Sisteme de calcul în timp real

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

I. INTRODUCERE:

În cadrul ariei disciplinare a Ingineriei Electrice, există diverse alte întrebuințări ale sistemelor de calcul în afara aplicațiilor de control sau reglare automată. Astfel de aplicații pot fi clasificate ca fiind aplicații uzuale (obișnuite, zilnice) sau speciale, în funcție de complexitatea procesului deservit. Câteva exemple de astfel de aplicații ar fi:

- Preluarea informațiilor de la sisteme secundare de calcul;
- Schimbul de date în timp real dintre echipamente;
- Centralizarea datelor achiziționate de la senzori;
- Sincronizarea și prioritizarea sarcinilor de lucru la nivel global al sistemului privit ca și rețea;

Pentru a deservi aplicațiile amintite, este necesar un sistem de calcul, care poate să realizeze sarcinile de lucru într-un așa zis paralelism aparent (ex. multi-tasking). Această funcție permite executarea mai multor aplicații în paralel. De asemenea, este necesară posibilitatea controlului execuției aplicației (ex. pornirea sau oprirea / suspendarea execuției aplicației). Totodată, este necesar, ca sistemul de calcul să poată gestiona fișierele stocate în memoria internă sau externă, prin așa-zisul „sistem de gestionare al fișierelor” (eng. File System Management, ex. NTFS / FAT – Windows, EXT / HFS – UNIX – Linux – Darwin). În aceeaș măsură, unitatea centrală de procesare a sistemului de calcul, trebuie să fie un micro-procesor cu set redus de instrucțiuni (ex. A.R.M. – Advanced R.I.S.C. Machine / R.I.S.C. – Reduced Instructions Set Computer). Un astfel de sistem de calcul, poartă denumirea de „micro – computer încorporat / înglobat” (eng. Embedded Micro – Computer).

Pentru a satisface toate funcțiile amintite, micro-computerul rulează un așa – zis „sistem de operare”. Spre deosebire de computerul personal (ex. stație de lucru sau server), „micro-sistemele de calcul” sunt sisteme de calcul dedicate aplicației pe care o deservesc. De asemenea, sistemul de operare pe care îl rulează este astfel conceput încât, să consume cât mai puține resurse, să pornească într-un interval de timp cât mai scurt și să lanseze cât mai puține aplicații sau sevicii în fundal (necesare rulării sistemului de operare). Practic, sistemul de operare este optimizat la maxim pentru a deservi aplicația pentru care a fost conceput. Câteva exemple de astfel de sisteme de operare ar fi:

- Windows CE (eng. Compact Embedded) [1];
- Windows Embedded (Industry PRO) [2];
- Windows 10 core IoT [3];
- Embedded Linux (ex. Raspbian, Yocto / Clanton, Ubuntu for ARM, Suse for ARM etc...) [4];

Toate aceste sisteme de operare pe lângă faptul că dispun de o interfață de interacțiune cu factorul uman (ex. consolă de comandă sau interfață grafică), permit conectarea de la distanță la consola lor de comandă sau la interfața de lucru (eng. Remote Desktop). Pentru accesul la consola de comandă se utilizează protocolul de comunicație SSH (eng. Secure Shell). De asemenea, sistemele de operare compacte, permit funcționarea sistemului de calcul în regim de server, anume, pot rula servicii și aplicații în fundal. Câteva exemple de aplicații server ar fi: FTP (eng. File Transfer Protocol), WEB / HTTP / PHP etc...

Realizat de: Asist. drd. Pintilie Lucian - Nicolae
Pentru disciplina: „Sisteme de calcul în timp real”
Adresă de e-mail: Lucian.Pintilie@emd.utcluj.ro

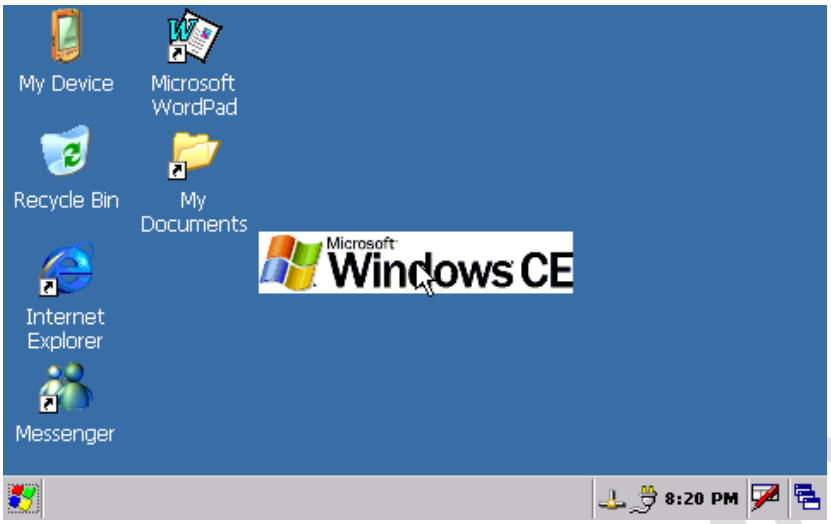

Fig. 1 – Windows CE - Compact Embedded [1]

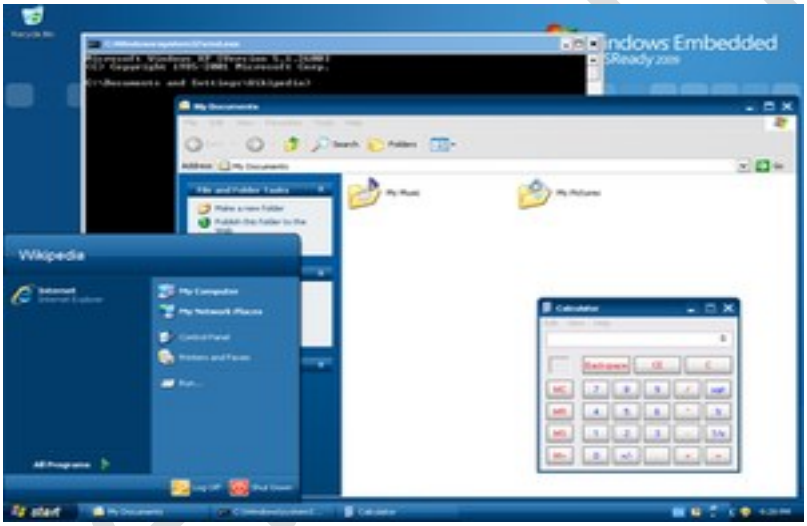

Fig. 2 – Windows Embedded – Industry PRO [2]

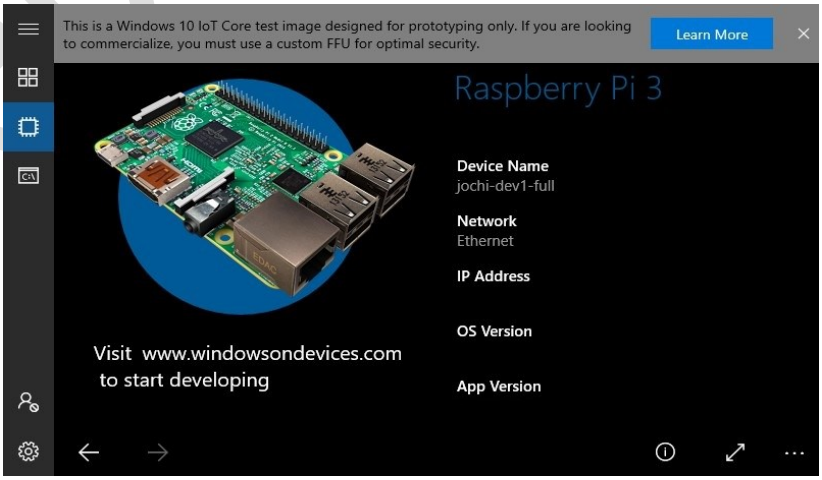

Fig. 3 – Windows 10 core IoT [3]

Realizat de: Asist. drd. Pintilie Lucian - Nicolae
Pentru disciplina: „Sisteme de calcul în timp real”
Adresă de e-mail: Lucian.Pintilie@emd.utcluj.ro

