

Proiect Raspberry Pi - Twitter

Moldovan Cătălin

Grupa: 30225

Data: 23.11.2017

E-mail: catalin.moldovan97@gmail.com

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 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);

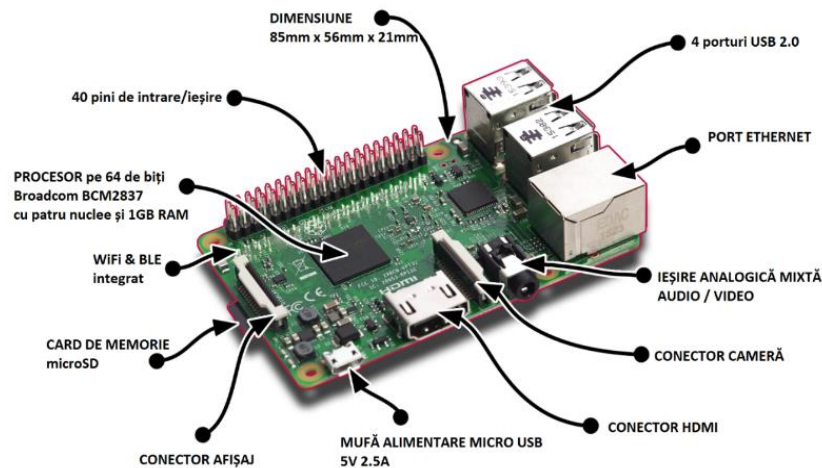


Figura 1. Plăcuța Raspberry Pi 3 Model B

3. SOFTWARE

Python 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).

4. 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 *RPLC*^[4] (`sudo pip3 install RPLCD`). La începutul fișierului python se va scrie instrucțiunea **from RPLCD import CharLCD** 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(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 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.

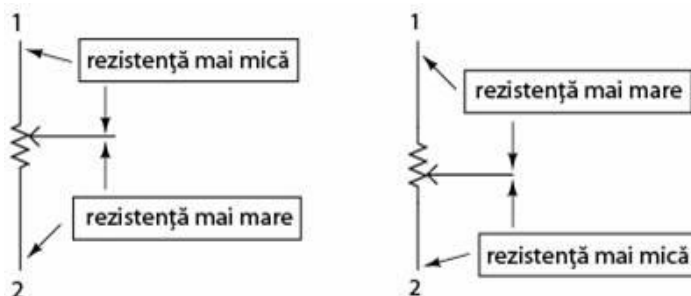


Figura 2. Funcționare potențiomtru

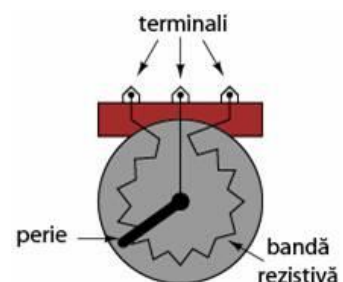


Figura 3. Potențiomtru rotativ

5. 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 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

Utilizarea plăcii Raspberry Pi 3 de la distanță se va face prin aplicația PuTTY. 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.

6. MONTAJ

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

1. **`sudo pip3 install CharLCD - import CharLCD`**
2. **`sudo pip3 install RPLCD - from RPLCD import CharLCD`**
3. **`sudo pip3 install twitter - import twitter`**
4. **`sudo pip3 install python-twitter`**

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 Potentiometru - 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)

3V3	1	2	5V
SDA I2C	3	4	5V
SCL I2C	5	6	Ground
GPI04	7	8	GPI014
Ground	9	10	GPI015
GPI017	11	12	GPI018
GPI027	13	14	Ground
GPI022	15	16	GPI023
3V3	17	18	GPI024
GPI010	19	20	Ground
GPI09	21	22	GPI025
GPI011	23	24	GPI08
Ground	25	26	GPI07
ID_SD	27	28	ID_SC
GPI05	29	30	Ground
GPI06	31	32	GPI012
GPI013	33	34	Ground
GPI019	35	36	GPI016
GPI026	37	38	GPI020
Ground	39	40	GPI021

