**Proiect Raspberry Pi - Twitter**

***Moldovan Cătălin***

**Grupa:** 30225

**Data:** 21.12.2017

**E-mail:** [catalin.moldovan97@gmail.com](mailto: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 conecatat la Raspberry Pi pentru afișarea newsfeed-ul 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)

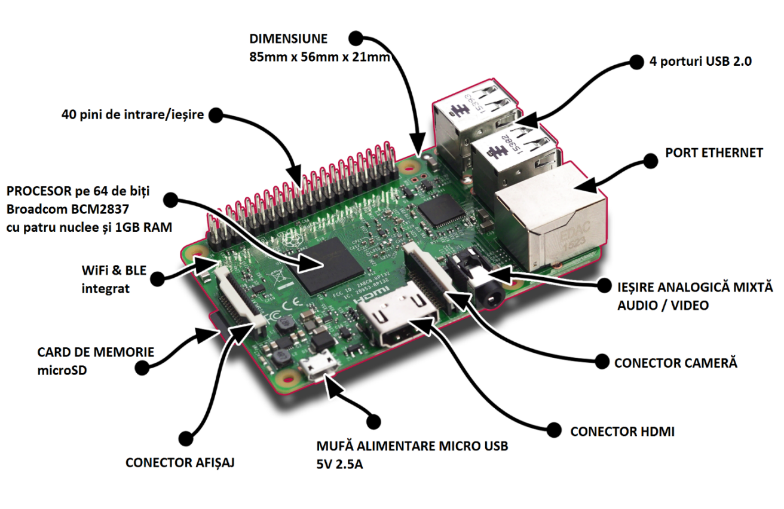
**AVANTAJE FAȚĂ DE SOLUȚIILE CUNOSCUTE:**

În cadrul conferințelor/hackathon/concurs este foarte utilă pentru afisarea 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.

1. **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](https://www.raspberrypi.org/products/raspberry-pi-3-model-b/)numeroase acoperind complet funcționalitatea unui sistem de calcul (audio, video, porturi USB, conectivitate de rețea):

* Procesor [SoC](https://en.wikipedia.org/wiki/System_on_a_chip) pe 64 de biți din familia [ARMv8-A](https://www.arm.com/products/processors/armv8-architecture.php), 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 |

1. **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 contrucții 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 bibioteci 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).

1. **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 înre 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 potențiometru de 10K pentru reglarea contrastului (sau folosirea unor rezistențe). După ce am reglat potențiometrul 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 urmeaza 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 =

*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 2.** Funcționare potențiometru **Figura 3.** Potențiometru rotativ |

1. **INSTALARE**

Pentru pornirea instalăriisistemului 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 placii fără periferice proprii (tastatură, mouse și monitor):

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

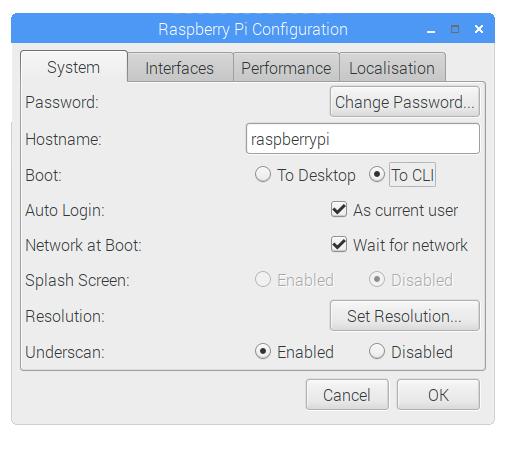
Schimbare configurare Raspberry Pi folosind *Terminalul*:

1. **sudo** raspi-config

*Raspberry Pi Software Configuration Tool (raspi-config):*

**3 Boot Options**

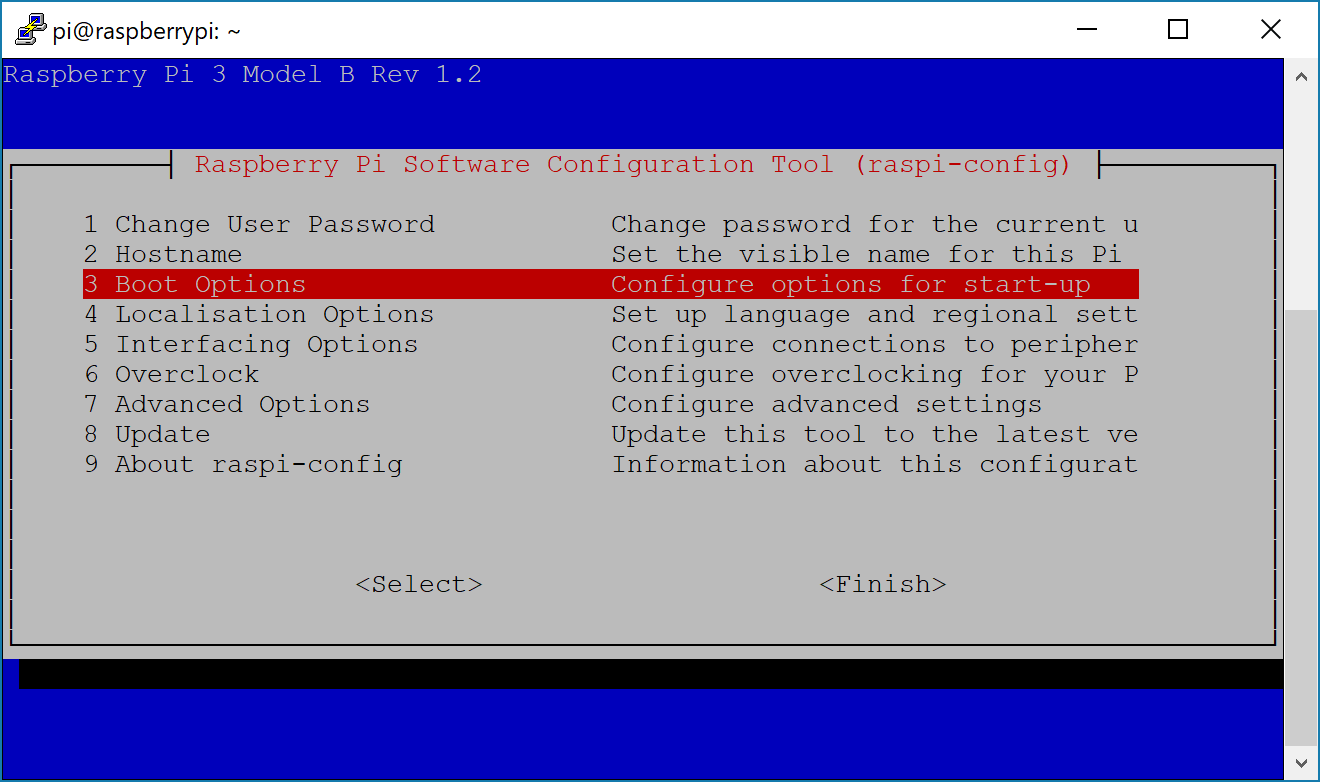
* **B1 Desktop / CLI**

****

**Figura 4.** Configurare Raspberry Pi în mod Desktop

**Boot:**To Desktop

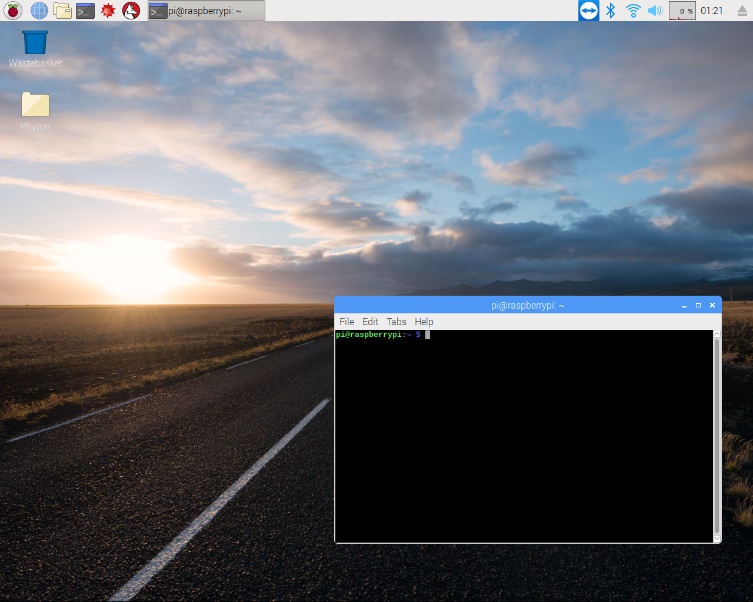
To CLI

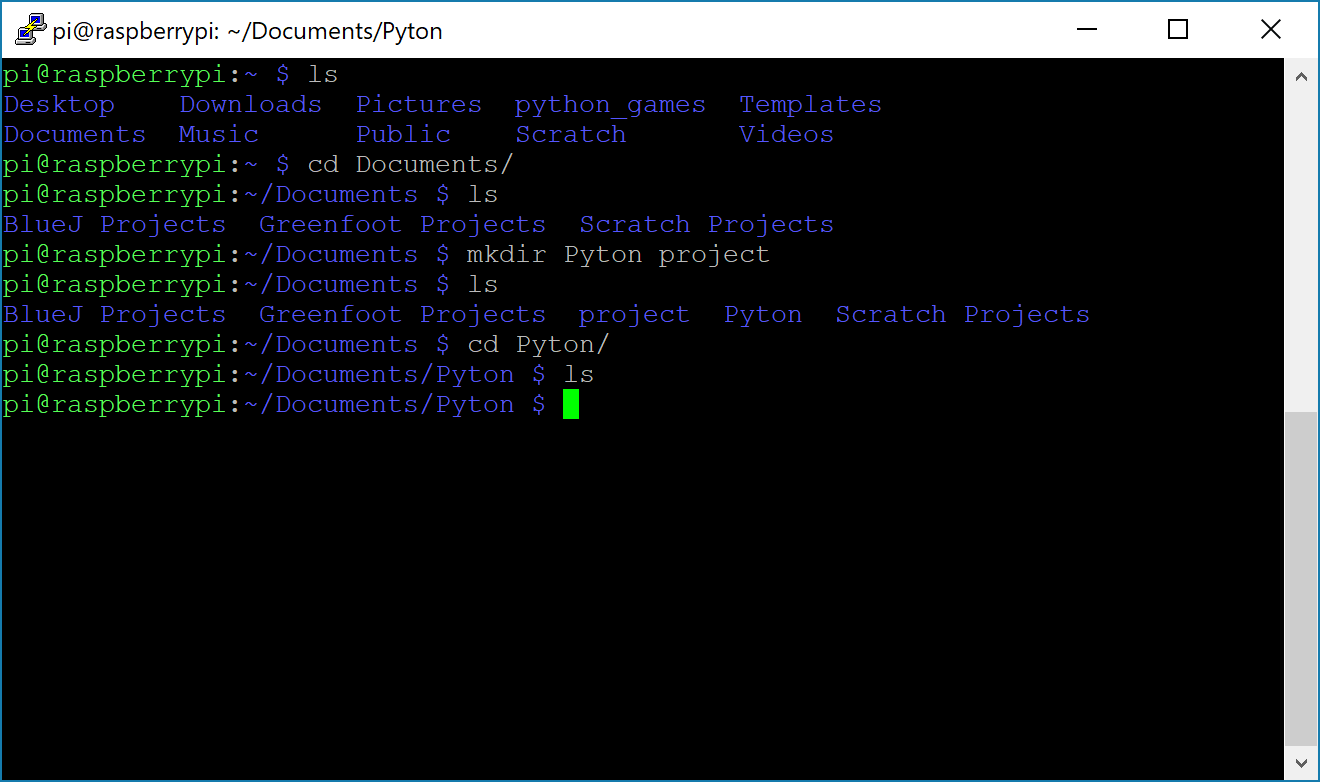
****

**Figura 5.** Configurare Raspberry Pi din Terminal

**3 Boot Options**

* **B1 Desktop / CLI**

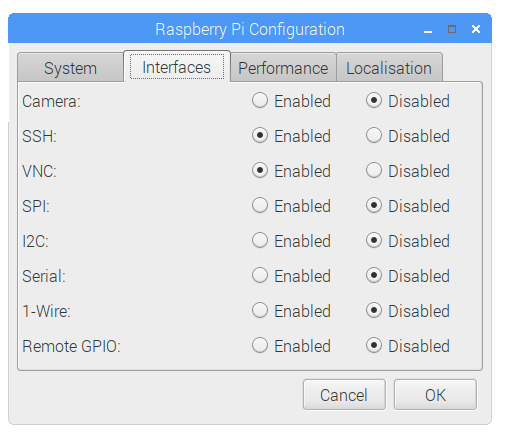
****

****

**Figura 6.** Modalități folosire Raspberry Pi

**Desktop / CLI**

Utilizarea plăcii Raspberry Pi 3 de la