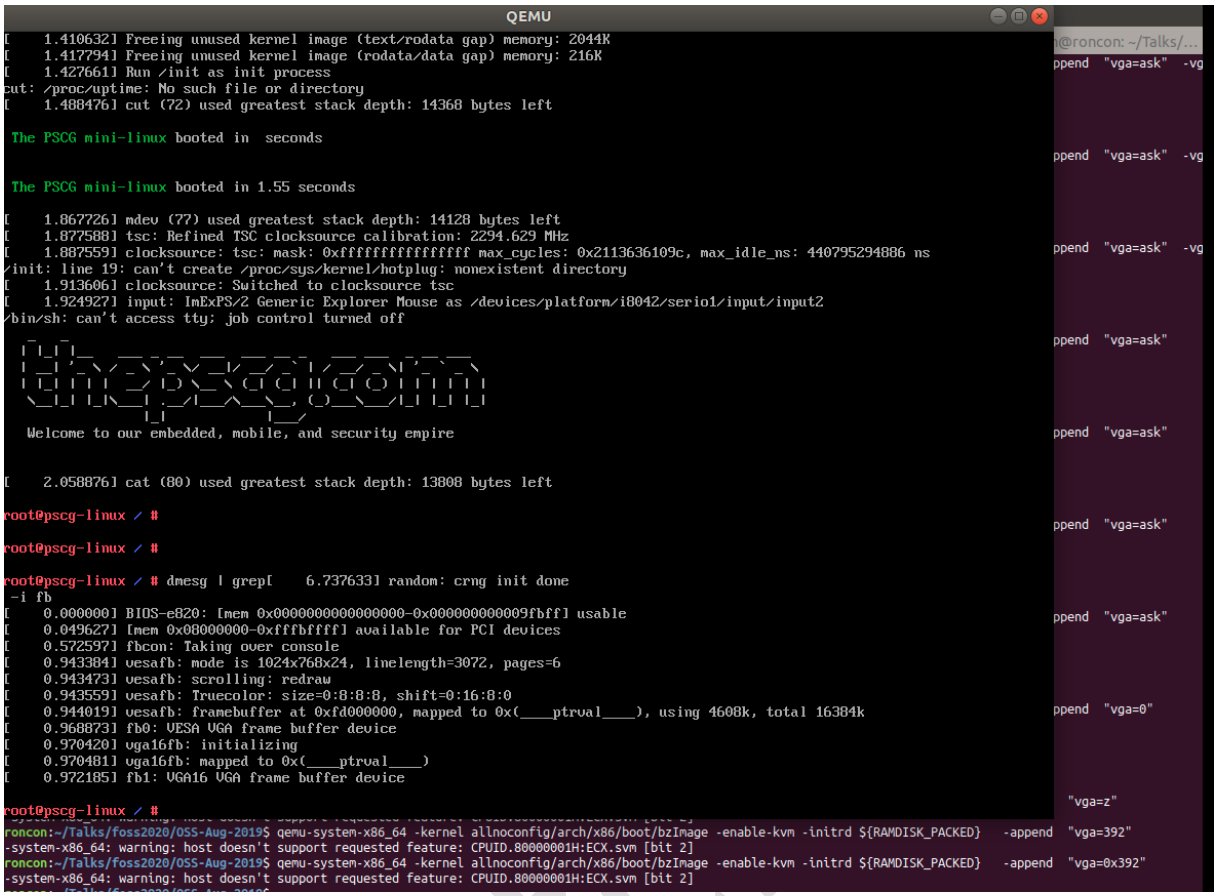

Fig. 4 – Embedded Linux [4]

Micro – sistemele de calcul care rulează astfel de sisteme de operare, pot fi considerate ca și sisteme de calcul dedicate aplicației. Pe lângă unitatea centrală de calcul specializată, un micro – sistem de calcul (micro – computer) dispune și de periferice specializate precum:
- Intrări și ieșiri digitale de uz general (eng. GPIO – General Purpose Input / Output);
- Interfețe de comunicare inter-echipament (ex. I^2C, SPI, UART, USART, Parallel, One Wire etc.);
- Interfețe universale de comunicare (ex. OTG-USB, Ethernet, S-ATA, mini-PCI-eXpress etc.);
- Interfețe de comunicație radio (ex. Bluetooth, WiFi, ZigBee, RF etc.);

Arhitectura constructivă a unui astfel de sistem poartă denumirea „System-on-a-Chip”, deoarece, toate componentele esențiale (ex. memorie RAM, unitatea centrală de procesare, adaptorul video etc.) ale sistemului de calcul sunt înglobate într-o singură capsulă de circuit integrat. Acest tip constructiv presupune reducerea semnificativă a gabaritului sistemului de calcul, rezultând o structură foarte compactă, având un consum redus de energie. Singurul impediment, ar fi, impedimentul termic al unui astfel de sistem, deoarece dacă cel puțin una dintre componente degajă căldură la nivelul capsulei, toate celelalte structuri din acea capsulă vor fi afectate. De asemenea, în cazul defectării unui etaj din interiorul capsulei, întrega structură va fi compromisă în mod ireversibil.

Câteva exemple de nume de micro-computere des întâlnite în practică ar fi:
- Intel Galileo / Edison / Atom / NUC [5] [6] [7] [8];
- Raspberry PI [9];
- Orange PI [10];

Realizat de: Asist. drd. Pintilie Lucian - Nicolae
Pentru disciplina: „Sisteme de calcul în timp real”
Adresă de e-mail: Lucian.Pintilie@emd.utcluj.ro

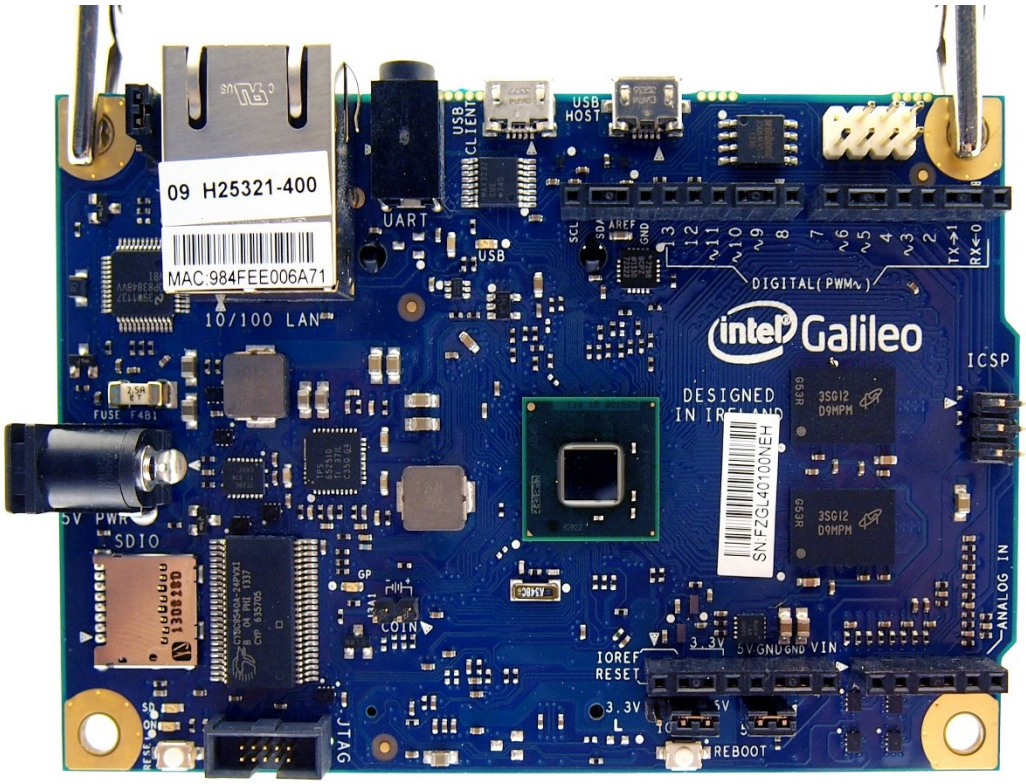

Fig. 5 – Intel Galileo – Gen II [5]

Fig. 6 – Intel Edison [6]

Realizat de: Asist. drd. Pintilie Lucian - Nicolae
Pentru disciplina: „Sisteme de calcul în timp real”
Adresă de e-mail: Lucian.Pintilie@emd.utcluj.ro

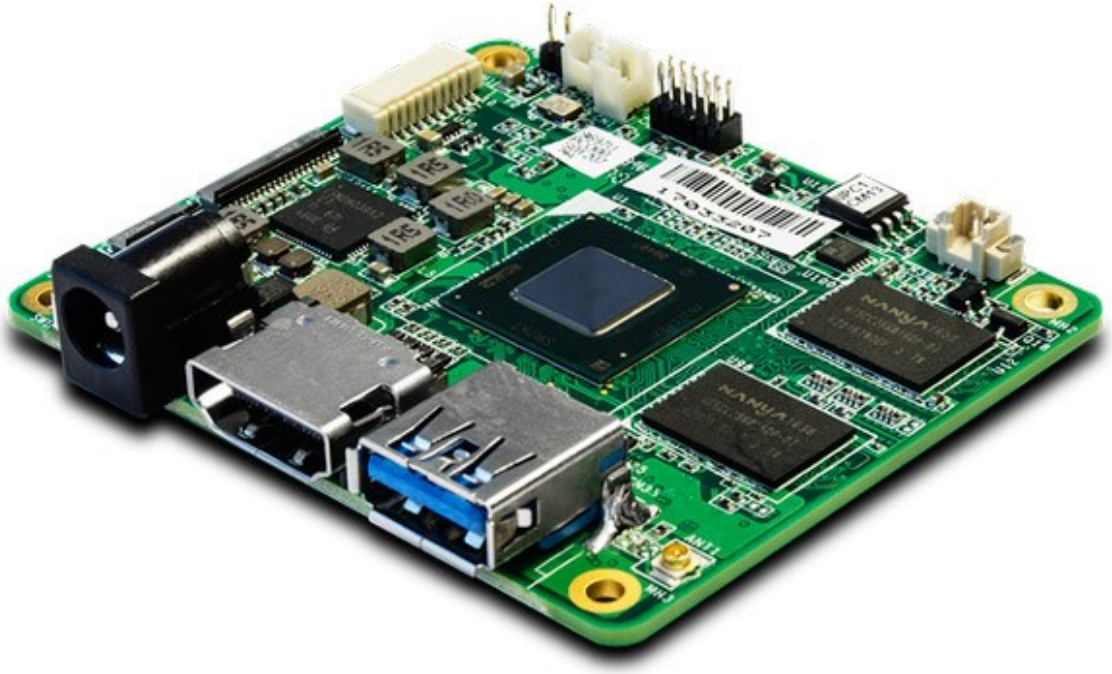

Fig. 7 – Intel Atom – Quad Core x5 [7]

