



Facultatea de Automatica si Calculatoare
Departamentul de Calculatoare

Raspberry Pi - Twitter

Moldovan Cătălin
Grupa: 30225

Coordonator: prof. dr. ing. Radu Munteanu
Data: 18.01.2018



Cuprins

1. Introducere	3
Descrierea soluției de rezolvare	3
Domeniul de aplicație	3
Avantaje față de soluțiile cunoscute	3
2. Noțiuni introductive	3
3. Software	4
4. Comenzi Python.....	5
5. Componente	6
6. Instalare	7
7. Conectare WiFi	9
8. Accesare Raspberry Pi de la distanță	10
9. Comenzi	11
10. Montaj	12
11. Acces Twitter	13
12. Simulare.....	14
13. Cod	14
14. Părți de cod.....	17
Configurare LCD	17
Excepții	17
Selectare părți Tweet	18
Organizare Tweet	18
Fișier de cuvinte jignitoare	19
Filtrare limbaj inadecvat.....	19
Afișare Tweet pe LCD	20
15. Rulare aplicație.....	21
Rulare aplicație pe consolă	21
Rulare aplicație pe display LCD	21
15. Listă componente	23
16. Referințe	24



Proiect Raspberry Pi - Twitter

Moldovan Cătălin

Grupa: 30225

Data: 18.01.2018

E-mail: catalin.moldovan97@gmail.com

GitHub: <https://github.com/catamold/raspberrypi-twitter-lcd>

Universitatea Tehnică din Cluj-Napoca

1. INTRODUCERE

DESCRIEREA SOLUȚIEI DE REZOLVARE:

Proiectul va folosi un display conectat la Raspberry Pi pentru afișarea newsfeed-ului sau tweet-urilor cu un anumit hashtag. În unele cazuri mesajul este nevoie să fie afișat pe 2 rânduri, de exemplu pentru display-ul LCD 16x2. Fiecare tweet poate fi setat să fie vizibil 30 de secunde, iar mai apoi să urmeze altă postare. Timpul de vizibilitate și alegerea hashtag-ului se pot modifica cu ajutorul unei telecomande care transmite informația către un *IR Remote* conectat la Raspberry Pi.

Mesajele postate pe *Twitter* vor trece printr-un sistem de selecție, prin care se verifică dacă mesajul conține cuvinte obscene, injurii, iar apoi tweet-ul este sau nu afișat. Astfel toate cuvintele nedorite sunt stocate într-un fișier text.

DOMENIUL DE APLICAȚIE:

Python, Raspbian (Sistem de operare Raspberry Pi)

AVANTAJE FAȚĂ DE SOLUȚIILE CUNOSCUTE:

În cadrul conferințelor/hackathon/concurs este foarte utilă pentru afișarea unor tweet-uri în timp real ale participanților sau a oricărei persoane cu un anumit hashtag (ex. #hackathon2k17Cluj). Totodată este foarte utilă în a citi newsfeed-ul fără a interacționa cu pagina *Twitter* de pe un calculator/telefon.

2. NOȚIUNI INTRODUCTIVE

Raspberry Pi^[1] este o placă de dezvoltare de tip SBC (Single Board Computer) – un sistem de calcul nemodular implementat pe un singur cablaj electronic. Raspberry Pi este un calculator complet permițând funcționalități obișnuite precum rularea unui sistem de operare (Linux sau Windows) și rularea de aplicații utilizator (jocuri, editoare de text, medii de programare, redarea de muzică și filme, aplicații de teleconferință, aplicații Internet). În plus, placa Raspberry Pi oferă posibilitatea de a conecta diverse componente electronice specifice sistemelor embedded: senzori, butoane, ecrane LCD sau pe 7 segmente, drivere de motoare, relee etc.



Placa Raspberry Pi 3 dispune de periferice integrate numeroase acoperind complet funcționalitatea unui sistem de calcul (audio, video, porturi USB, conectivitate de rețea):

- Procesor SoC pe 64 de biți din familia ARMv8-A, Broadcom BCM2837, ce lucrează la o frecvență de 1.2GHz și dispune de 4 nuclee de tip ARM Cortex-A53;
- 1GB de memorie RAM (folosită și ca memorie video, partajată cu procesorul grafic);
- Procesor grafic Broadcoam VideoCore IV 3D integrat pe aceeași pastilă de siliciu ca și procesorul principal;
- Ieșire digitală video / audio HDMI;
- Ieșire analogică video (composite video) / audio mixtă prin intermediul unei mufe jack 3.5mm;
- Mufă de rețea RJ45 Ethernet 10/100 Mbit/s;
- Conectivitate WiFi 802.11n;
- Conectivitate Bluetooth 4.1 / BLE;
- 4 porturi USB 2.0;
- 40 de pini de intrare / ieșire (GPIO);
- Slot card de memorie microSD (utilizat pentru instalarea sistemului de operare);
- Conectori dedicați pentru cameră video (CSI) și afișaj (DSI);

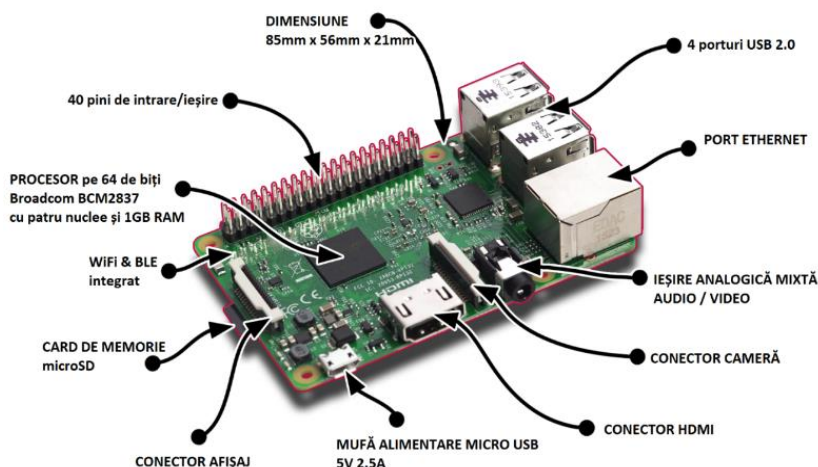


Figura 1. Plăcuța Raspberry Pi 3 Model B

3. SOFTWARE

Python^[2] este un limbaj de programare dinamic multi-paradigmă, creat în 1989 de programatorul olandez Guido van Rossum. Python pune accentul pe curățenia și simplitatea codului, iar sintaxa sa le permite dezvoltatorilor să exprime unele idei programatice într-o manieră mai clară și mai concisă decât în alte limbaje de programare ca C.