distanță se va face prin aplicatia 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 introdce IP-ul, urmând ca apoi să ne conectăm la Raspberry Pi, prin username și parolă.

****

**Figura 7.** Raspberry Pi Configuration

**SSH:**Enabled

**VNC:** Enabled

1. **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**

1. **sudo nano /etc/wpa\_supplicant/wpa\_supplicant.conf**
2. Adăugare linii de cod:

**network={**

**ssid="(Nume Wifi)"**

**psk="(Parola WiFi)"**

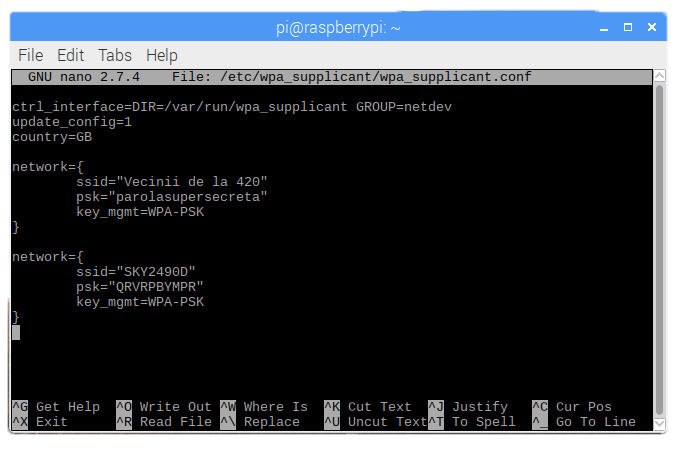
**}**

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

Ieșire fișier : **Ctrl + X** (Exit)

