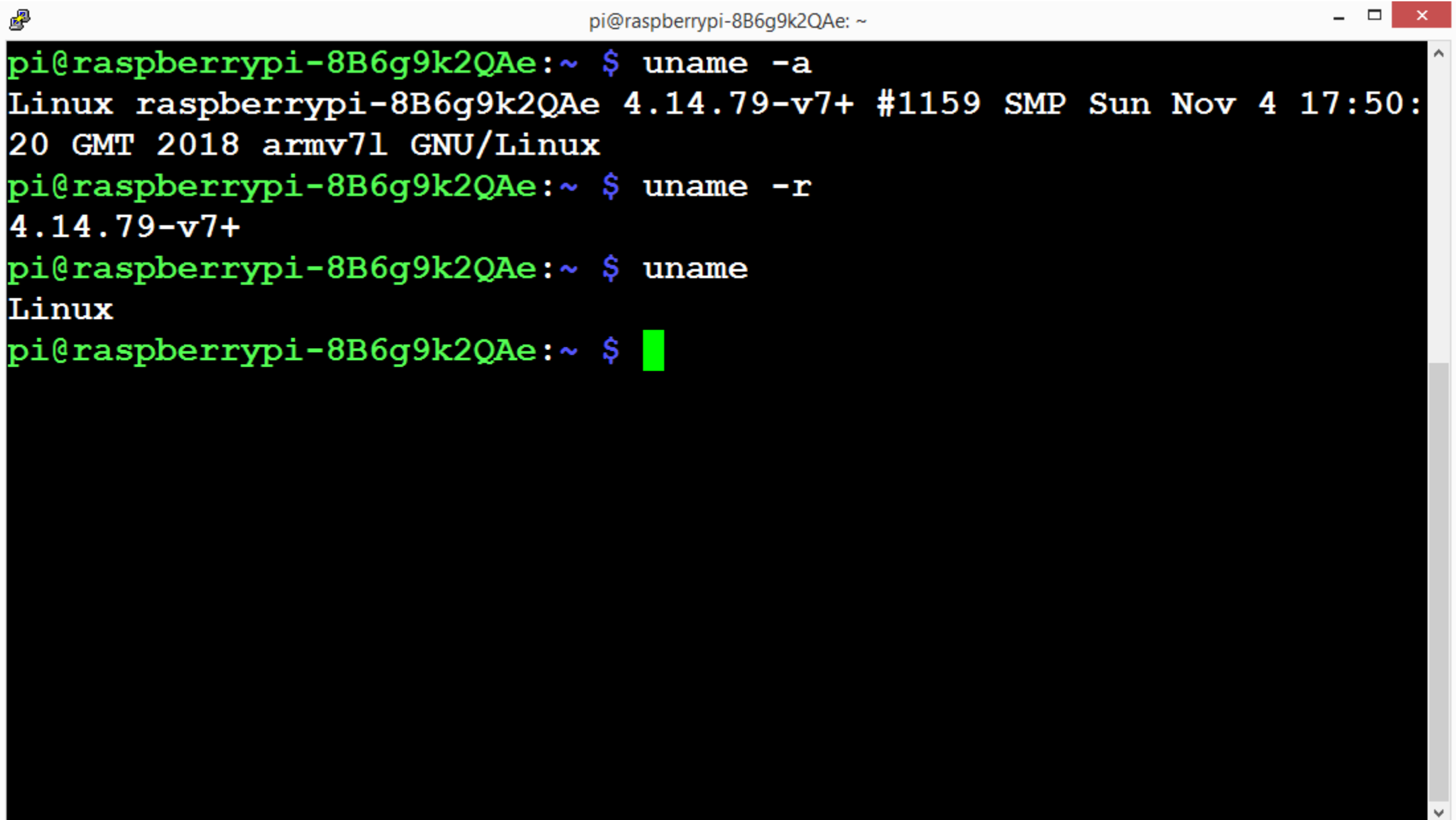
## GPIO Pinout Diagram



# Montaj experimental



# Protocol SSH – accesarea consolei de comandă la distanță

A screenshot of a terminal window titled "pi@raspberrypi-8B6g9k2QAe: ~". The terminal shows the execution of three 'uname' commands. The first command is 'uname -a', which outputs: "Linux raspberrypi-8B6g9k2QAe 4.14.79-v7+ #1159 SMP Sun Nov 4 17:50:20 GMT 2018 armv7l GNU/Linux". The second command is 'uname -r', which outputs: "4.14.79-v7+". The third command is 'uname', which outputs: "Linux". The prompt is green, and the cursor is a green block.