Python este un limbaj multi-paradigmă, concentrându-se asupra programării imperative, orientate pe obiecte și funcționale, ceea ce permite o flexibilitate mai mare în scrierea aplicațiilor. Din punctul de vedere al sintaxei, are un număr de construcții și cuvinte cheie cunoscute oricărui programator, dar prezintă și un concept unic: nivelul de indentare are semnificație sintactică. Blocurile de cod sunt delimitate prin simplă indentare.



Python include biblioteci pentru lucrul cu fișiere, arhive, fișiere XML și un set de biblioteci pentru lucrul cu rețeaua și principalele protocoale de comunicare pe internet (HTTP, Telnet, FTP).



Figura 2. Logo limbaj Python

4. COMENZI PYTHON

Declarare variabile:

```
1. a = 10 # a va fi considerat un număr întreg (integer)
2. b = 11.5 # b va fi considerat un număr rațional (double sau float)
3. c = "un rând de text" # c va fi considerat un șir (string)
4.
5. lista_mea = ["măr", "pară"]
6. lista_mea.append("strugure") # adaugă la listă
7. print lista_mea[2] # afișază: strugure
```

Afișare text:

```
1. print "Hello World!"
```

Comentarii:

```
1. #comentariu Python
```

Instrucțiunea IF:

```
1. if (statement):
2.     ...
3. else:
4.     ...
```

Instrucțiunea TRY:

```
1. try:
2.     ...
3. except IOError:
4.     ...
```

Instrucțiunea FOR:

```
1. for x in range(0, 9):
2.     ...
3.     break
```

Instrucțiunea WHILE:

```
1. while (statement):
2.     ...
3.     break
```



Instrucțiunea SWITCH:

1. `choices = {'a': 1, 'b': 2}`
2. `result = choices.get(key, 'default')`

5. COMPONENTE

Un *breadboard* este un dispozitiv care permite conectarea extrem de simplu a componentelor electronice, fără lipituri. Pentru a conecta dispozitivele, se folosesc *fire tată-tată* (cu pini la ambele capete), care se introduc în găurile din breadboard. Găurile existente în breadboard sunt conectate între ele (de obicei pe linie), astfel încât firele introduse pe aceeași linie vor fi conectate între ele.

T-Cobbler Plus este un add-on special conceput pentru conectarea mai ușoară a dispozitivelor în pinii de pe placa Raspberry Pi. Conectarea se face între cei 2x20 pini Raspberry Pi și un breadboard.

LCD-ul poate să afișeze 16 caractere pe 2 rânduri (în total 32 de caractere), are lumina de fundal de culoare albastră. Pentru a-l folosi cu Raspberry PI, este nevoie de fire de conectare și de un potențiomtru de 10K pentru reglarea contrastului (sau folosirea unor rezistențe). După ce am reglat potențiomtrul astfel încât să am un afișaj clar, l-am înlocuit cu o rezistență de 5K. Codul sursă utilizează librăria *CharLCD*^[3] (`sudo pip3 install CharLCD`) și *RPLCD*^[4] (`sudo pip3 install RPLCD`). La începutul fișierului python se va scrie instrucțiunea **from RPLCD import CharLCD** și **import RPi.GPIO as GPIO** pentru folosirea unor comenzi speciale, pentru utilizarea LCD-ului.

Proiectul conține anumite funcții pentru display: `lcd = CharLCD(...)` inițializează interfața LCD-ului și specifică dimensiunea display-ului (ex. `Lcd = CharLCD(numbering_mode = GPIO.BOARD, pin_rs=26, pin_e=24, pins_data=[22, 18, 16, 12])`), `lcd.write_string(...)` afișează pe LCD textul dorit (ex. `lcd.write_string("TEXT")`), `lcd.clear()` curăță ecranul complet (ex. `lcd.clear()`) și `lcd.cursor_pos(...)` mută cursorul la poziția specificată (ex. `lcd.cursor_pos(2,0)`). Textul care urmează a fi scris este scris la poziția specificată de această metodă. Astfel, pentru a scrie text pe linia 2, coloana 4, vom apela `lcd.cursor_pos(2,0)`.

Rezistorul este o piesă componentă din circuitele electrice și electronice a cărei principală proprietate este rezistența electrică. Rezistorul obișnuit are două terminale, conform legii lui Ohm, curentul electric care curge prin rezistor este proporțional cu tensiunea aplicată pe terminalele rezistorului.

$$I = \frac{U}{R}$$

Potențiometrul^[5] este asemănător unei rezistențe variabile, pe care se plimbă un cursor, cu rolul de a diviza tensiunea generată de sursa de energie. Cel folosit de mine ajunge până la 10K. Potențiometrul este un rezistor cu un element mobil poziționat cu ajutorul unei manete. Elementul mobil, denumit și perie, face contact cu un material rezistiv dezizolat, în oricare dintre punctele selectate manual. Pe măsură ce contactul periei se apropie de terminalul 1 și se îndepartează de terminalul 2, rezistența spre terminalul 1 scade iar cea către terminalul 2 crește.

Dacă apropiem contactul de terminalul 2, vom obține efectul contrar. Rezistența între cele două puncte (1 și 2) este constantă indiferent de poziția contactului periei.

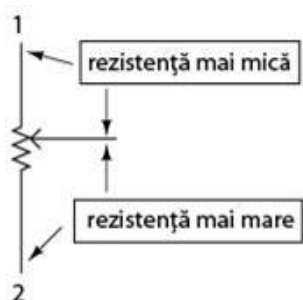


Figura 3. Funcționare potențiometru

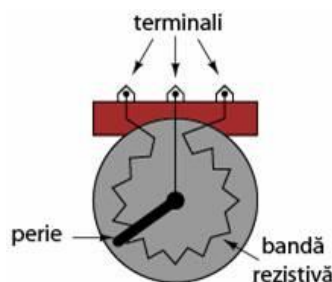


Figura 4. Potențiometru rotativ

6. INSTALARE

Pentru pornirea instalării sistemului de operare se inserează cardul în slotul microSD al plăcii și se pornește sistemul. Există două metode de configurare a plăcii Raspberry Pi: folosind utilitarele puse la dispoziție de interfața grafică a sistemului de operare Raspbian^[6] sau folosind utilitarele în linie de comandă (Terminal). Utilitarul principal de configurare se numește raspi-config și poate fi accesat atât prin intermediul interfeței grafice cât și în linie de comandă.