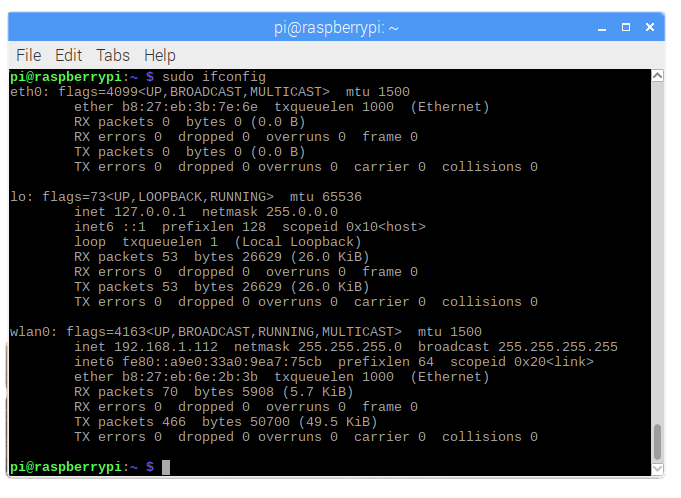
1. **sudo ifconfig**

Căutăm la **wlan0** IP-ul

****

**Figura 8.** Conectare WiFi

**sudo nano ... (2.)**

****

**Figura 9.** IP **wlan0**

192.168.1.112

1. **ACCESARE RASPBERRY PI DE LA DISTANTA**

Se deschide *PuTTY Configuration*

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

login as: **(user)**

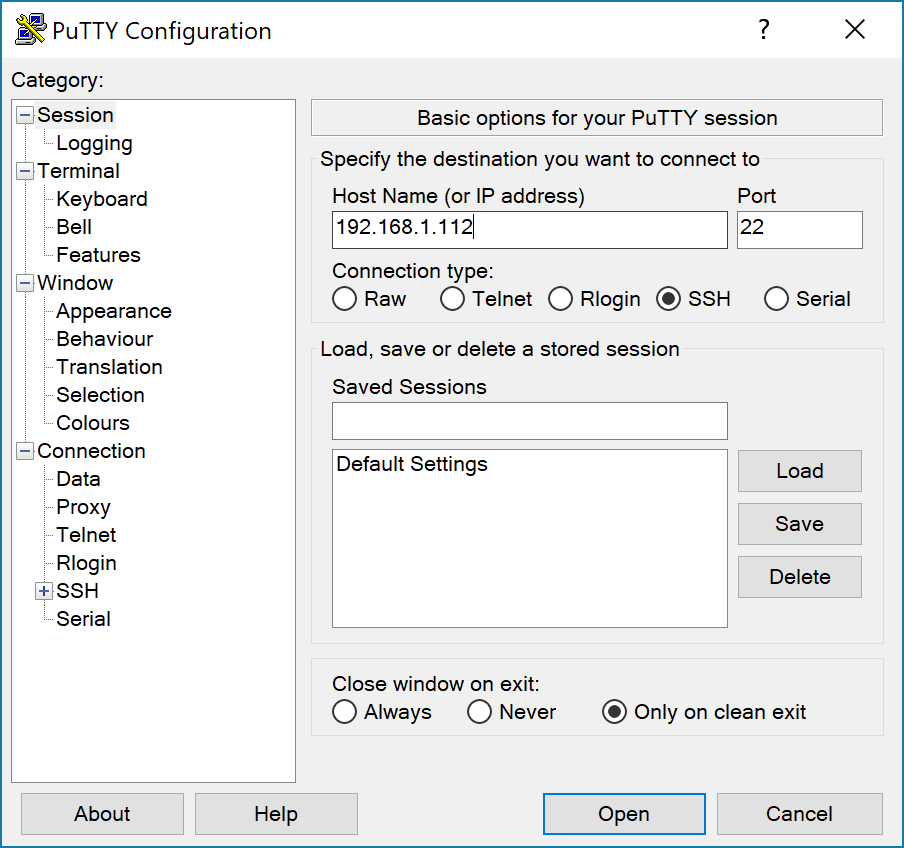
**(USER)**@**(IP)**’s password: **(password)**

Delogare:

**sudo shutdown -h now**

Închidere consolă:

**exit**

****

**Figura 10.** PuTTY Configuration

1. **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 (fisier) (locatie)** – Face o copie a fișierul în locația dată (Copy - Paste)

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

**mv (fisier) (nume\_fisier)** – Redenumire fișier

**rm (fisier)** – Ștergere fișier

**rmdir (director)** – Ștergere director

**mkdir (nume\_director)**– Creare director nou

**cat (fisier)** – Vizualizare conținut fișier

**nano (fisier)** – Editare fișier

Cut : **Ctrl + K**

Paste : **Ctrl + U**

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

Ieșire fișier : **Ctrl + X** (Exit)

**python (fisier.py)** – Rulare program Python

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

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

1. **MONTAJ**

Pentru a putea include în Python anumite librării, se vor instala următoarele librării folosind *Terminalul*:

1. **sudo** pip **install** CharLCD - **import** CharLCD
2. **sudo** pip **install** RPLCD - **from** RPLCD **import** CharLCD
3. **sudo** pip **install** twitter*[8]* - **import** twitter
4. **sudo** pip **install** python-twitter*[9]*
5. **sudo** pip **install** HTMLParser

Legăturile între LCD și placa Raspberry Pi 3 sunt prezentate mai jos, după cum urmează:

LCD (**GND**) - RaspberryPi Pin 06 (**GND**)

LCD (**VCC**) - RaspberryPi Pin 02 (**5V**)

LCD (**V0**) - 5k Ω / 10k Potențiometru - RaspberryPi Pin 06 (**GND**)

LCD (**RS**) - RaspberryPi Pin 26 (**GPIO7**)

LCD (**RW**) - RaspberryPi Pin 06 (**GND**)

LCD (**E**) - RaspberryPi Pin 24 (**GPIO8**)