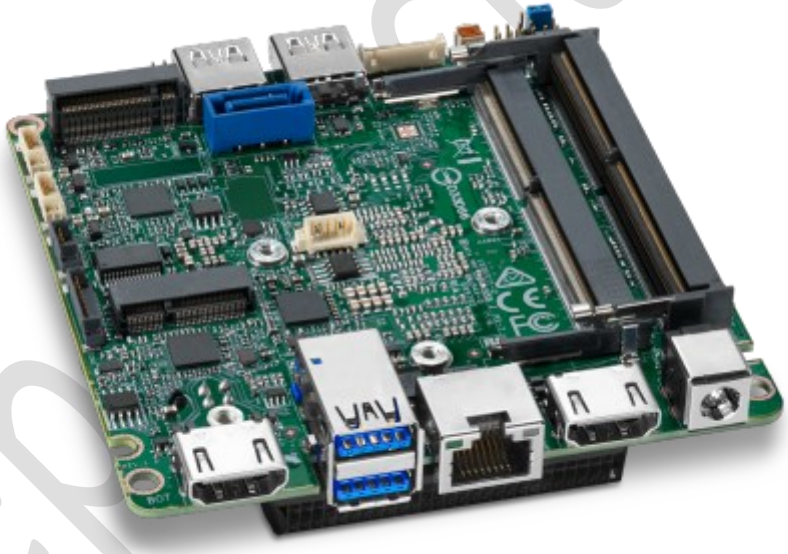

Fig. 8 – Intel NUC [8]

Realizat de: Asist. drd. Pintilie Lucian - Nicolae
Pentru disciplina: „Sisteme de calcul în timp real”
Adresă de e-mail: Lucian.Pintilie@emd.utcluj.ro

Fig. 9 – Orange PI [9]

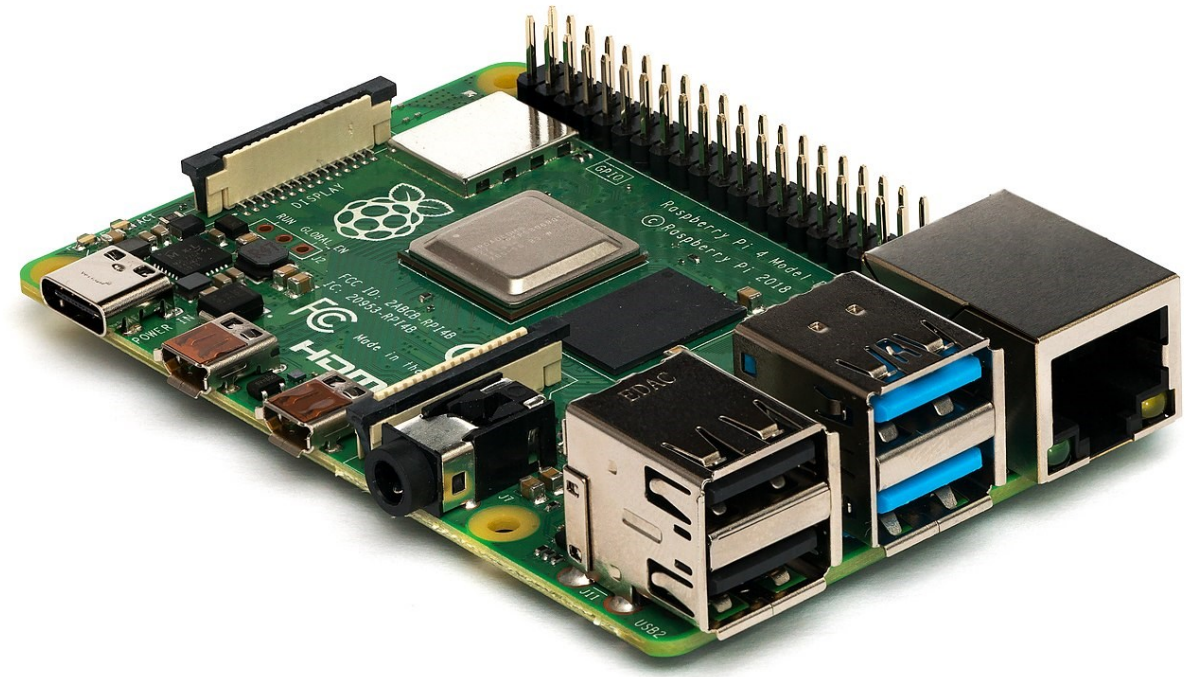

Fig. 10 – Raspberry PI 4 - Model B [10]

Realizat de: Asist. drd. Pintilie Lucian - Nicolae
Pentru disciplina: „Sisteme de calcul în timp real”
Adresă de e-mail: Lucian.Pintilie@emd.utcluj.ro

II. METODE DE UTILIZARE ȘI PROGRAMARE:

Pentru a proiecta o aplicație cu ajutorul unui astfel de micro - sistem de calcul, este necesară determinarea rolului său funcțional în cadrul aplicației. Câteva exemple de roluri funcționale ale unui astfel de sistem de calcul ar fi:
- Server dedicat unei aplicații (ex. server pentru instrumentație SCADA, server OPC);
- Server de comunicație (ex. rutarea diverselor protocoale ex. ModBus – TCP / IP);
- Server centralizator de informații (ex. SQL, Borlan – Paradox Engine, FOX PRO server);
- Server WEB pentru interfațarea proceselor de la distanță;
- Interfață de interacțiune între proces și factorul uman (eng. Human Machine Interface - HMI);

Deoarece, majoritatea aplicațiilor necesită executarea lor în fundal sau în mod „server", nu este deci absolut necesară utilizarea interfeței grafice (eng. GUI – Graphical User Interface) la nivel local (adică la nivelul micro – computerului). Astfel, pentru a deservi rolurile amintite mai sus, se preferă în mod preponderent sistemul de operare de tip Embedded Linux sau cu nucleu (eng. kernel) UNIX. Avantajul major al sistemelor de operare de tip UNIX / Linux constă în faptul că, majoritatea funcțiilor și aplicațiilor pot rula în absența interfeței grafice, și pot răspunde tuturor necesităților utilizatorului. De asemenea, este un sistem de operare cu un consum redus de resurse, care, permite accesul la consola de comandă de la distanță prin intermediul protocolului SSH (eng. Secure Shell). Sistemele de operare de tip UNIX / Linux sunt specializate pentru aplicațiile de tip „server" în care procesele și serviciile se pot executa în fundal, fără ca să perturbe funcționarea normală a aplicației de bază.

Un bun exemplu de micro – computer cu sistem de operare de tip UNIX / Linux, ar fi platforma de dezvoltare RaspBerry PI. Respectivul micro – computer, dispune de toate perifericele necesare unui sistem de calcul dedicat aplicației.

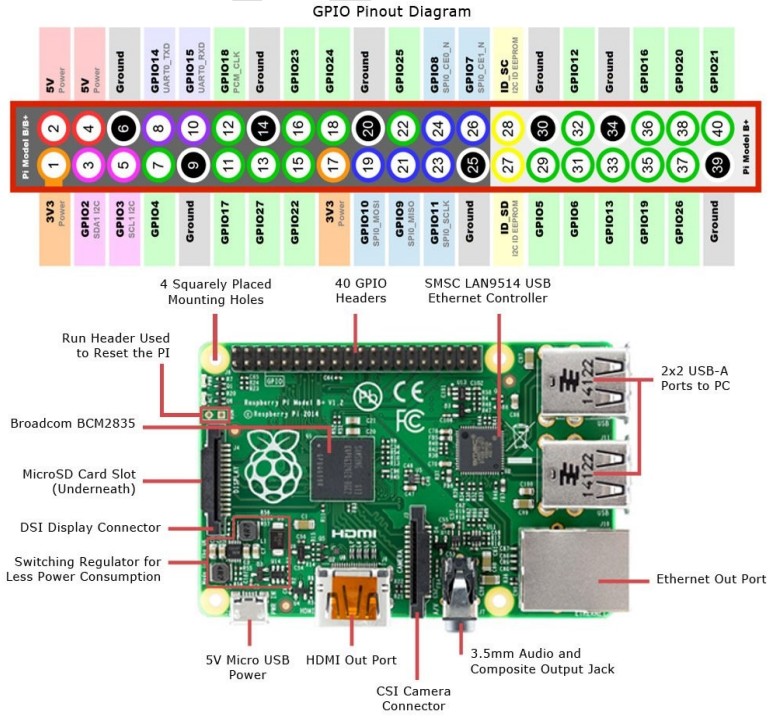

Fig. 11 – Raspberry PI 3 – Model B – prezentarea perifericelor [11]

Realizat de: Asist. drd. Pintilie Lucian - Nicolae
Pentru disciplina: „Sisteme de calcul în timp real"
Adresă de e-mail: Lucian.Pintilie@emd.utcluj.ro

Programarea sau dezvoltarea aplicațiilor pe baza unui astfel de dispozitiv, se poate realiza în diverse moduri (fie prin limbaje grafice de programare, fie prin limbaje scrise):
- Programare în mod direct, utilizând limbajul Python;
- Programare în mod indirect, utilizând mediul de simulare și testare Matlab – Simulink;
- Programare în mod indirect, utilizând platforma grafică WEB, Node – Red;

În continuare, vor fi exemplificate aplicații din fiecare categorie. Astfel, se va utiliza următorul montaj experimental în care se vor regăsi:
- patru diode electroluminiscente (LED) colorate diferit;
- două butoane cu apăsare și revenire (eng. PushButton);