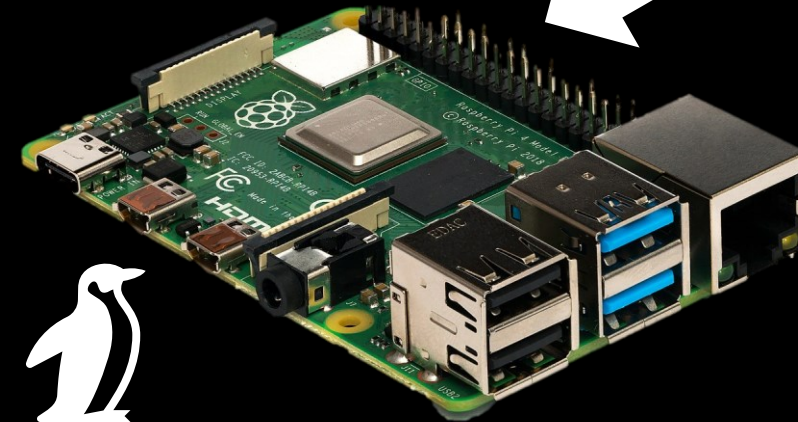
```
pi@raspberrypi-8B6g9k2QAe:~ $ uname -a
Linux raspberrypi-8B6g9k2QAe 4.14.79-v7+ #1159 SMP Sun Nov 4 17:50:
20 GMT 2018 armv7l GNU/Linux
pi@raspberrypi-8B6g9k2QAe:~ $ uname -r
4.14.79-v7+
pi@raspberrypi-8B6g9k2QAe:~ $ uname
Linux
pi@raspberrypi-8B6g9k2QAe:~ $ █
```