Comanda `sudo` permite execuția de utilitare sistem privilegiate (pot fi executate doar cu drepturi de administrare a sistemului). Chiar dacă contul de conectare inițial are drepturi depline asupra sistemului (drepturi de administrator sau root), sistemul de operare nu permite execuția unor comenzi sau utilitare importante fără a fi precedate de comanda `sudo` pentru a preveni accidentele de configurare.

Se va instala update-urile pentru pachetele software din distribuție, folosind comenzile în *Terminal*:

1. **sudo** apt-get update
2. **sudo** apt-get dist-upgrade

Se va activa serviciile de acces la distanță, pentru folosirea plăcii fără periferice proprii (tastatură, mouse și monitor):

1. **VNC** - pentru accesul în mod grafic
2. **SSH** - pentru accesul în linie de comandă

Schimbare configurare Raspberry Pi folosind *Terminalul*:

1. **sudo** raspi-config

Raspberry Pi Software Configuration Tool (raspi-config):

3 Boot Options

➔ **B1 Desktop / CLI**

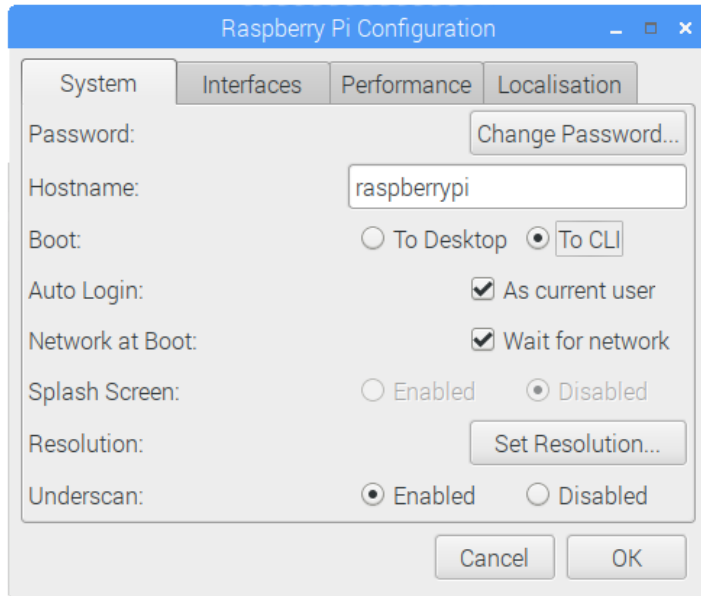


Figura 5. Configurare Raspberry Pi în mod Desktop
Boot: To Desktop
To CLI

Figura 6. Configurare Raspberry Pi din Terminal
3 Boot Options
→ B1 Desktop / CLI

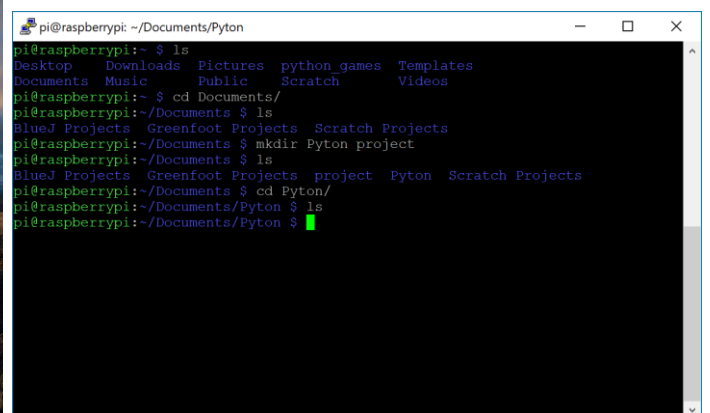
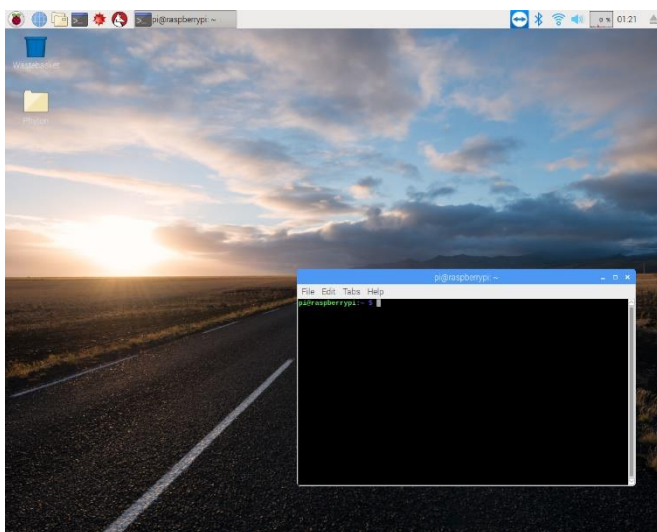
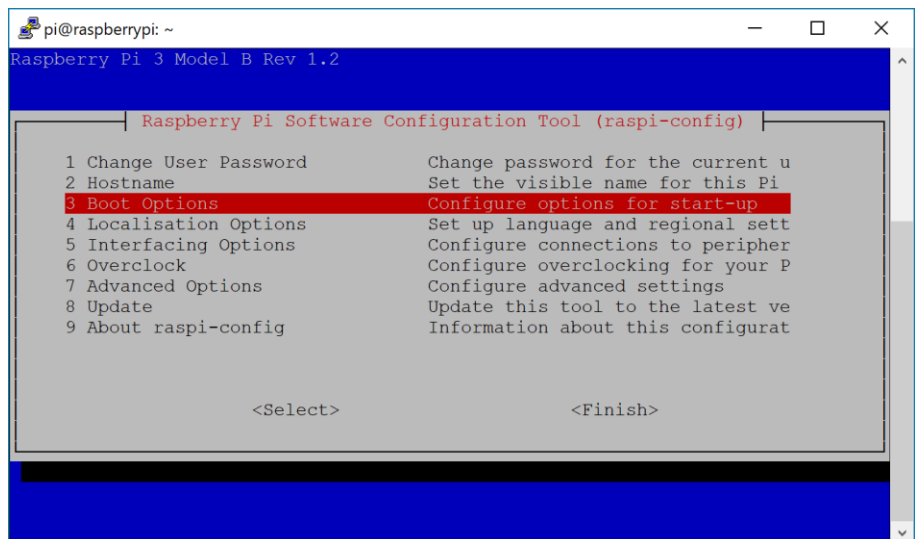