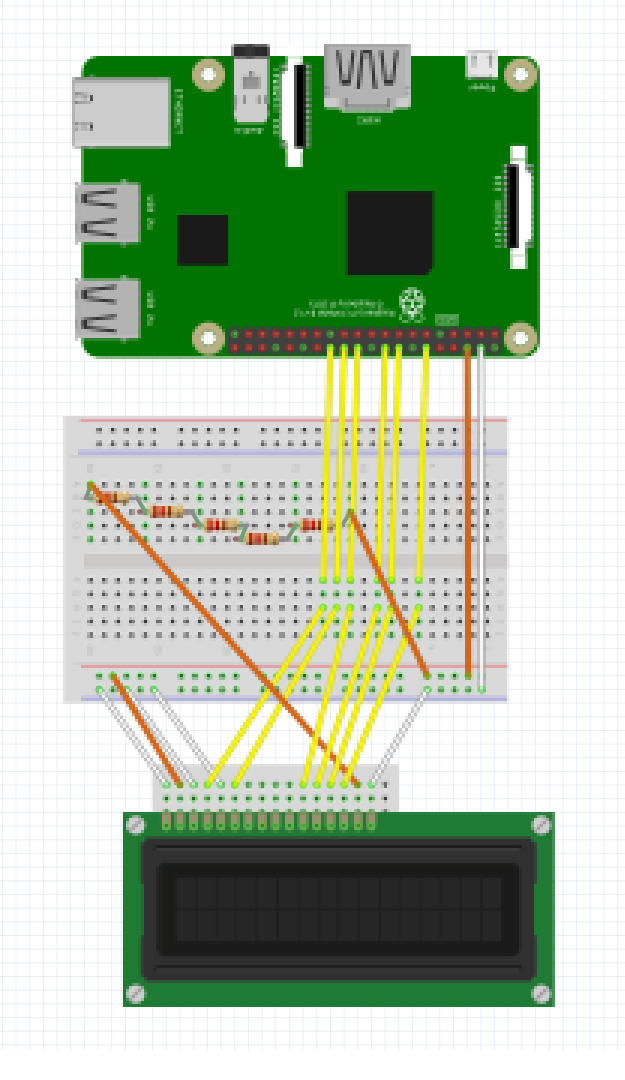
LCD (**D0-D3**) - Nu se folosesc

LCD (**D4**) - RaspberryPi Pin 22 (**GPIO25**)

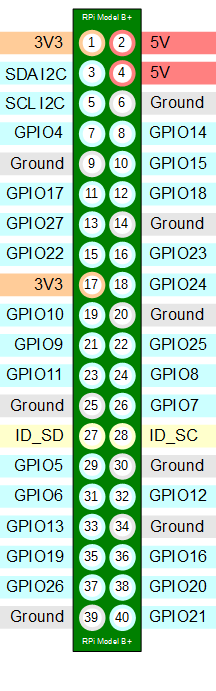
LCD (**D5**) - RaspberryPi Pin 18 (**GPIO24**)

LCD (**D6**) - RaspberryPi Pin 16 (**GPIO23**)

LCD (**D7**) - RaspberryPi Pin 12 (**GPIO18**)

 LCD (**A**) - 330 Ω - RaspberryPi Pin 02 (**5V**)

LCD (**K**) - RaspberryPi Pin 06 (**GND**)