# PYTHON

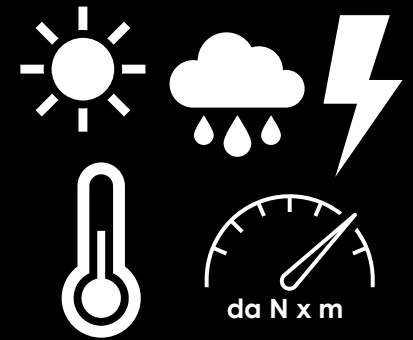
Calculatorul  
gazdă



TCP / IP - SSH



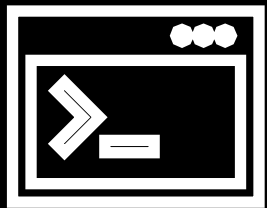
I2C, SPI,  
UART, GPIO



Senzori  
digitali

RaspBerry Pi

cu sistem de operare - Linux

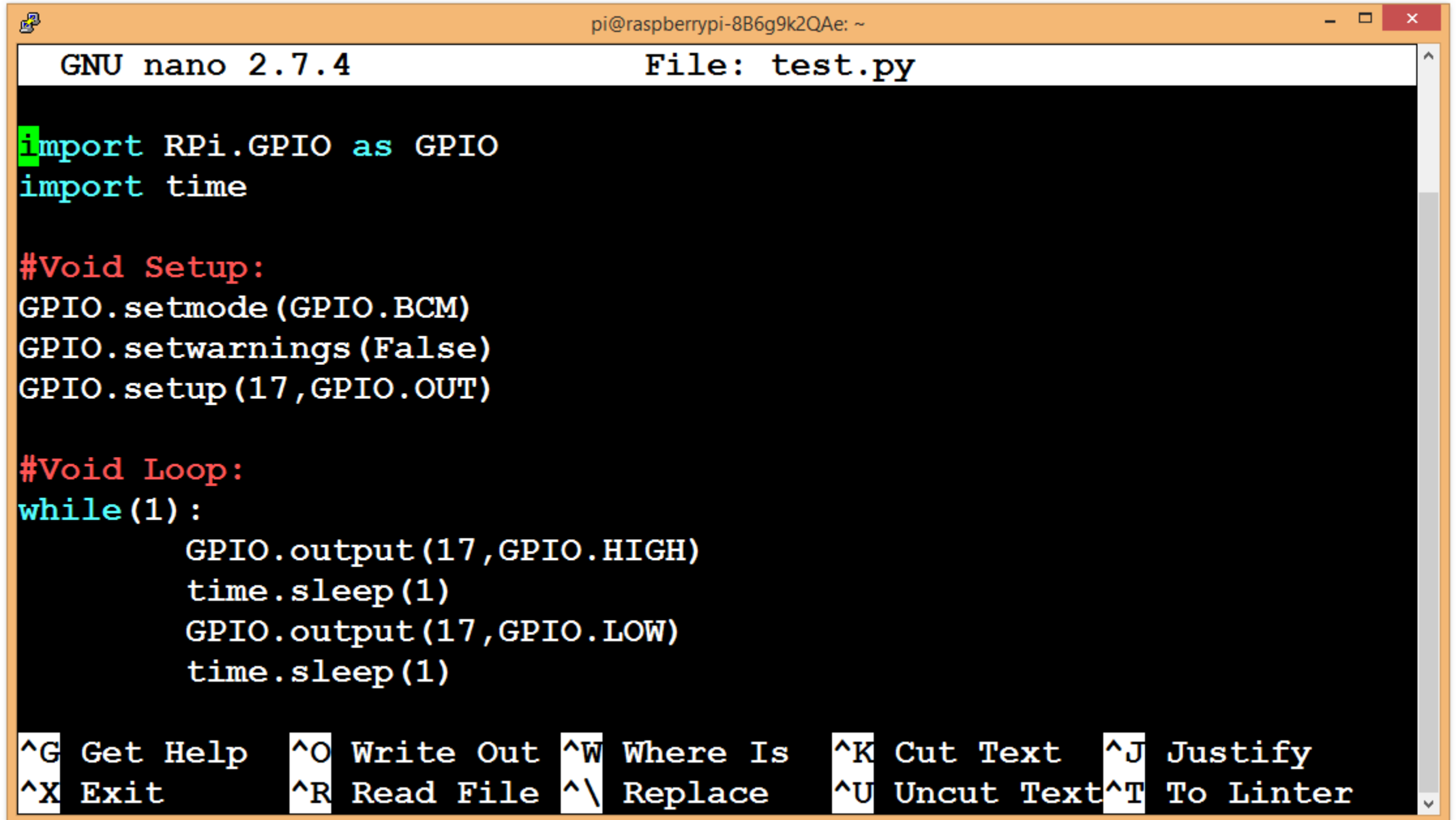


Consola de comandă

accesată de la distanță prin protocol SSH



# Redactarea unui program în limbaj Python



The image shows a terminal window with a nano editor. The window title is "pi@raspberrypi-8B6g9k2QAe: ~". The editor title is "GNU nano 2.7.4" and the file name is "File: test.py". The code in the editor is as follows:

```
import RPi.GPIO as GPIO
import time

#Void Setup:
GPIO.setmode(GPIO.BCM)
GPIO.setwarnings(False)
GPIO.setup(17,GPIO.OUT)

#Void Loop:
while(1):
    GPIO.output(17,GPIO.HIGH)
    time.sleep(1)
    GPIO.output(17,GPIO.LOW)
    time.sleep(1)
```

At the bottom of the editor, there is a help menu with the following options:

<b>^G</b> Get Help	<b>^O</b> Write Out	<b>^W</b> Where Is	<b>^K</b> Cut Text	<b>^J</b> Justify
<b>^X</b> Exit	<b>^R</b> Read File	<b>^\</b> Replace	<b>^U</b> Uncut Text	<b>^T</b> To Linter