Figura 7. Modalități folosire Raspberry Pi



Desktop / CLI

Utilizarea plăcii Raspberry Pi 3 de la distanță se va face prin aplicația PuTTY^[7]. PuTTY este un program gratuit și open-source ce emulează un terminal fiind un client pentru SSH, Telnet, rlogin, și raw TCP protocol precum și client pentru serial console. Conectarea la placa Raspberry Pi prin intermediul rețelei se va face prin cunoașterea adresei IP a acesteia. Se lansează aplicația PuTTY și se va introduce IP-ul, urmând ca apoi să ne conectăm la Raspberry Pi, prin username și parolă.

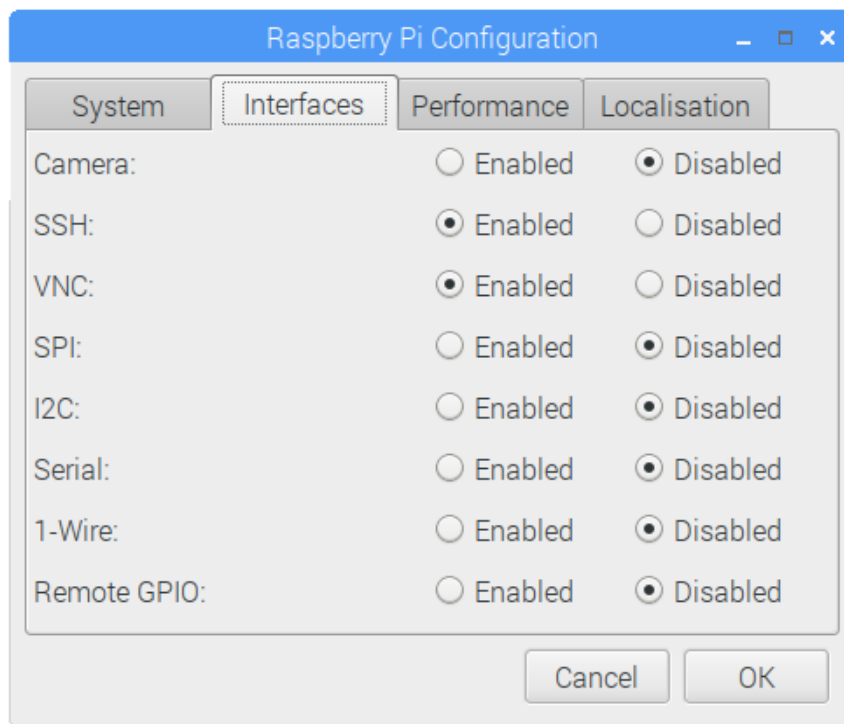


Figura 8. Raspberry Pi Configuration

SSH: Enabled

VNC: Enabled

7. CONECATARE WIFI

Pornire Raspberry Pi în mod de *Terminal*

1. **sudo iwlist wlan0 scan**

Deplasare în sus Terminal : **Shift + PageUp**

Deplasare în jos Terminal **Shift + PageDown**

2. **sudo nano /etc/wpa_supplicant/wpa_supplicant.conf**

3. Adăugare linii de cod:

```
network=
{
    ssid="(Nume Wifi)"
    psk="(Parola Wifi)"
}
```

Salvare fișier : **Ctrl + O** (Write Out)

Ieșire fișier : **Ctrl + X** (Exit)

4. **sudo ifconfig**

Căutăm la **wlan0** IP-ul



```

pi@raspberrypi: ~
File Edit Tabs Help
GNU nano 2.7.4 File: /etc/wpa_supplicant/wpa_supplicant.conf

ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev
update_config=1
country=GB

network={
    ssid="Vecinii de la 420"
    psk="parolasupersecreta"
    key_mgmt=WPA-PSK
}

network={
    ssid="SKY2490D"
    psk="QVRPBYMPR"
    key_mgmt=WPA-PSK
}

^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos
^X Exit ^R Read File ^\ Replace ^U Uncut Text ^T To Spell ^_ Go To Line

```

**Figura 9. Conectare
WiFi
sudo nano ... (2.)**

```

pi@raspberrypi: ~
File Edit Tabs Help
pi@raspberrypi:~ $ sudo ifconfig
eth0: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
    ether b8:27:eb:3b:7e:6e txqueuelen 1000 (Ethernet)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1 (Local Loopback)
    RX packets 53 bytes 26629 (26.0 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 53 bytes 26629 (26.0 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.1.112 netmask 255.255.255.0 broadcast 255.255.255.255
    inet6 fe80::a9e0:33a0:9ea7:75cb prefixlen 64 scopeid 0x20<link>
    ether b8:27:eb:6e:2b:3b txqueuelen 1000 (Ethernet)
    RX packets 70 bytes 5908 (5.7 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 466 bytes 50700 (49.5 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

pi@raspberrypi:~ $