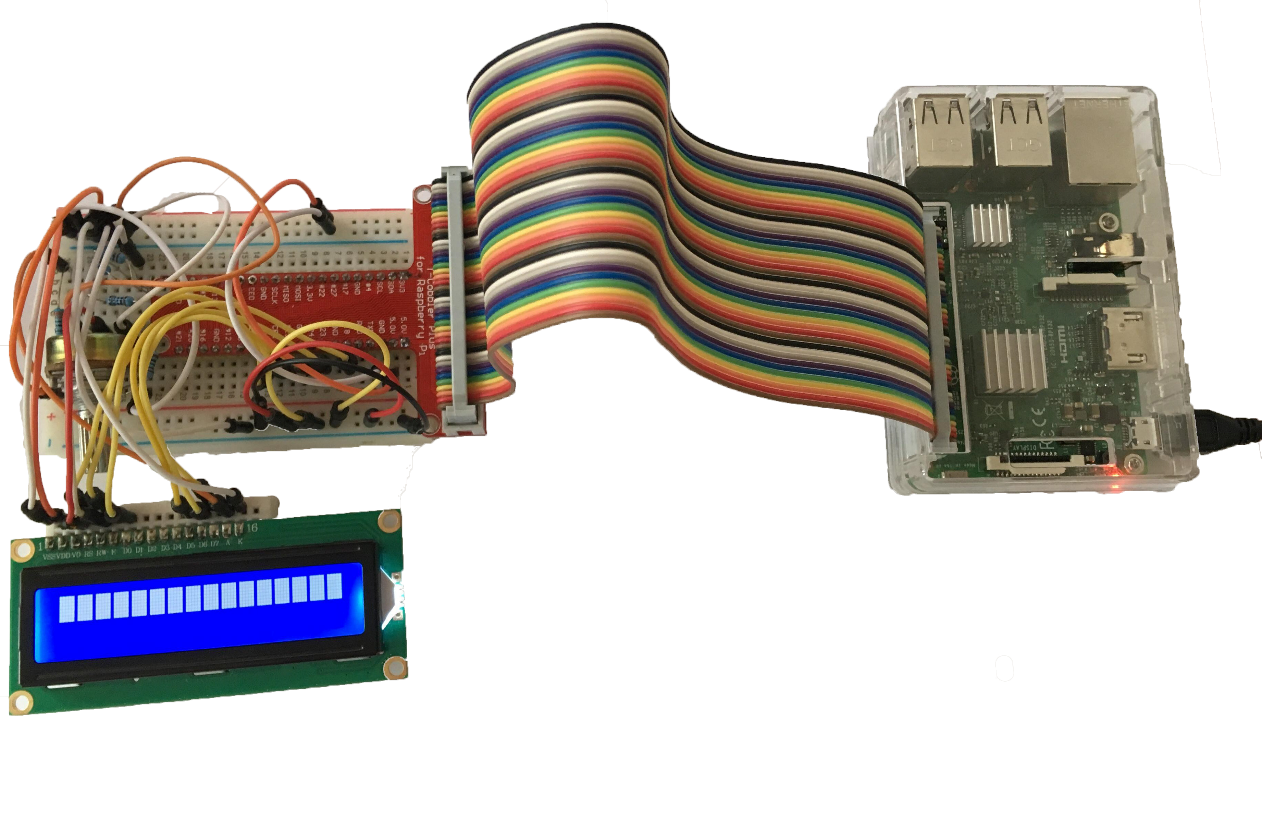


**Figura 11.** Montaj folosind programul *fritzing*

**+** Schemă Raspberry Pi 3

Rezistente: 5 x 1k Ω

LCD: 16x2



**Figura 12.** Montaj componete

1. **ACCES TWITTER**

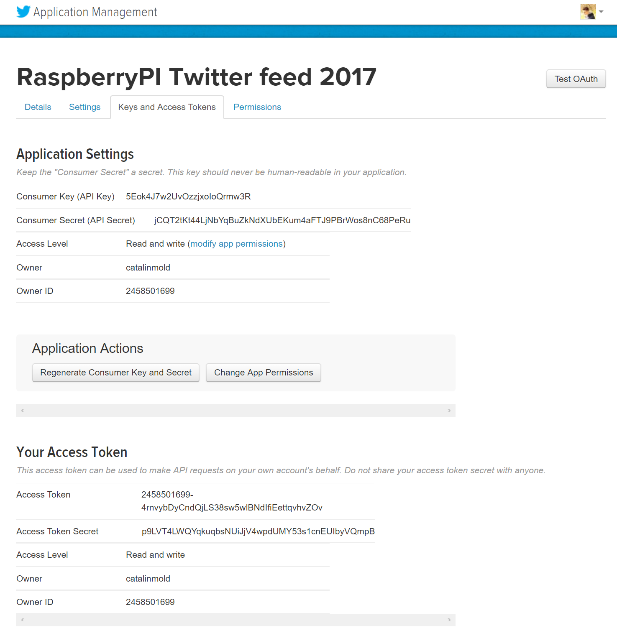
Pentru a putea utiliza informațiile de pe contul de twitter, se va crea o nouă aplicație de pe site-ul oficial *Twitter Apps[10]*, urmând ca apoi să se genereze un *Consumer Key and Secret*. Acesul se va face prin *Citire*. Toate datele de mai jos se vor atașa în codul python:

Consumer Key (API Key): 5Eok4J7w2UvOzzjxoIoQrmw3R

Consumer Secret (API Secret): jCQT2tKt44LjNbYqBuZkNdXUbEKum4aFTJ9PBrWos8nC68PeRu

Access Token: 2458501699-4rnvybDyCndQjLS38sw5wlBNdIfiEettqvhvZOv

Access Token Secret: p9LVT4LWQYqkuqbsNUiJjV4wpdUMY53s1cnEUIbyVQmpB

****

**Figura 13.** Twitter Apps

1. **SIMULARE**

Se deschide *PuTTY Configuration*

Host Name (or IP adress): **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

**Ctrl + O**

File Name to Write: lcd-twitter **ENTER**

**Ctrl + X**

$ **python lcd-twitter.py**

**Ctrl + C**

**$ sudo shutdown -h now**

**$ exit**

1. **COD**

Pentru instalare python pe Raspberry Pi:

**sudo** apt-get **install** python-pip

Se va crea un fișier cu extensia *.py[11]*:

**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;

|  |
| --- |
|  |
|  |
|  | import CharLCD  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='2458501699-4rnvybDyCndQjLS38sw5wlBNdIfiEettqvhvZOv', |
|  | 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 |
|  |  |
|  | 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") |

1. **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 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. &amp = @)

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](https://twitter.com/BarackObama) [May 29](https://twitter.com/BarackObama/status/869178832082874369)

More

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, texul poate fi împărțit în maxim 10 linii de text. Variabila *count* va aduna numar de caractere din username-ul celui ce a postat tweet-ul, textul postat plus încă două caractere ": " pentru design.

***Afișare Tweet pe LCD:***

|  |
| --- |
| 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](https://twitter.com/BarackObama): Forever grateful for the service and sacrifice of all who fought to protect our freedoms and defend this country we love.