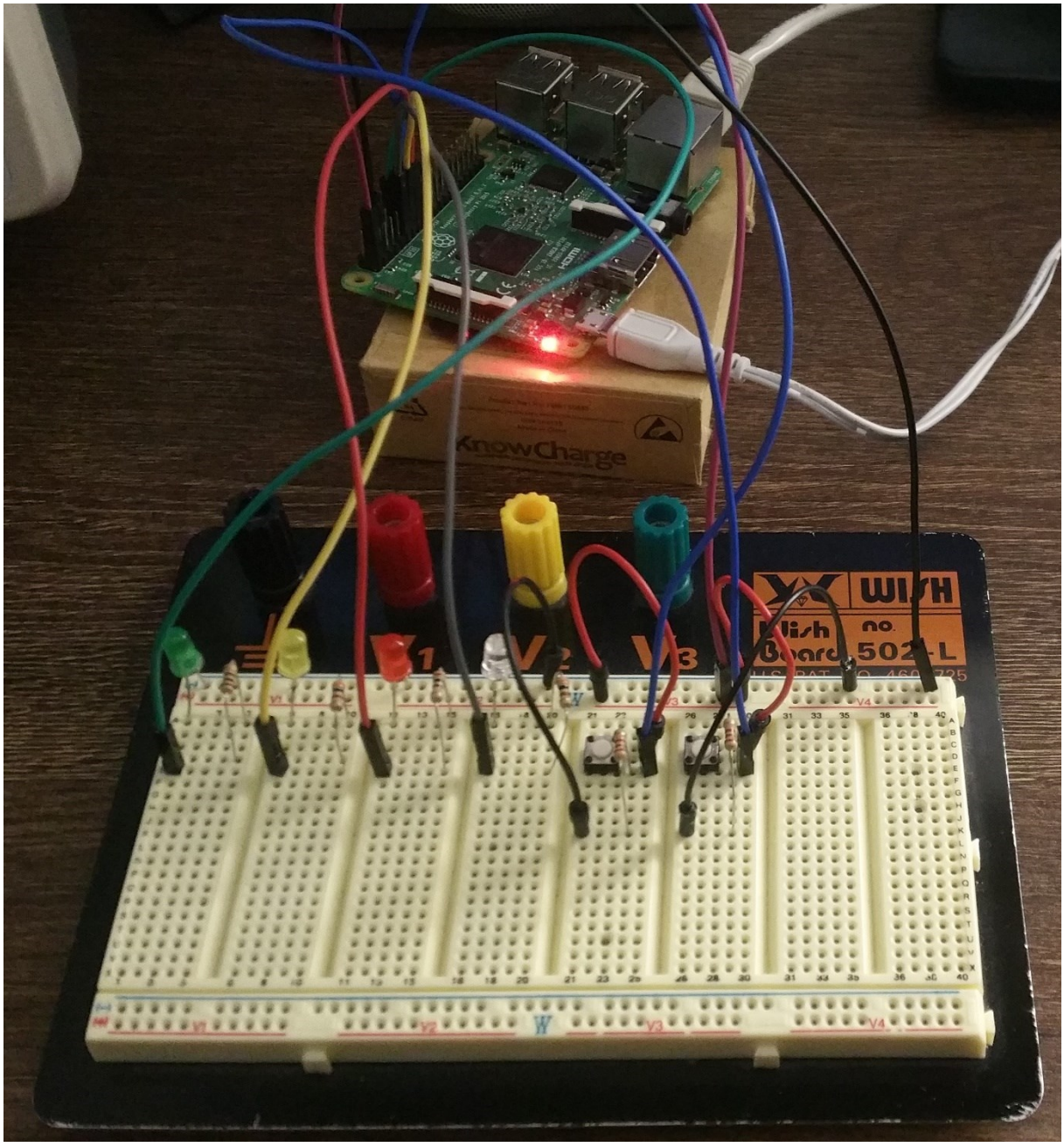

Fig. 12 – Montaj experimental

Realizat de: Asist. drd. Pintilie Lucian - Nicolae
Pentru disciplina: „Sisteme de calcul în timp real”
Adresă de e-mail: Lucian.Pintilie@emd.utcluj.ro

Elementele din circuit sunt conectate la platforma RaspBerry PI conform tabelului:

Element din circuit	Terminal de conexiune	Funcție terminal
Diodă electroluminiscentă (LED) verde	GPIO_18	Ieșire digitală
Diodă electroluminiscentă (LED) galbenă	GPIO_27	Ieșire digitală
Diodă electroluminiscentă (LED) roșie	GPIO_22	Ieșire digitală
Diodă electroluminiscentă (LED) albă	GPIO_17	Ieșire digitală
Buton cu apăsare și revenire 1	GPIO_23	Intrare digitală
Buton cu apăsare și revenire 2	GPIO_24	Intrare digitală

1. Programare în limbaj Python:

În vederea dezvoltării unei aplicații în limbaj Python, este necesară conectarea platformei RaspBerry Pi la rețeaua locală Ethernet. De asemenea, este importantă cunoașterea adresei IP din rețea a platformei. Determinarea adresei IP, poate fi realizată pe baza paginii de administrare a dispozitivului local de rutare sau pe baza unor programe specializate în sondarea (scanarea) adreselor de rețea, precum Angry IP Scanner.

Pentru a accesa consola de comandă a platformei, este necesară instalarea programului PuTTY. În cadrul acestui program se va menționa adresa IP a platformei RaspBerry PI, iar prin intermediul protocolului SSH, se va accesa consola de comandă.

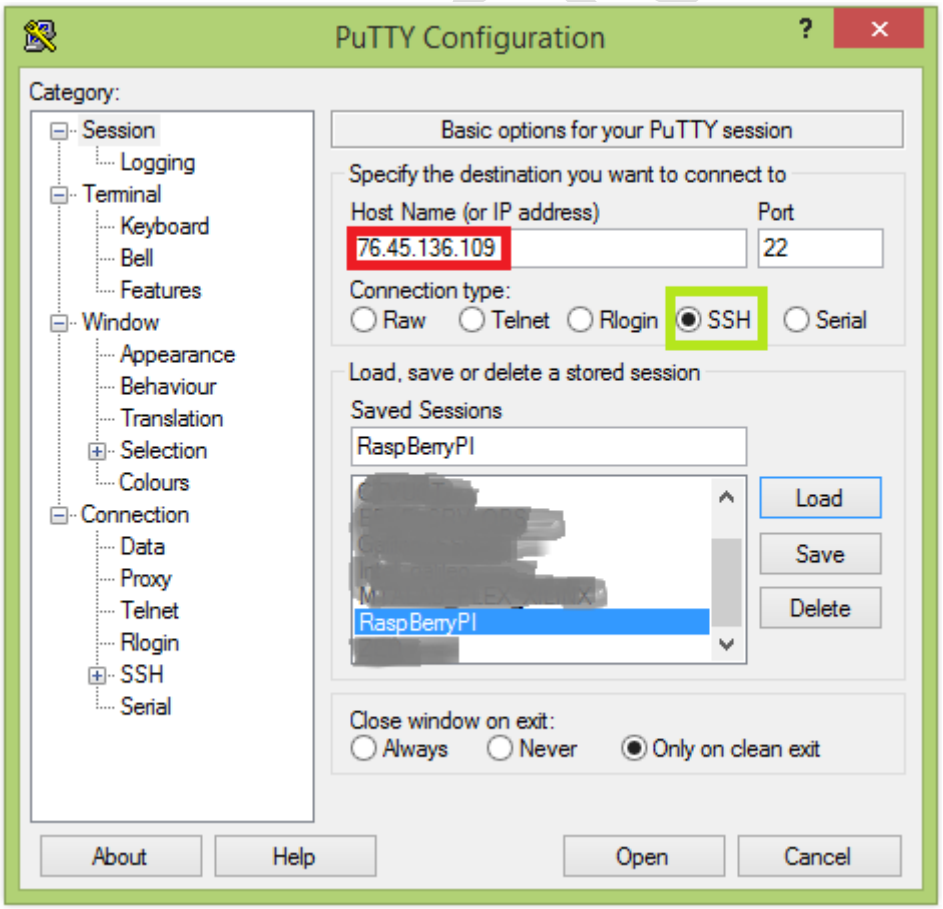

Fig. 13 – Configurarea aplicației PuTTY - specificarea adresei IP (roșu) și a protocolului (verde)

Realizat de: Asist. drd. Pintilie Lucian - Nicolae
Pentru disciplina: „Sisteme de calcul în timp real”
Adresă de e-mail: Lucian.Pintilie@emd.utcluj.ro