# MATLAB - SIMULINK

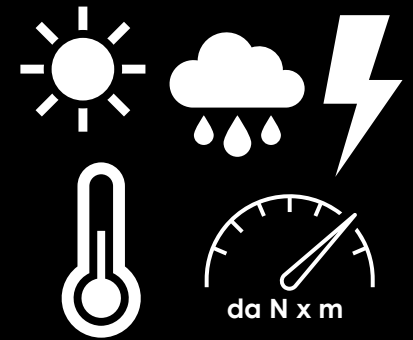
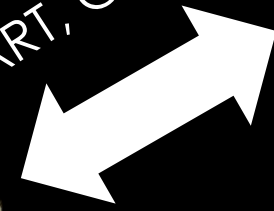
Calculatorul gazdă



TCP / IP

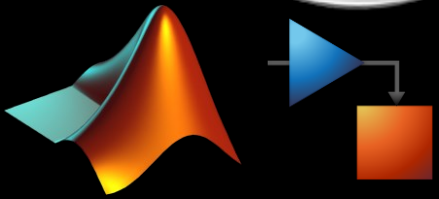


I2C, SPI,  
UART, GPIO



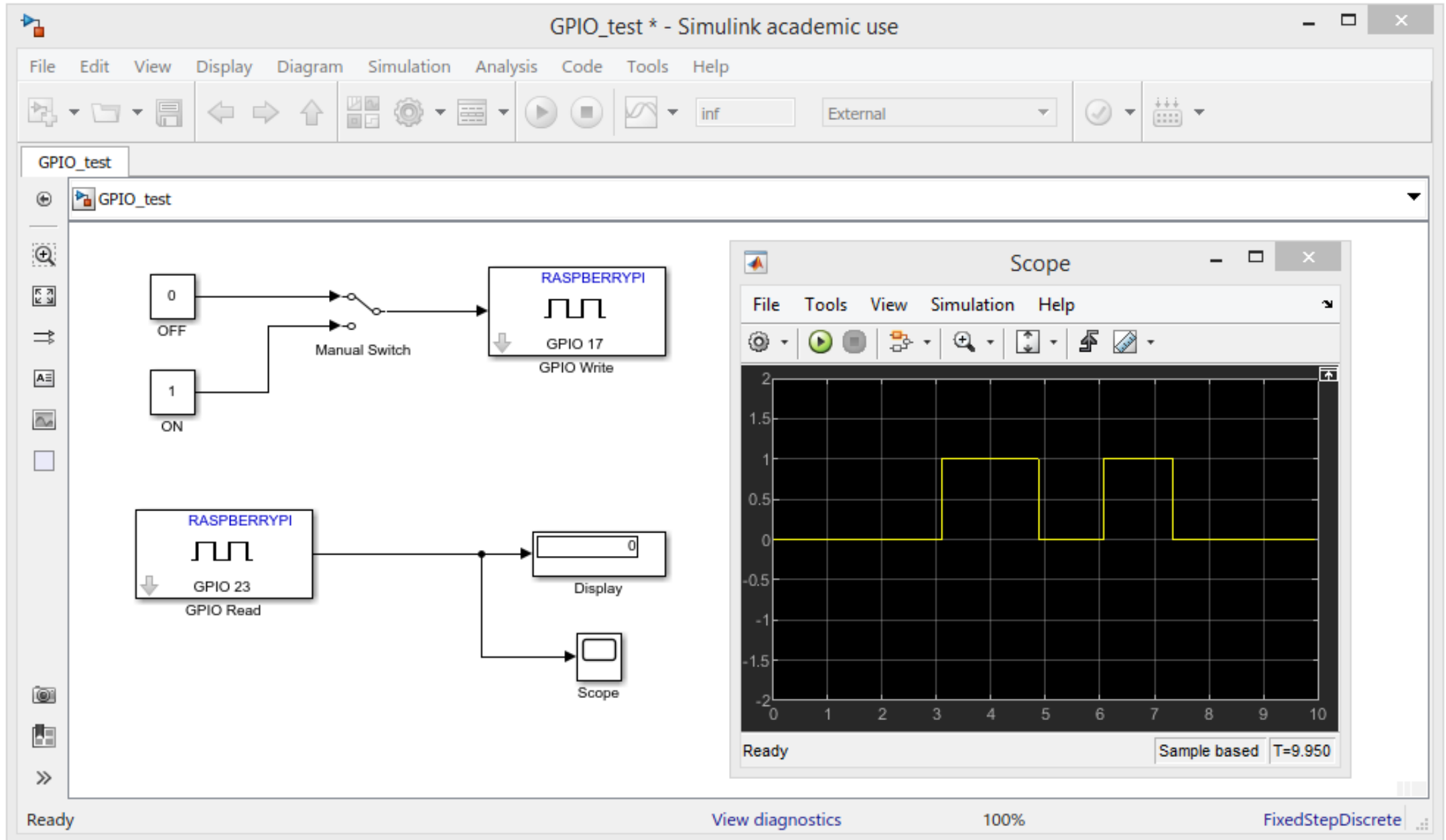
Senzori digitali

Raspberry Pi  
cu sistem de operare - Linux



Mediul de simulare  
Matlab - Simulink

# Implementarea aplicațiilor sub formă de model Simulink

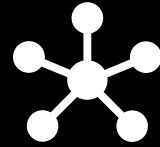


# NODE - RED

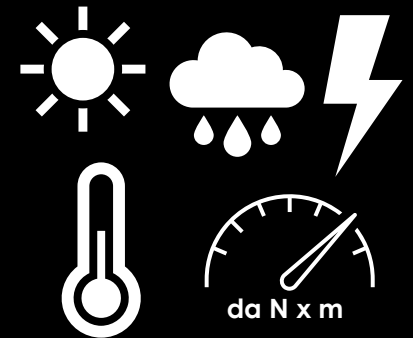
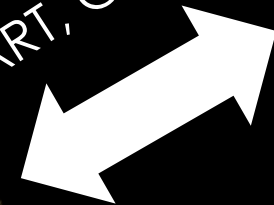
Calculatorul gazdă