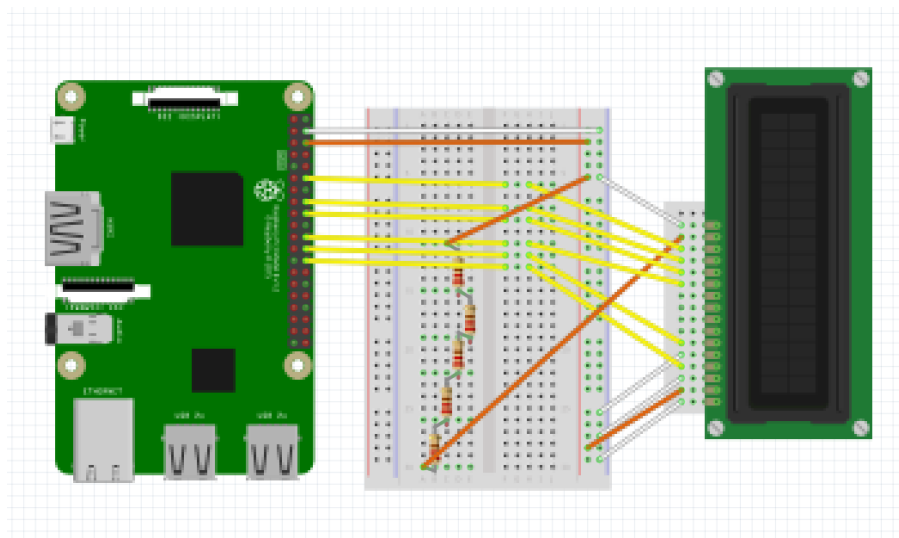


Figura 4. Montaj folosind programul *fritzing*
 + Schemă Raspberry Pi 3
 Rezistente: 5 x 1k Ω
 LCD: 16x2

7. 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*, 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): **5E6e6U7wZUvQmjp6bQmwwSR**

Consumer Secret (API Secret): **3CQI2EK44LJ6YqBwZkN4KU6EKmm4mFT3P2Ww6mC8P6m**

Access Token: **2458301693-4mrytDyCndQL832w5wEN42E6m4q6m2Dv**

Access Token Secret: **pSLVT4LWQYq6m6mNULJV4wp6UMY33m1mEULhyVQmpB**

8. COD

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

Se va crea un fișier cu extensia *.py*:

nano raspbi-twitter.py – deschidere fișier

Ctrl + O – Salvare fișier

Ctrl + X – Ieșire fișier

Câteva funcții utilizate în python3:

... – comentariu;

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

Cod python:

```
import twitter
```

```
import CharLCD
```

```
from RPLCD import CharLCD
```

```
lcd = CharLCD(pin_rs=26, pin_e=24, pins_data=[22, 18, 16, 12]) # configurare LCD
```

```
api=twitter.Api(consumer_key='586h4J7w2UvOmfjmbQzmm5R',  
                consumer_secret='jCQY2dX34EjB6YqBwZ3N4XU6EKmm4eFT3SP2zWw6mC8P62m',  
                access_token_key='2458501699-4mwy6DyCndQJL38w5wIBN42E8ettq4wZ0v',  
                access_token_secret='y6LVT4LWQYq4m6fmbNULjV4wp6UMY33u1m2U6hyVQmp8')
```

```
lcd.clear()
```

```
lcd.write_string('Pornire program')
```

```
time.sleep(5)
```

9. LISTĂ COMPONENTE



Fig 5. Raspberry Pi 3 Model B

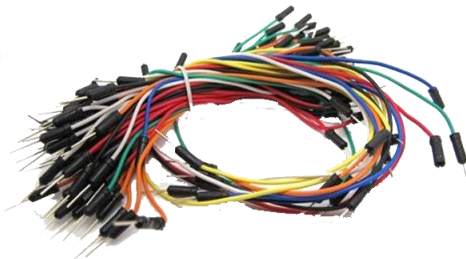


Fig 6. Set fire tată-tată

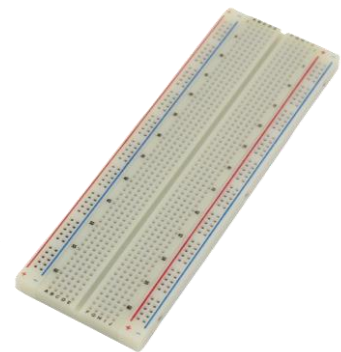


Fig 7. Breadboard



Fig 8. Display LCD 16x2



Fig 9. T - Cobbler Plus



Fig 10. Potențiomtru

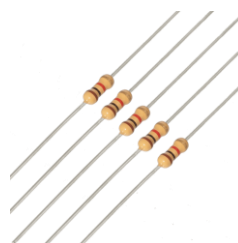


Fig 11. Rezistențe 1k Ohm

10. REFERINȚE

- [1] *Raspberry Pi 3 Model B*
<https://www.raspberrypi.org>
- [2] *Limbaajul 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] *Potențiomtru*
<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] *Figura 1 (Raspberri Pi 3)*
<https://robofunblogblog.files.wordpress.com/2017/06/1.png?w=1312>
- [14] *Figura 2 (Funcționare potențiomtru)*
http://www.creeaza.com/files/electronica-electricitate/1210_poze/image004.jpg
- [15] *Figura 3 (Potențiomtru rotativ)*
http://www.creeaza.com/files/electronica-electricitate/1210_poze/image006.jpg
- [16] *Figura 4 (Montaj)*
[@fritzing - http://fritzing.org](http://fritzing.org)
- [17] *Figurile 5 - 11 (Listă componente)*
Imaginile au fost descărcate și prelucrate