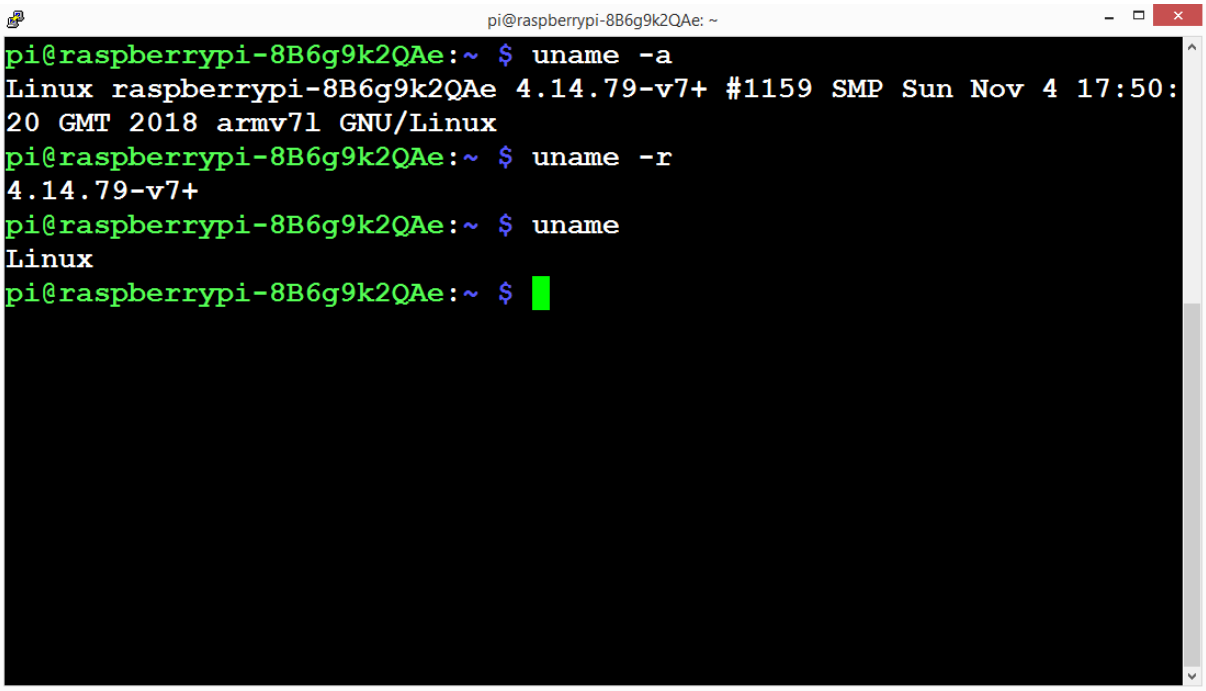

Fig. 14 – Consola de comandă a platformei RaspBerry PI prin intermediul programului PuTTY

Dezvoltarea aplicației pe platforma RaspBerry PI se va realiza cu ajutorul calculatorului gazdă și a programului PuTTY conform schemei principiale de mai jos:

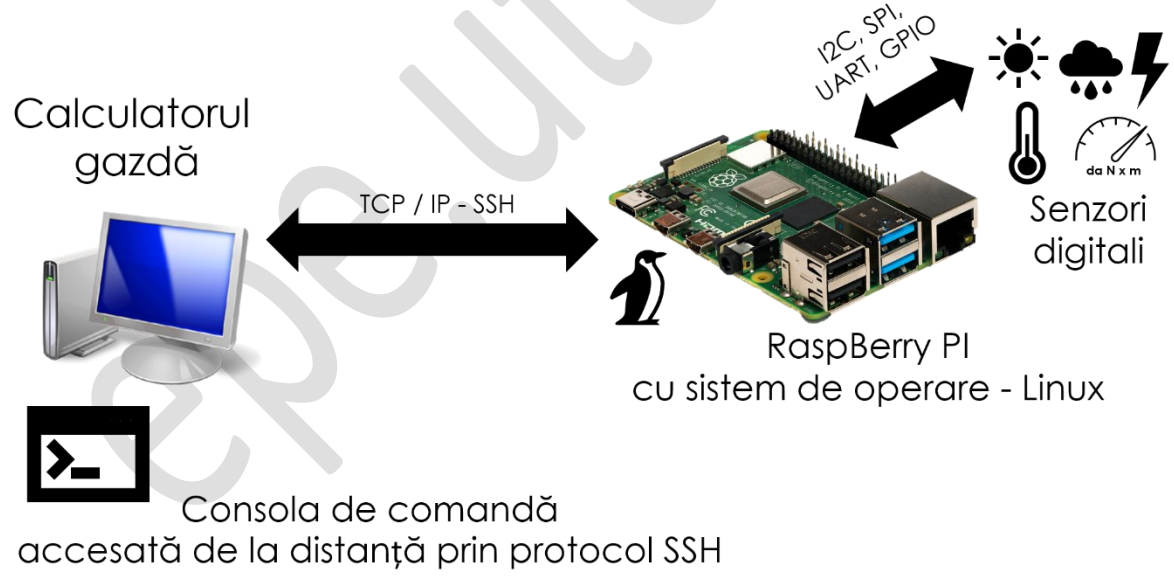

Fig. 15 – Dezvoltarea aplicațiilor în limbaj Python prin intermediul calculatorului gazdă

În consola de comandă a platformei RaspBerry PI, se va lansa în execuție editorul de text „NANO” cu ajutorul comenzii (se va înlocui <nume_fișier_nou> cu numele dorit):

sudo nano <nume_fișier_nou>.py

Realizat de: Asist. drd. Pintilie Lucian - Nicolae
Pentru disciplina: „Sisteme de calcul în timp real”
Adresă de e-mail: Lucian.Pintilie@emd.utcluj.ro

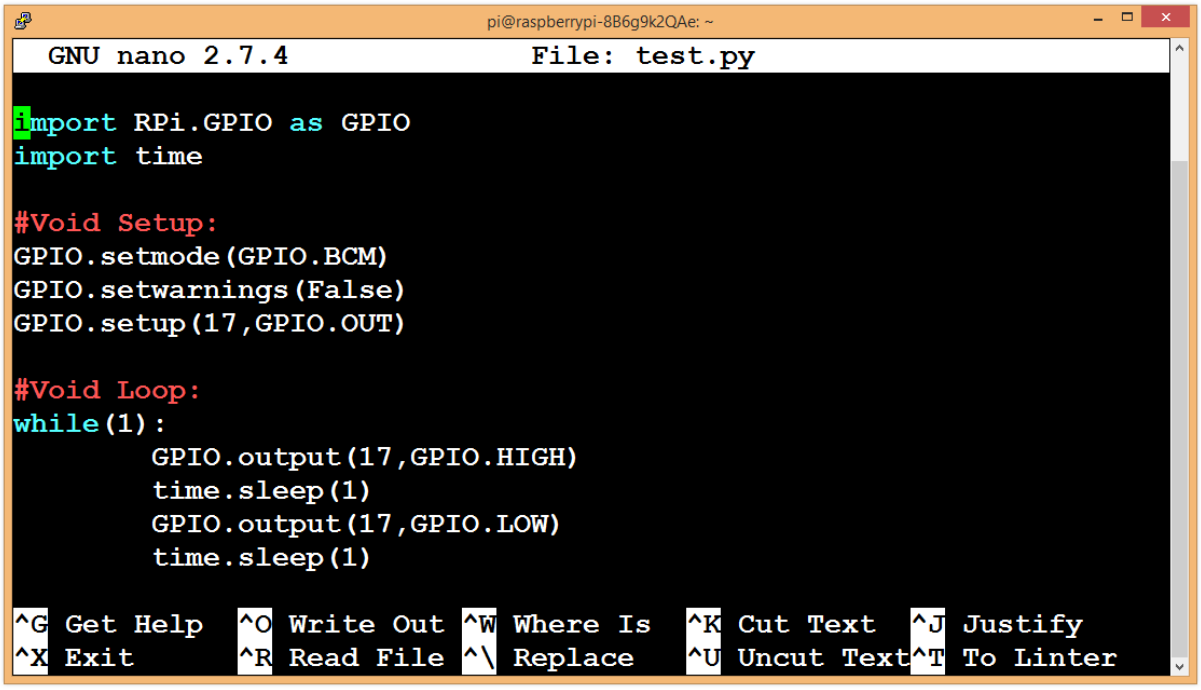

Fig. 16 – Editorul de text NANO – redactarea conținutului aplicației „test.py”

Pentru exemplificare, se va creea un nou fișier - script Python cu numele „test.py”, care va avea următorul conținut:

```
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)
```

Cu ajutorul combinației de taste „Ctrl + X” se va închide sesiunea de editare a fișierului, iar prin intermediu tastei „Y” se va confirma aplicarea modificărilor creeate fișierului original. În urma acestor operații, se va putea accesa din nou consola de comandă.

Pentru a executa aplicația creeată se va introduce comanda:

```
python test.py
```

Realizat de: Asist. drd. Pintilie Lucian - Nicolae
Pentru disciplina: „Sisteme de calcul în timp real”
Adresă de e-mail: Lucian.Pintilie@emd.utcluj.ro

În urma executării aplicației, dioda electroluminiscentă (LED) de culoară albă, va semnaliza în mod intermitent. Pentru a suspenda execuția aplicației, se va introduce combinația de taste „Ctrl + C”.

OBSERVAȚIE: 1. Față de limbajul C++ / C standard limbajul Python introduce următoarele reguli de formatare a codului – program:

Funcție - comandă	C++ / C standard	Python
Introducerea bibleotecilor de funcții	#include	import
Introducerea de comentarii	// sau */	#
Delimitarea structurilor de cod	{ }	Aliniere la dreapta

2. Față de limbajul Wiring C (Arduino IDE) există următoarele diferețe și comenzi – funcții echivalente:

Funcție - comandă	Wiring C - Arduino	Python
Definire funcție pin / terminal	pinMode()	GPIO.setup()
Stabilire stăre digitală pin / terminal	digitalWrite()	GPIO.output()
Introducere timp de întârziere	delay()	time.sleep()

Al doilea exemplu, constă în sesizarea modificării stării digitale a unui pin / terminal care are funcția de intrare digitală la care este atașat un buton cu apăsare și revenire. Pentru a creea un nou fișier, se va introduce următoare comandă în consolă:

```
sudo nano switch.py
```

În conținutul fișierului – script Python se vor regăsi următoarele declarații:

```
import RPi.GPIO as GPIO
import time

#Void Setup:
GPIO.setmode(GPIO.BCM)
GPIO.setwarnings(False)
GPIO.setup(23,GPIO.IN, pull_up_down=GPIO.PUD_DOWN)

#Void Loop:
while(1):
        if GPIO.input(23) == GPIO.HIGH:
                print("Button pressed!")
        time.sleep(0.5)
```

Din nou, cu ajutorul combinației de taste „Ctrl + X” se va închide sesiunea de editare a fișierului, iar prin intermediu tastei „Y” se va confirma aplicarea modificărilor creeate.