```

**Figura 10. IP wlan0
192.168.1.112**

8. ACCESARE RASPBERRY PI DE LA DISTANȚĂ

Se deschide *PuTTY Configuration*

Host Name (or IP adress): **(IP)**

login as: **(user)**

(USER)@(IP)'s password: **(password)**

Delogare:

sudo shutdown -h now

Închidere consolă:

exit

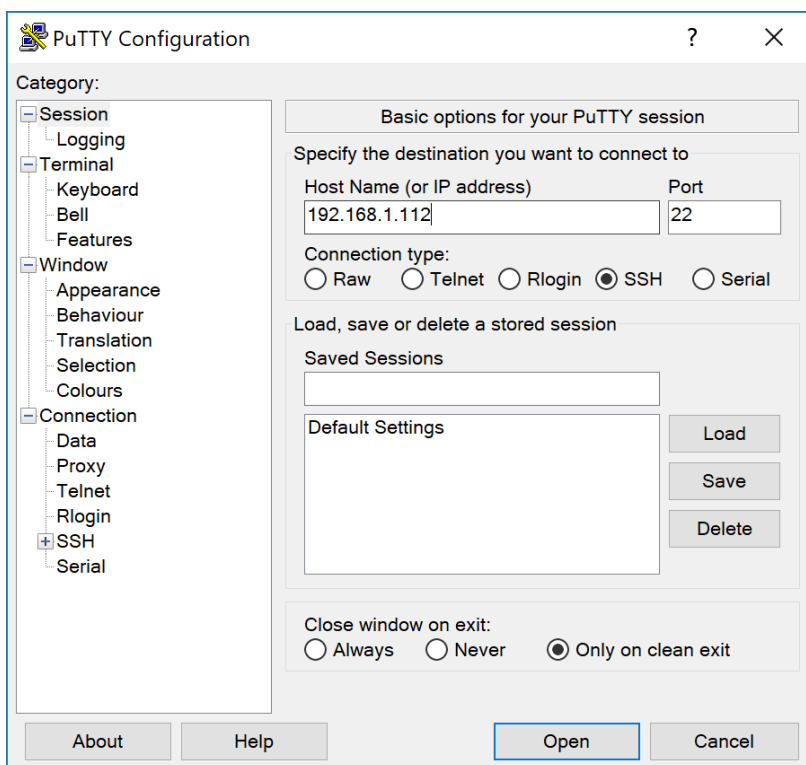


Figura 11. PuTTY Configuration

9. COMENZI

ls – Afișare conținutul directorului

ls -al – Afișarea tuturor fișierelor cu detalii despre conținut

cd (director) – Navighare director

cd .. – Deplasare la directorul precedent

clear – Curățare ecran

exit – Ieșire Terminal

Tab – Completare automată a numelui introdus

cp (fișier) (locatie) – Face o copie a fișierul în locația dată (Copy - Paste)

mv (fișier) (locatie) – Mută un fișier și îl plasează în altă locație (Cut - Paste)

mv (fișier) (nume_fișier) – Redenumire fișier

rm (fișier) – Ștergere fișier

rmdir (director) – Ștergere director

mkdir (nume_director) – Creare director nou

cat (fișier) – Vizualizare conținut fișier

nano (fișier) – Editare fișier

Cut : **Ctrl + K**

Paste : **Ctrl + U**

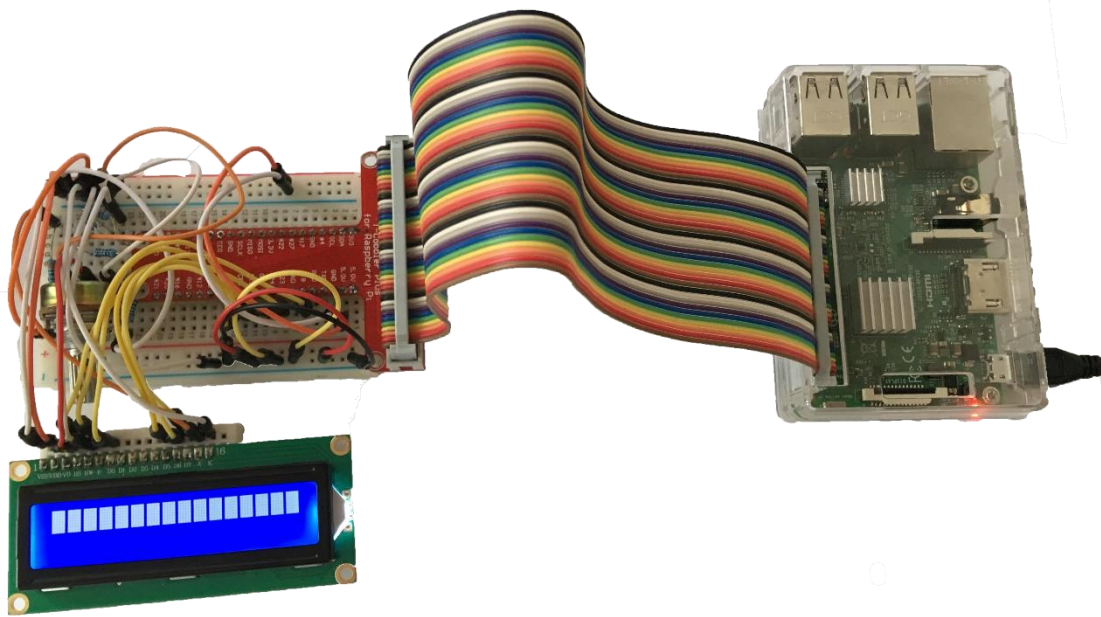
Salvare fișier : **Ctrl + O** (Write Out)

Ieșire fișier : **Ctrl + X** (Exit)

python (fișier.py) – Rulare program Python

Trimitere semnal de întrerupere la procesul actual : **Ctrl + C**

Trimitere semnal de ieșire la proces : **Ctrl + **


Figura 13. Montaj componente

11. ACCES TWITTER

Pentru a putea utiliza informațiile de pe contul de twitter, se va crea o nouă aplicație de pe site-ul oficial *Twitter Apps*^[10], urmând ca apoi să se genereze un *Consumer Key and Secret*. Acesul se va face prin *Citire*. Toate datele de mai jos se vor atașa în codul python:

Consumer Key (API Key): **5Eek4J7w2UvOzjxoloQmw3R**

Consumer Secret (API Secret): **JCQT2K44JNBYqBz3M4XUaEKum4FT7W7Ww8nCGPz8a**

Access Token: **2458501699-4mrydDyCndQJL538sw5wBNdHfEetqzhuZ0v**

Access Token Secret: **p8LV74LWQYq4mrydDyCndQJL538sw5wBNdHfEetqzhuZ0v**

Figura 14. Twitter Apps




12. SIMULARE

Se deschide *PuTTY Configuration*
 Host Name (or IP address): **192.168.1.112**
 login as: **pi**
pi@192.168.1.112's password: *****

```
$ sudo apt-get update
$ sudo apt-get dist-upgrade
    Do you want to continue? [Y/n] Y
$ sudo pip install CharLCD
$ sudo pip install RPLCD
$ sudo pip install twitter
$ sudo pip install python-twitter
$ sudo pip install HTMLParser
$ cd Documents/
$ mkdir Python-tweet
$ cd Python-tweet
$ nano lcd-twitter.py
    # COD lcd-twitter.py
Ctrl + O
    File Name to Write: lcd-twitter ENTER
Ctrl + X
$ nano rudewords.txt
    # COD rudewords.txt
Ctrl + O
    File Name to Write: rudewords ENTER
Ctrl + X
$ python lcd-twitter.py
Ctrl + C
$ sudo shutdown -h now
$ exit
```

13. COD

Pentru instalare python pe Raspberry Pi:
sudo apt-get install python-pip

Se va crea un fișier cu extensia *.py*^[1]:
nano rasppi-twitter.py – deschidere fișier

Ctrl + O – Salvare fișier

Ctrl + X – Ieșire fișier

Câteva funcții utilizate în python:

... – comentariu;

print ... – afișare Terminal

time.sleep(...) – se va introduce un număr care reprezintă timpul de înghet al programului în secunde;

Fișiere aplicație:

lcd-twitter.py – Cod python aplicație

rudewords.txt – Listă de cuvinte nepermise



```

import twitter
from RPLCD import CharLCD
import time
import HTMLParser
import RPi.GPIO as GPIO

lcd = CharLCD(numbering_mode=GPIO.BOARD, pin_rs=26, pin_e=24,
pins_data=[22, 18, 16, 12])

api=twitter.Api(consumer_key='5Eok4J7w2UvOzzjxoIoQrmw3R',
                consumer_secret='jCQT2tKt44LjNbYqBuZkNdXUbEKum4aFTJ9PBrWos8nC68PeRu',
                access_token_key='2458501699-
4rnnvybDyCndQjLS38sw5w1BNdIfiEettqvhvZ0v',
                access_token_secret='p9LVT4LWQYqkuqbsNuiJjV4wpdUMY53s1cnEUIbyVQmpB')

htmlParser = HTMLParser.HTMLParser()
lcd.clear()
lcd.write_string("Pornire program")
time.sleep(5)
lcd.clear()

try:
    while True:
        print "Cautare twitter..."
        try:
            homeTimeline=api.GetHomeTimeline(count=1)
        except:
            lcd.clear()
            lcd.write_string("Eroare se reincearca conexiunea...")
            continue
        print "Tweet gasit..."

        tweetUser = homeTimeline[0].user.screen_name
        tweetText = homeTimeline[0].text
        tweetText = htmlParser.unescape(tweetText)
        tweetText = tweetText.replace('\n', ' ')

        count = (len(tweetUser) + len(tweetText) + 2) / 32
        print count
        allText = tweetUser+": "+tweetText
        print allText

```



```

file = open('rudewords.txt', 'r')
line = file.readline()
rudewcount = 0
while line:
    line = file.readline()
    rudewcount += 1
file.close()

file = open('rudewords.txt', 'r')
line = file.readline()
rudew = [None]*(rudewcount)
i = 0
while line:
    rudew[i] = line
    line = file.readline()
    i += 1
file.close()

allow = 1
splittext = tweetText.split(" ")
textcount = len(splittext)
for i in range(0, rudewcount):
    if rudew[i][len(rudew[i])-1] == "\n":
        rudew[i] = rudew[i][:len(rudew[i])-1]

for i in range(0, textcount):
    for j in range(0, rudewcount):
        if splittext[i] == rudew[j]:
            allow = 0
            break
        else:
            continue

if allow == 1:
    for i in range(count):
        textToWrite = allText[32*i : 32*(i+1)]
        print textToWrite
        time.sleep(5)
        lcd.clear()
        lcd.write_string(textToWrite)

    time.sleep(10)

except KeyboardInterrupt:
    pass
finally:

```




```
lcd.clear()
lcd.write_string("Termianre program")
```

14. PĂRȚI DE COD

Configurare LCD:

```
import CharLCD
from RPLCD import CharLCD
import RPi.GPIO as GPIO[12]

lcd = CharLCD(numbering_mode=GPIO.BOARD, pin_rs=26, pin_e=24,
pins_data=[22, 18, 16, 12])
```

numbering_mode=GPIO.BOARD

-> Specifică pinii utilizați de Raspberry Pi

pin_rs=26, pin_e=24

-> LCD RS (26), LCD E (24)

pins_data=[22, 18, 16, 12]

-> LCD D4 (22), LCD D5 (18), LCD D6 (16), LCD D7 (12)

Excepții:

```
try:
    while True:
        print "Cautare twitter..."
        try:
            homeTimeline=api.GetHomeTimeline(count=1)
        except:
            lcd.clear()
            lcd.write_string("Eroare se reincearca conexiunea...")
            continue
        print "Tweet gasit..."
```

try:

except:

-> Codul între instrucțiunile *try* și *except* este executat. Dacă nu se produce nicio excepție, clauza *except* este omisă și executarea instrucțiunii *try* este terminată. Dacă se produce o excepție în timpul executării *try*, restul codului este omis. Dacă tipul de excepție se potrivește cu excepția numită după cuvântul cheie *except*, clauza excepțională este executată și apoi executarea continuă după instrucțiunea *try*.

api.GetHomeTimeline^[13]

-> Folosire News-feed de la Twitter

**Selectare părți Tweet:**

```
tweetUser = homeTimeline[0].user.screen_name
tweetText = homeTimeline[0].text
tweetText = htmlParser.unescape(tweetText)
tweetText = tweetText.replace('\n', ' ')
```

```
homeTimeline[0].user.screen_name
```

-> Username utilizator

```
homeTimeline[0].text
```

-> Text tweet

```
htmlParser.unescape(tweetText)
```

-> Înlocuire caractere HTML în format normal (ex. & = @)

```
tweetText.replace('\n', ' ')
```

-> Înlocuire linii noi cu spații

Organizare Tweet:

```
count = (len(tweetUser) + len(tweetText) + 2) / 32
allText = tweetUser+": "+tweetText
```

ex. Tweet:

Barack Obama @BarackObama May 29

Forever grateful for the service and sacrifice of all who fought to protect our freedoms and defend this country we love.

```
len(tweetUser)
```

@BarackObama -> Numele utilizatorului (13 caractere)

```
len(tweetText)
```

Forever grateful for the service and sacrifice of all who fought to protect our freedoms and defend this country we love.