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

rever grateful for the service a

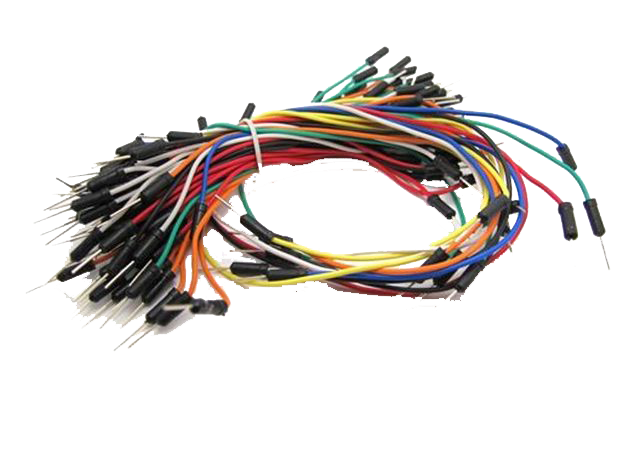
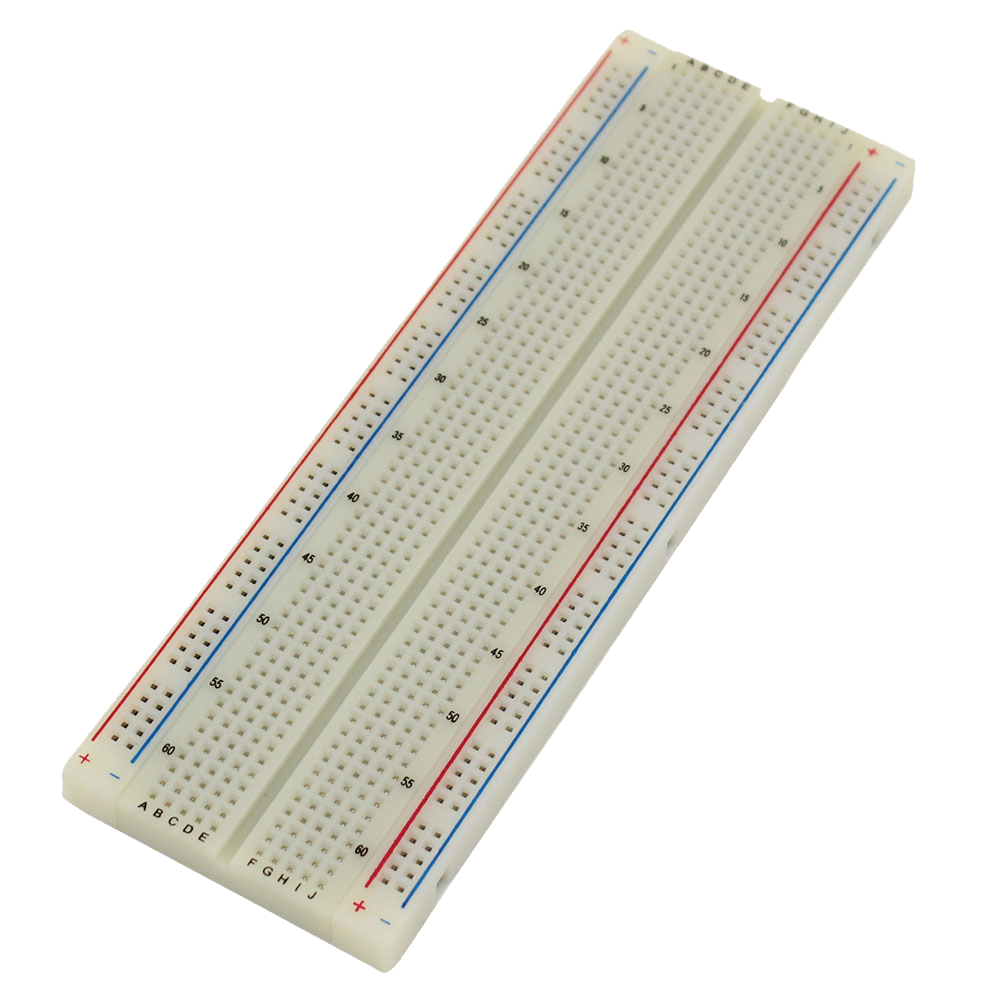
nd sacrifice of all who fought t