TCP / IP - HTTP



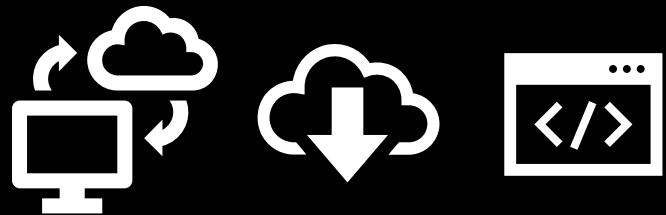
I2C, SPI,  
UART, GPIO



Senzori digitali

RaspBerry PI

cu sistem de operare – Linux și Node - Red



Program de navigare WEB

# Implementarea aplicațiilor în platforma WEB - NodeRed

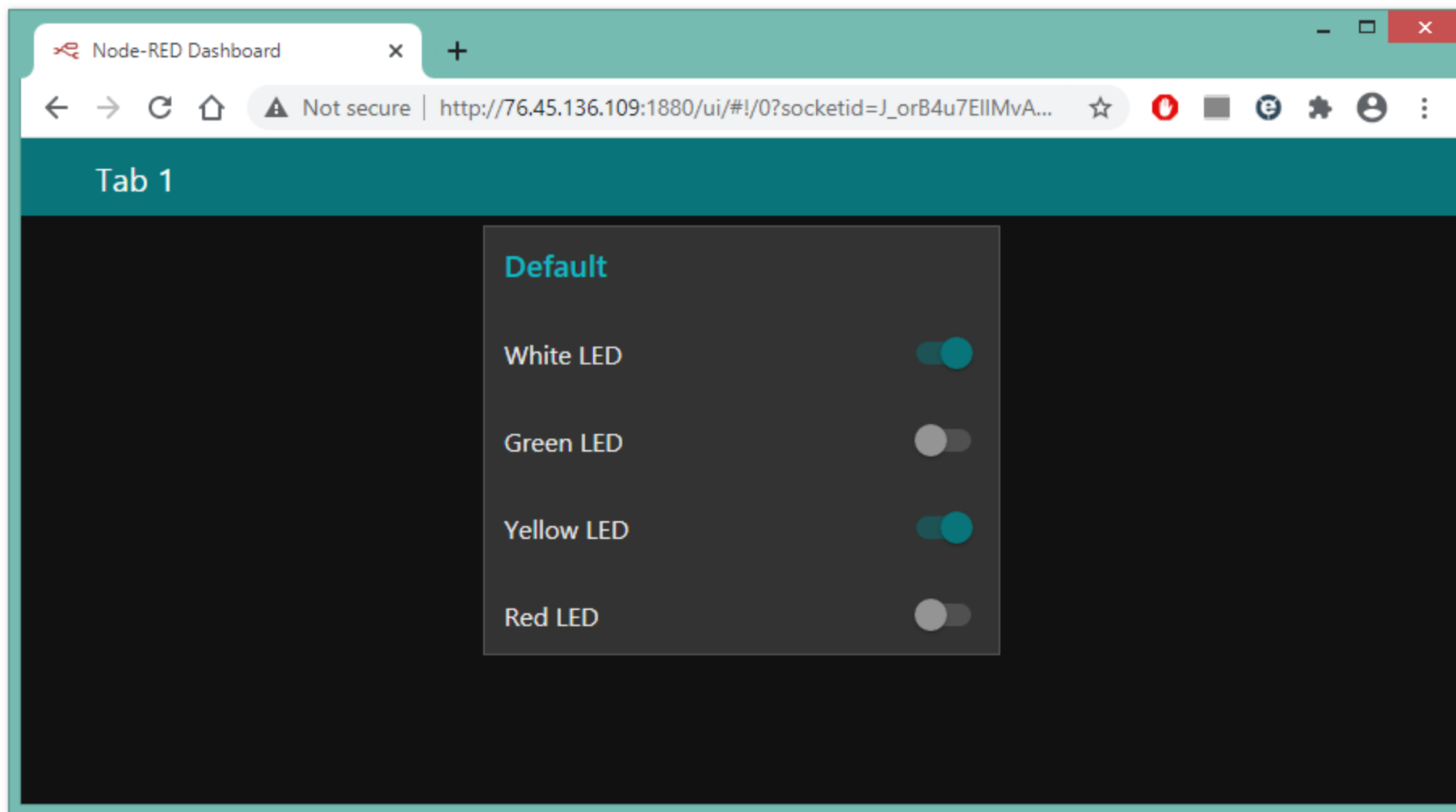
The image shows a screenshot of the Node-RED web interface in a browser. The browser's address bar shows the URL `http://76.45.136.109:1880/#flow/c4d1aff9.314c8`. The Node-RED interface includes a top navigation bar with the Node-RED logo and a "Deploy" button. Below this is a search bar for nodes and a panel for "Flow 1".

The "Flow 1" workspace contains four parallel flow elements, each consisting of an LED control node and a GPIO pin node:

- White LED:** The LED node has a green "on" indicator. It is connected to the "PIN: 11" node, which has a green "true" indicator.
- Green LED:** The LED node has a green "on" indicator. It is connected to the "PIN: 12" node, which has a green "true" indicator.
- Yellow LED:** The LED node has a green "on" indicator. It is connected to the "PIN: 13" node, which has a green "true" indicator.
- Red LED:** The LED node has a green "on" indicator. It is connected to the "PIN: 15" node, which has a green "true" indicator. This node is highlighted with an orange border.

The left sidebar shows a "common" category of nodes including "inject", "debug", "complete", "catch", "status", "link in", "link out", and "comment". A "function" category is also visible at the bottom of the sidebar.

# Implementarea aplicațiilor în platforma WEB - NodeRed





DEMONSTRAȚIE

# TEMĂ

- Pe baza materialelor prezentate, precizați:
  1. Diferențele dintre un micro – controller și un micro - computer;
  2. Rolul funcțional al micro - computerului într-un echipament;
  3. Un exemplu de aplicație în care ați utiliza micro - computerul;