-> Textul postat (121 caractere)

```
": "
```

-> Caractere pentru design (2 caractere)

```
len(tweetUser) + len(tweetText) + 2
```

-> Numărul total de caractere (136 caractere)

```
count = (len(tweetUser) + len(tweetText) + 2) / 32
```

-> Împărțim textul în mai multe linii pentru a afișa pe display (5 linii de text)

Twitter a dublat din luna septembrie 2016 numărul de caractere pe care le poți scrie într-un tweet de la 140 la **280** de caractere. Așadar, textul poate fi împărțit în maxim 10 linii de text. Variabila *count* va aduna număr de caractere din username-ul celui ce a postat tweet-ul, textul postat plus încă două caractere ": " pentru design.

**Fișier de cuvinte jignitoare:**

```

file = open('rudewords.txt', 'r')
    line = file.readline()
    rudewcount = 0
    while line:
        line = file.readline()
        rudewcount += 1
    file.close()

file = open('rudewords.txt', 'r')
line = file.readline()
rudew = [None]*(rudewcount)
i = 0
while line:
    rudew [i] = line
    line = file.readline()
    i += 1
file.close()

```

```

file = open('rudewords.txt', 'r')
    -> Deschidere fișier în modul de citire
line = file.readline()
    -> Citește câte o linie din fișier
rudewcount += 1
    -> Numarul de cuvinte din fișier
rudew = [None]*(rudewcount)
    -> Initializare variabilă cu numărul exact de cuvinte din fișier

```

Filtrare limbaj inadecvat:

```

allow = 1
splittext = tweetText.split(" ")
textcount = len(splittext)
for i in range(0, rudewcount):
    if rudew [i][len(rudew [i])-1] == "\n":
        rudew [i] = rudew [i][:len(rudew [i])-1]

for i in range(0, textcount):
    for j in range (0, rudewcount):
        if splittext [i] == rudew [j]:
            allow = 0
            break
        else:
            continue

```



```

splittext = tweetText.split(" ")
    -> Desparte textul tweet-ului în String-uri în momentul în care întâlnește spațiu
textcount = len(splittext)
    -> Numărul de cuvinte din splittexr
if rudew [i][len(rudew [i])-1] == "\n":
    -> Verifică dacă ultimul caracter din cuvânt este rând nou (se notează cu \n)
rudew [i] = rudew [i][:len(rudew [i])-1]
    -> Se elimină rândul nou din cuvânt
    Funcție asemănătoare cu strncpy( rudew, rudew, strlen(rudew)-1 )
if splittext [i] == rudew [j]:
    -> Verifică dacă un cuvânt din tweet face parte din lista de cuvinte jignitoare

```

Afișare Tweet pe LCD:

```

if allow == 1:
    for i in range(count):
        textToWrite = allText[32*i : 32*(i+1)]
        print textToWrite
        time.sleep(5)
        lcd.clear()
        lcd.write_string(textToWrite)

time.sleep(10)

```

ex. Afișare Tweet:

@BarackObama: Forever grateful for the service and sacrifice of all who fought to protect our freedoms and defend this country we love.

```

if allow == 1:
    -> Verifică dacă tweet-ul va fi afișat pe ecran
for i in range(count):
    -> Bucla se va repeta de exact în câte părți este secționat textul ce se afișează pe LCD
textToWrite = allText[32*i : 32*(i+1)]
    -> De fiecare dată când se accesează bucla, textToWrite va conține următoarea linie a
    textului
lcd.write_string(textToWrite)
    -> Se afișează pe display-ul LCD textul memorat din textToWrite

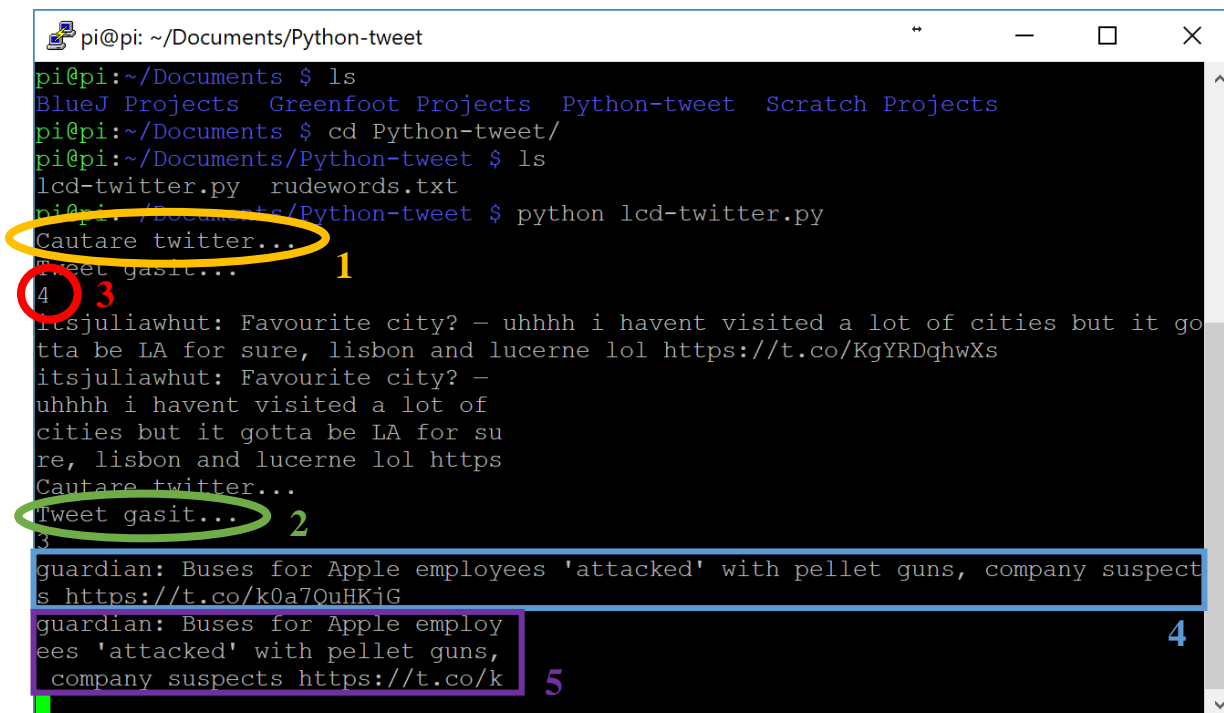
```

ex. Spațiere Tweet:

@BarackObama: Forever grateful f	#Linia 1
or the service and sacrifice of	#Linia 2
all who fought to protect our fr	#Linia 3
eedoms and defend this country w	#Linia 4
e love.	#Linia 5

15. RULARE APLICAȚIE

Rulare aplicație pe consolă:



```

pi@pi: ~/Documents/Python-tweet
pi@pi:~/Documents $ ls
BlueJ Projects  Greenfoot Projects  Python-tweet  Scratch Projects
pi@pi:~/Documents $ cd Python-tweet/
pi@pi:~/Documents/Python-tweet $ ls
lcd-twitter.py  rudewords.txt
pi@pi:~/Documents/Python-tweet $ python lcd-twitter.py
Cautare twitter...
Tweet gasit...
4
itsjuliawhut: Favourite city? - uhhhh i havent visited a lot of cities but it go
tta be LA for sure, lisbon and lucerne lol https://t.co/KgYRDqhwXs
itsjuliawhut: Favourite city? -
uhhhh i havent visited a lot of
cities but it gotta be LA for su
re, lisbon and lucerne lol https
Cautare twitter...
Tweet gasit...
3
guardian: Buses for Apple employees 'attacked' with pellet guns, company suspect
s https://t.co/k0a7QuHKiG
guardian: Buses for Apple employ
ees 'attacked' with pellet guns,
company suspects https://t.co/k

```

Figura 15. Aplicație rulată pe consolă Raspberry Pi

- 1 -> Aplicația caută cel mai recent tweet postat
- 2 -> Aplicația a găsit un tweet pe care îl prelucrează
- 3 -> Numărul de linii de text ce trebuie afișate pe display-ul LCD pe rând
- 4 -> Textul complet ce urmează a fi afișat pe display-ul LCD
- 5 -> Fiecare linie de text aranjate astfel încât să fie transmise spre display-ul LCD

Rulare aplicație pe display LCD:



Figura 16. Afișare Partea 1 LCD



Figura 17. Afișare Partea 2 LCD



Figura 18. Afișare Partea 3 LCD

Postare twitter:



Figura 19. Postare twitter

Postare twitter pe display LCD:



Figura 20. Afișare prima parte din tweet pe display LCD

16. LISTĂ COMPONENTE



Fig 21. Raspberry Pi 3 Model B

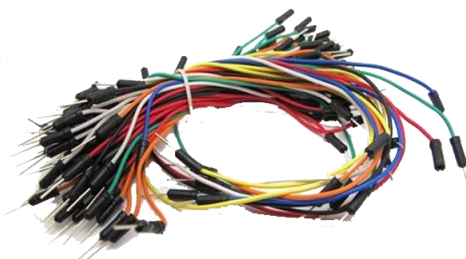


Fig 22. Set fire tată-tată

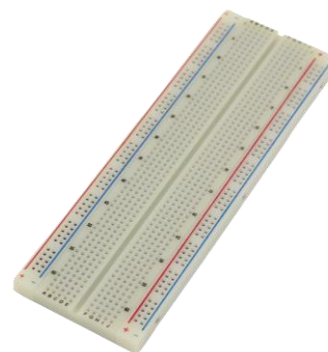


Fig 23. Breadboard



Fig 24. Display LCD 16x2



Fig 25. T - Cobbler Plus



Fig 26. Potențiomtru

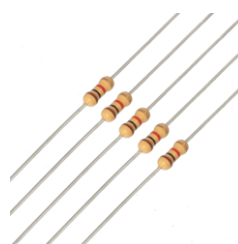


Fig 27. Rezistențe 1k Ohm



17. REFERINȚE

- [1] *Raspberry Pi 3 Model B*
<https://www.raspberrypi.org>
- [2] *Limbajul de programare Python*
<https://ro.wikipedia.org/wiki/Python>
- [3] *CharLCD 0.5.1*
<https://pypi.python.org/pypi/CharLCD/0.5.1>
- [4] *RPLCD 0.3.0*
<https://pypi.python.org/pypi/RPLCD/0.3.0>
- [5] *Potențiometr*
<http://www.creeaza.com/tehnologie/electronica-electricitate/Potentiometrul851.php>
- [6] *Instalare Raspbian*
<https://www.robofun.ro/raspberry-pi-si-componente/raspberry-pi-v3>
- [7] *PuTTY*
<http://www.putty.org>
- [8] *Twitter 1.18.0*
<https://pypi.python.org/pypi/twitter/1.18.0>
- [9] *Python-twitter 3.3*
<https://pypi.python.org/pypi/python-twitter/3.3>
- [10] *Chei Twitter*
<https://apps.twitter.com>
- [11] *Creare fișier Raspbian*
https://wiki.gentoo.org/wiki/Nano/Basics_Guide
- [12] *Python 3.6.3*
<https://docs.python.org/3/>
- [13] *Interfață Python la Twitter API*
<http://python-twitter.readthedocs.io/en/latest/twitter.html>
- [14] *Figura 1 (Raspberri Pi 3)*
<https://robofunblogblog.files.wordpress.com/2017/06/1.png?w=1312>
- [15] *Figura 2 (Limbaj de programare Python)*
<https://www.tutorialspoint.com/python>
- [16] *Figura 3 (Funcționare potențiometr)*
http://www.creeaza.com/files/electronica-electricitate/1210_poze/image004.jpg
- [17] *Figura 4 (Potențiometr rotativ)*
http://www.creeaza.com/files/electronica-electricitate/1210_poze/image006.jpg
- [18] *Figura 12 (Montaj)*
@fritzing - <http://fritzing.org>
- [19] *Figurile 5-11, 13-20*
Imagini personale
- [20] *Figurile 21 - 27 (Listă componente)*
Imaginile au fost descărcate și prelucrate