o protect our freedoms and defen

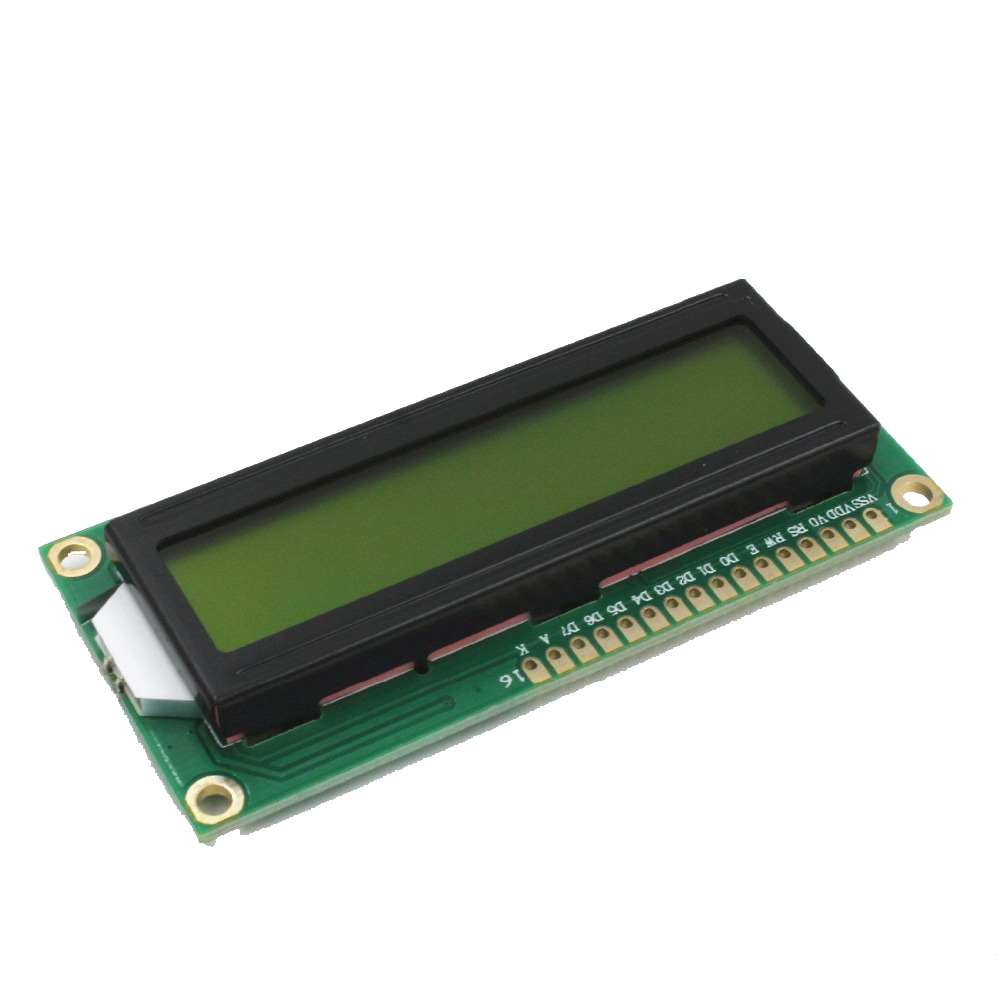
d this country we love.

1. **LISTĂ COMPONENTE**

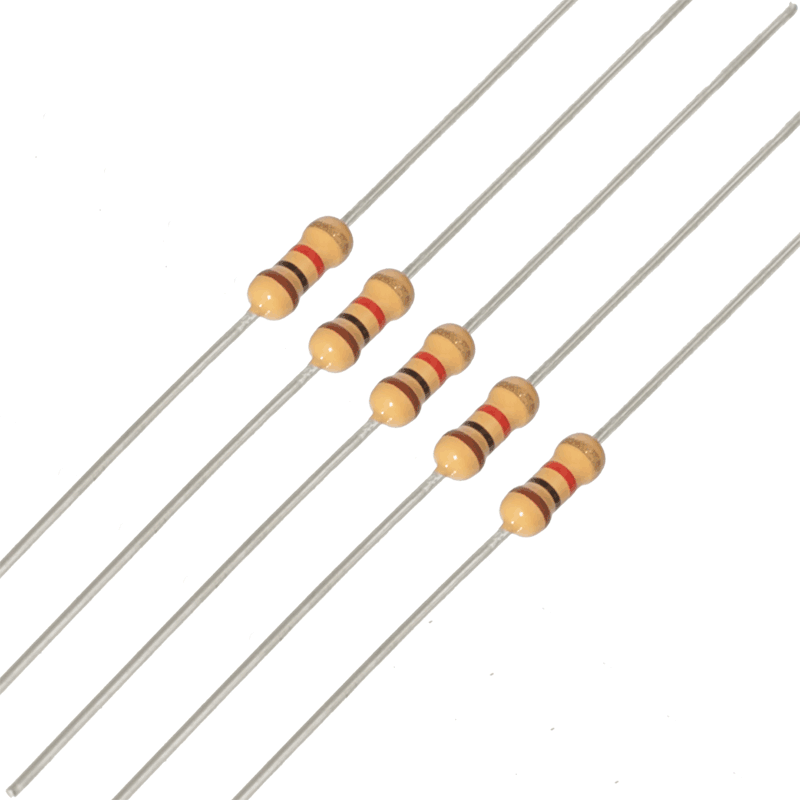
****

****

**Fig 14.** Raspberry Pi 3 Model B **Fig 15.** Set fire tată-tată **Fig 16.** Breadboard

****

**Fig 17.** Display LCD 16x2 **Fig 18.** T - Cobbler Plus

****

**Fig 19.** Potențiometru **Fig 20.** Rezistențe 1k Ohm

1. **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] *Potențiometru*

<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* (Funcționare potențiometru)

<http://www.creeaza.com/files/electronica-electricitate/1210_poze/image004.jpg>

[16] *Figura 3* (Potențiometru rotativ)

<http://www.creeaza.com/files/electronica-electricitate/1210_poze/image006.jpg>

[17] *Figura 11* (Montaj)

@frtizing - <http://fritzing.org>

[18] *Figurile 4-10, 12-13*

Imagini personale

[19] *Figurile 14 - 20* (Listă componente)

Imaginile au fost descărcate și prelucrate