Realizat de: Asist. drd. Pintilie Lucian - Nicolae
Pentru disciplina: „Sisteme de calcul în timp real”
Adresă de e-mail: Lucian.Pintilie@emd.utcluj.ro

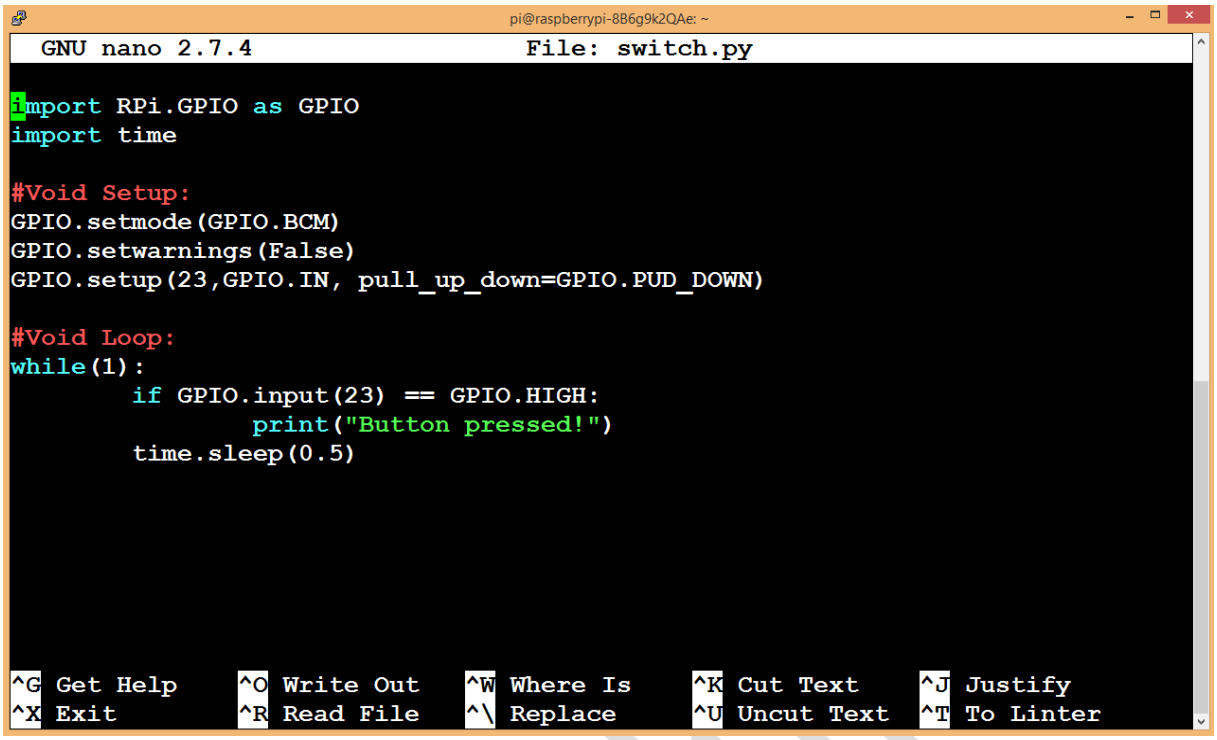

Fig. 17 – Editorul de text NANO – redactarea conținutului aplicației „switch.py”

Pentru a executa aplicația creeată se va introduce comanda:

python switch.py

În urma executării aplicației, în consolă nu se va afișa nimic, până la apăsarea butonului fizic atașat la intrarea digitală GPIO_23. La apăsarea butonului în consolă se va afișa mesajul:

Button pressed!

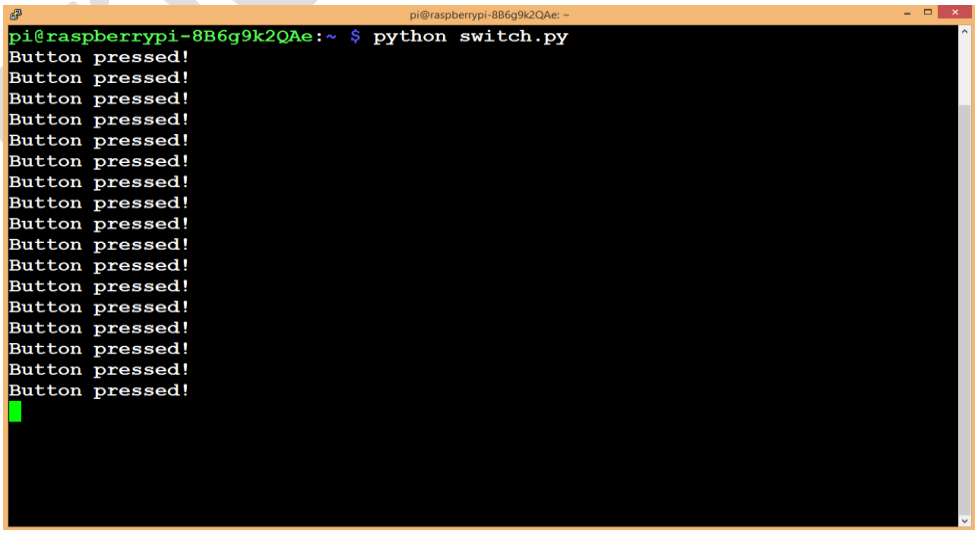

Fig. 18 – Executarea aplicației „switch.py” – afișarea mesajului „Button pressed!” în consolă

Realizat de: Asist. drd. Pintilie Lucian - Nicolae
Pentru disciplina: „Sisteme de calcul în timp real”
Adresă de e-mail: Lucian.Pintilie@emd.utcluj.ro

2. Programare cu ajutorul mediului Matlab - Simulink:

În cadrul mediului Matlab – Simulink, la fel ca și în cazul platformei Arduino, platforma RaspBerry PI, are o vastă paletă de instrumente, în cadrul căreia se regăsesc și blocurile pentru control al ieșirilor și intrărilor digitale de uz general (GPIO). Pe baza acestor blocuri, și pe baza instrumentelor standard Simulink pentru observare și interacționare cu modelul (ex. osciloscop, afișaj, comutator manual sau constante), se va concepe un model simplu de comandă și achiziție de date de la intrările și ieșirile de uz general:

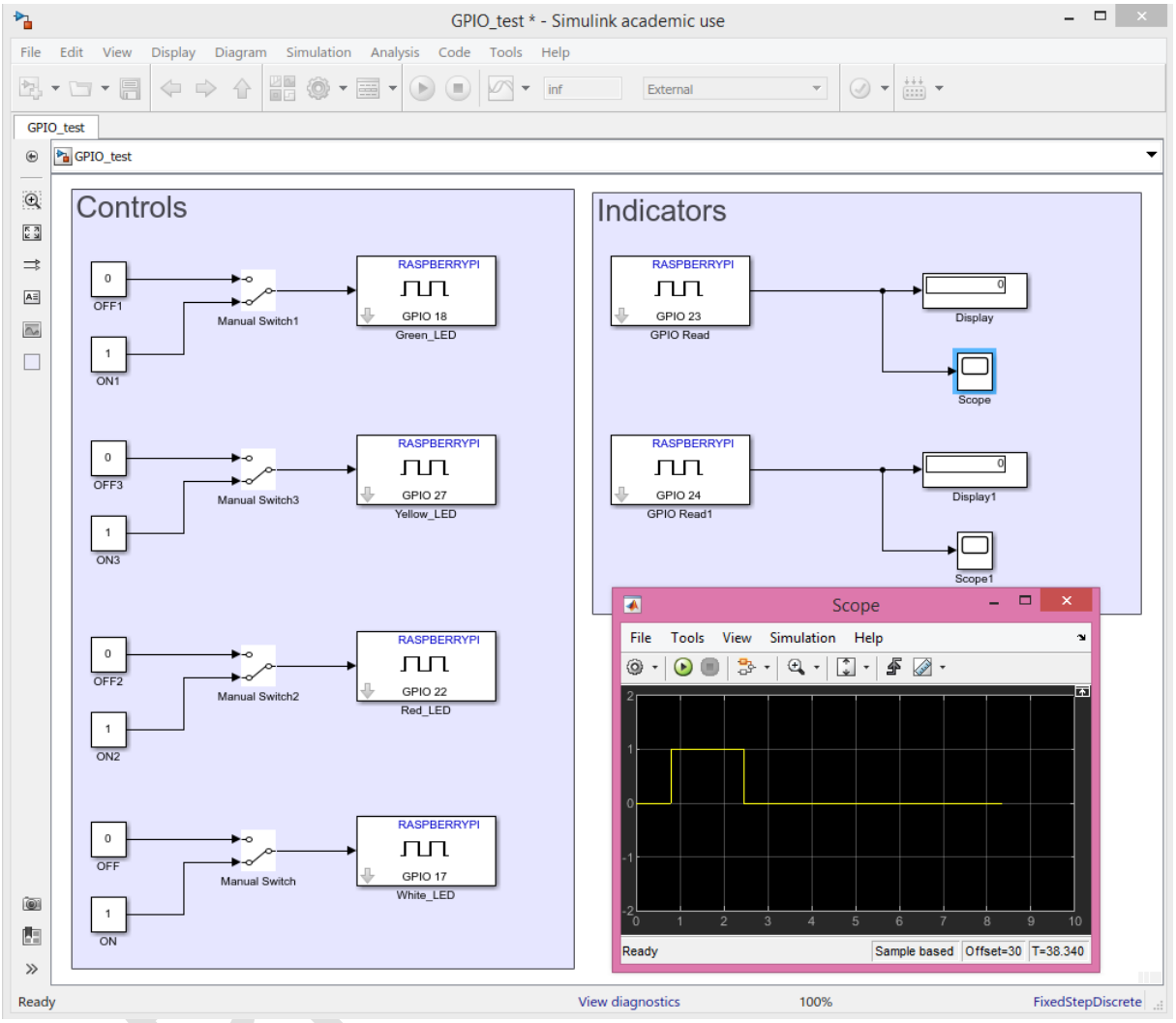

Fig. 19 – Modelul Matlab – Simulink de interacțiune în timp real cu platforma RaspBerry PI

Executarea modelului se va realiza în mod „Extern” (External), adică, modelul va fi transpus în limbaj Python, după care va fi încărcat în memoria platformei RaspBerry Pi. Între calculatorul gazdă și platformă se va stabili o conexiune de tip „TCP / IP” permițând astfel modificarea parametrilor elementelor din circuit în timp real în timpul rulării simulării.

Prin acționarea comutatoarelor manuale, se vor comuta stările ieșirilor digitale, aprinzând sau stingâng diodele electroluminiscente. Pe afișaje se va afișa valoarea „1” la apăsarea butoanelor legate la platformă. Pe osciloscopul virtual se va observa evoluția în timp a stării digitale a unui pin / terminal.

Realizat de: Asist. drd. Pintilie Lucian - Nicolae
Pentru disciplina: „Sisteme de calcul în timp real”
Adresă de e-mail: Lucian.Pintilie@emd.utcluj.ro

3. Programare cu ajutorul platformei Node - Red:

Pentru a lansa în execuție serverul WEB Node – Red, se va inroduce comanda:

```
node-red-start
```

În urma acestei comenzi, se vor afișa în consolă instrucțiunile de accesare a paginii WEB pentru administrare a panoului frontal de interacțiune:

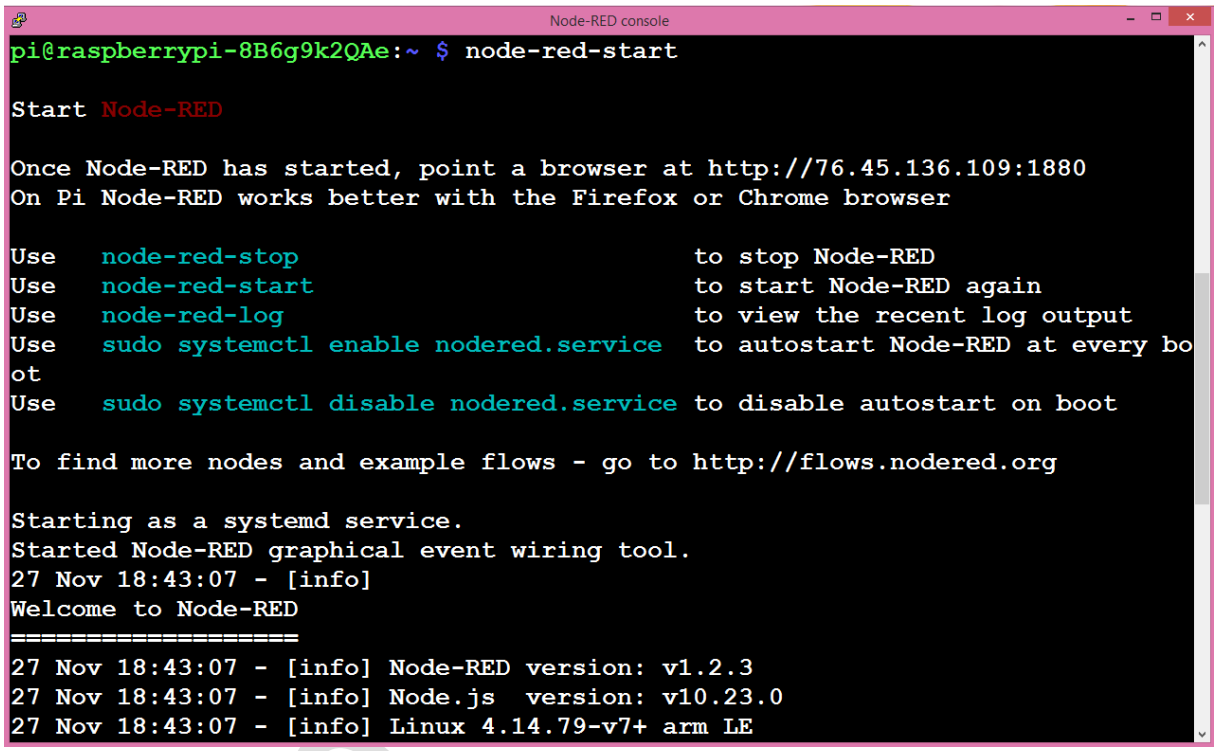

Fig. 20 – Lansarea în execuție a serverului WEB Node – Red

În cadrul unui program pentru navigare WEB (ex. Internet Explorer, Mozilla, Google Chrome), în bara de adrese, se va trece următoarea adresă:

http://<adresă IP RaspBerry PI>:1880/

(În cazul de față, adresa IP a platformei RaspBerry PI este 76.45.136.109, deci adresa de accesare a paginii de administrare va fi http://76.45.136.109:1880/).

În cadrul paginii de administrare se regăsesc diverse instrumente necesare implementării unei aplicații WEB, precum:
- elemente indicatoare (ex. indicatoare cu ac, grafice, indicatoare de nivel etc.);
- elemente de control (ex. comutatoare acționate manual, cursoarea liniare etc.);

Pe lângă elementele interfeței grafice, se pot regăsi, de asemenea, diverse blocuri pentru funcții de calcul și condiționare, dar, cel mai important, se regăsesc blocuri pentru interacțiune cu intrările și ieșirile de uz general ale platformei RaspBerry PI.

Realizat de: Asist. drd. Pintilie Lucian - Nicolae
Pentru disciplina: „Sisteme de calcul în timp real”
Adresă de e-mail: Lucian.Pintilie@emd.utcluj.ro

Pe baza blocurilor de control ale intrărilor și ieșirilor digitale de uz general se va construi următorul model. Pentru a afișa starea digitală a intrărilor se vor folosi două indicatoare cu ac și un afișaj grafic în timp. În urma finalizării modelului prin intermediul butonului „Deploy” (încărcare model), se va transfera modelul în memoria platformei. Odată încărcat, modelul se va lansa în execuție în mod automat.

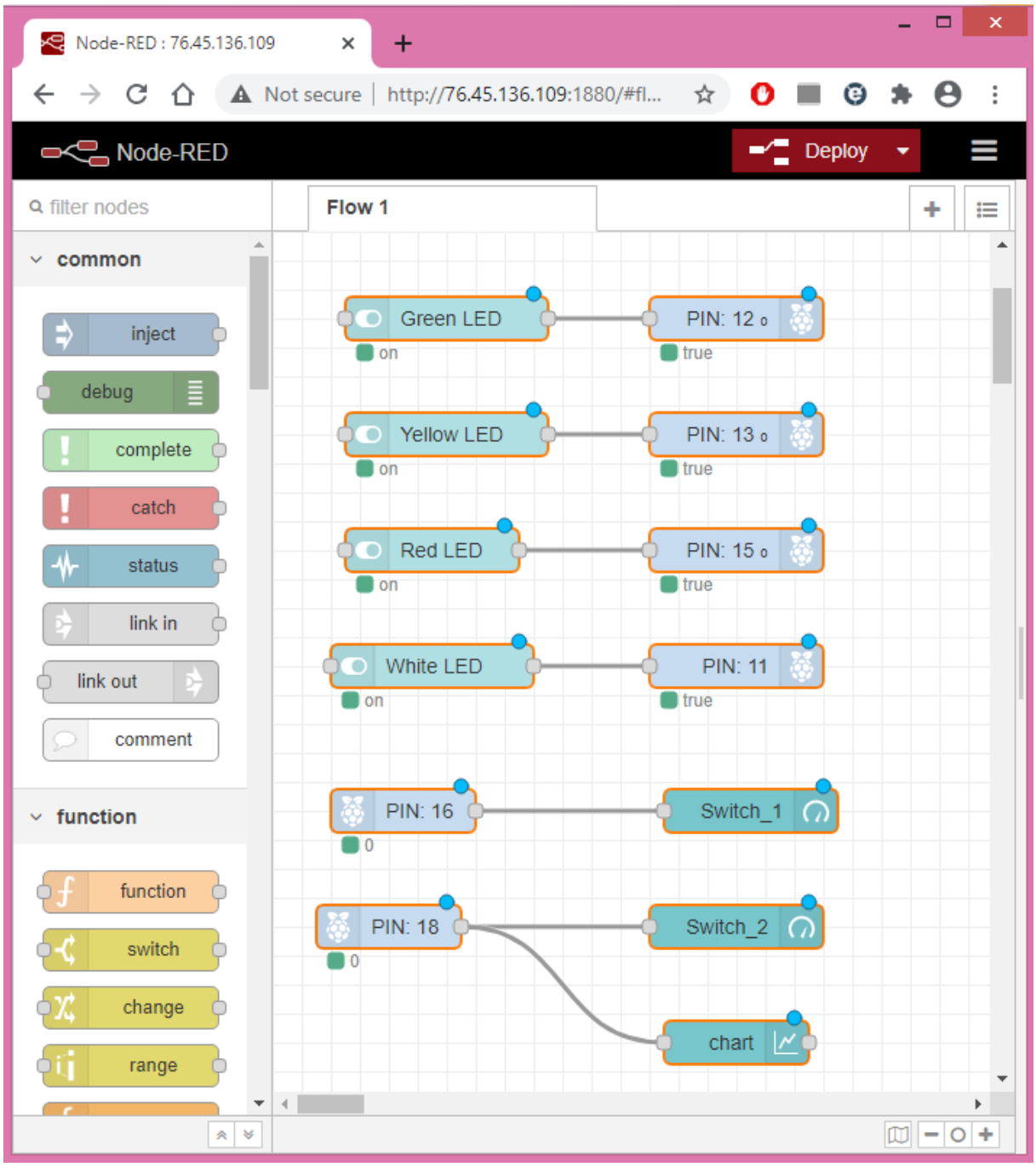

Fig. 21 – Interfața grafică WEB Node - Red pentru administrare a panoului frontal

Pentru a accesa panoul frontal proiectat cu ajutorul modelului implementat, se va folosi următoarea adresă: http://<adresă_IP_RaspBerry_PI>:1880/ui

Realizat de: Asist. drd. Pintilie Lucian - Nicolae
Pentru disciplina: „Sisteme de calcul în timp real”
Adresă de e-mail: Lucian.Pintilie@emd.utcluj.ro

Fig. 22 – Panoul frontal de interacțiune a utilizatorului cu elementele din circuit

Prin acționarea comutatoarelor din interfața „Controls” se vor putea comuta ieșirile digitale, la care au fost atașate diodele electroluminiscente. Prin acționarea butoanelor fizice cu apăsare și revenire, se vor deplasa acele indicatoarelor din categoria „Indicators” iar pe graficul „chart” se va înregistra variația în timp a stării digitale înregistrată de la al doilea buton.

Aplicația poate fi accesată de pe orice dispozitiv mobil (ex. telefon sau tabletă)!

Realizat de: Asist. drd. Pintilie Lucian - Nicolae
Pentru disciplina: „Sisteme de calcul în timp real”
Adresă de e-mail: Lucian.Pintilie@emd.utcluj.ro

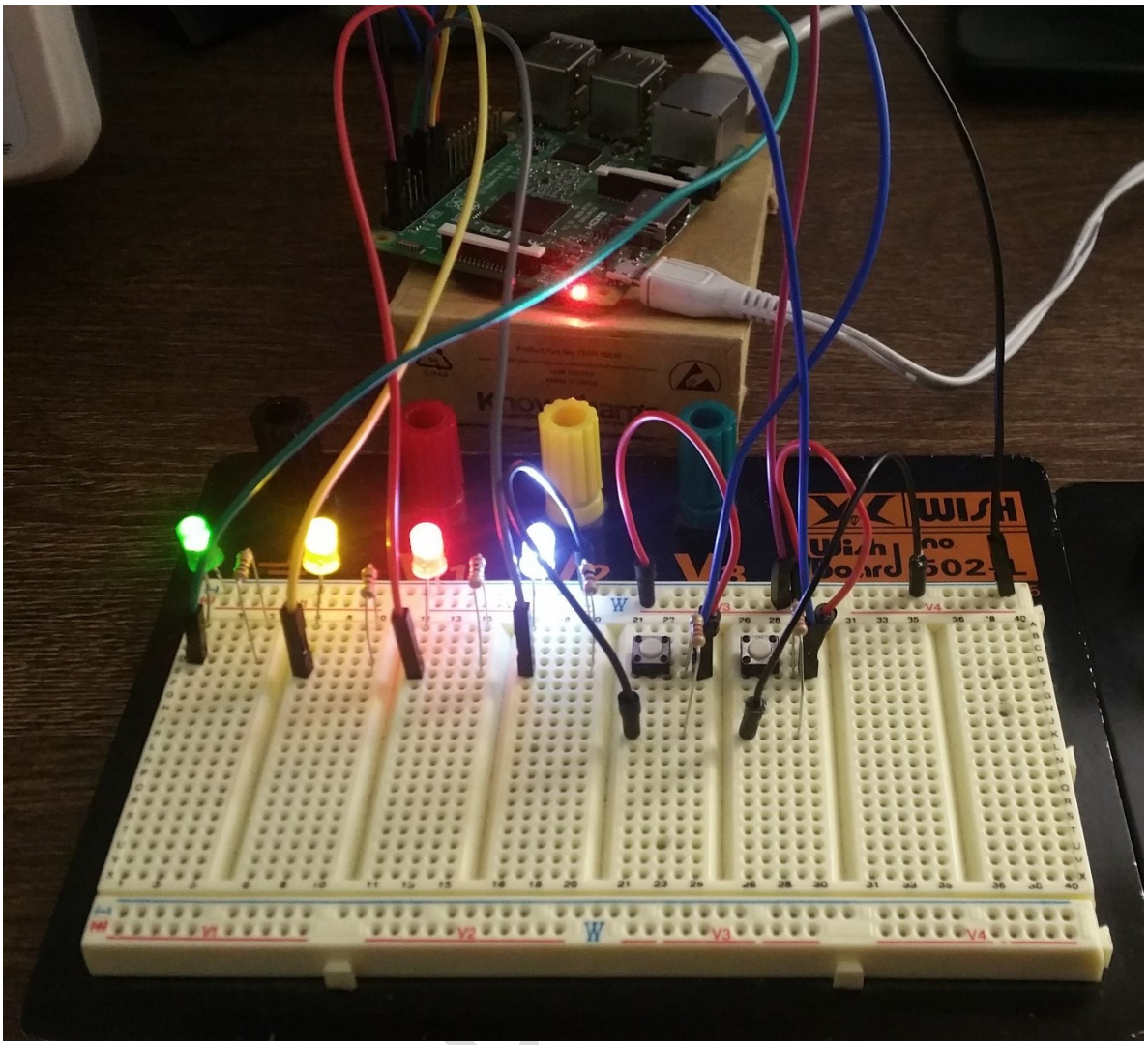

Fig. 23 – Aprinderea diodelor electroluminiscente la modificarea comutatoarelor virtuale

III. CONCLUZII:

1. Platformele de dezvoltare cu sistem de operare, permit lansarea în execuție a mai multor aplicații în mod simultan (datorită capabilității redată de funcția „multi – tasking”);

2. Sistemele de operare utilizate de către micro – computerele dedicate pentru aplicație, sunt sisteme de operare optimizate la maxim, fără interfață grafică, dar care permit accesarea de la distanță a funcțiilor vitale îndeplinite de sistemul de operare;

3. Controlul execuției unei aplicații este posibil datorită sistemului de operare (ex. lansarea în execuție sau suspendarea aplicației). Din acest motiv, dezvoltarea aplicațiilor nu necesită reprogramarea memoriei de lucru a platformei asemenea unui micro - controller;

4. Funcțiile deservite de către un micro – computer sunt preponderent orientate spre modul de lucru „server” și se preferă rularea în fundal a aplicațiilor în absența unei interfețe grafice.

Realizat de: Asist. drd. Pintilie Lucian - Nicolae
Pentru disciplina: „Sisteme de calcul în timp real”
Adresă de e-mail: Lucian.Pintilie@emd.utcluj.ro

IV. BIBLEOGRAFIE:

1. Windows CE:
https://en.wikipedia.org/wiki/Windows_CE_5.0

2. Windows Embedded (Industry PRO):
https://en.wikipedia.org/wiki/Windows_Embedded_Industry

3. Windows 10 core IoT:
https://docs.microsoft.com/en-us/windows/iot-core/develop-your-app/iotcoredefaultapp

4. Embedded Linux:
https://github.com/topics/embedded-linux?l=shell

5. Intel Galileo Gen II:
https://en.wikipedia.org/wiki/Intel_Galileo#/media/File:Embedded_World_2014_Intel_Galileo_01.jpg

6. Intel Edison:
https://upload.wikimedia.org/wikipedia/commons/4/49/Intel-Edison2.png

7. Intel Atom Quad-Core x5:
https://hothardware.com/news/up-core-dev-board-boasts-quad-core-intel-atom-x5-grunt-to-challenge-raspberry-pi

8. Intel NUC:
https://www.intel.com/content/www/us/en/products/boards-kits/nuc/boards/nuc7i3dnbe.html

9. Orange PI:
http://linuxgizmos.com/latest-orange-pi-offers-quad-a53-cores-and-2gb-of-ram/

10. Raspberry PI 4 – Model B:
https://en.wikipedia.org/wiki/Raspberry_Pi#/media/File:Raspberry_Pi_4_Model_B_-_Side.jpg

11. RaspBerry PI 3 – Model B – prezentarea perifericelor:
https://www.jameco.com/Jameco/workshop/circuitnotes/raspberry-pi-circuit-note.html

12. Bibleoteca UTCLUJ – Editura UTPress Cluj-Napoca – „Sisteme Embedded în Inginerie Electrică" – Ioana – Cornelia GROS, Lucian – Nicolae PINTILIE, Teodor Crișan PANĂ; ISBN: 2020 ISBN 978-606-737-431-5;
https://biblioteca.utcluj.ro/files/carti-online-cu-coperta/431-5.pdf

Realizat de: Asist. drd. Pintilie Lucian - Nicolae
Pentru disciplina: „Sisteme de calcul în timp real"
Adresă de e-mail: Lucian.Pintilie@emd.utcluj.ro