

Textul si imaginile din acest document sunt licentiate

Attribution-NonCommercial-NoDerivs
CC BY-NC-ND



Codul sursa din acest document este licentiat

Public-Domain

Esti liber sa distribui acest document prin orice mijloace consideri (email, publicare pe website / blog, printare, sau orice alt mijloc), atat timp cat nu aduci nici un fel de modificari acestuia. Codul sursa din acest document poate fi utilizat in orice fel de scop, de natura comerciala sau nu, fara nici un fel de limitari.

Raspberry PI si ULN2803 shield

Shield-ul ULN2803 se conecteaza impreuna cu placa Raspberry PI iar rolul lui este sa iti permita controlul unor sarcini de putere mai mare, sarcini pe care pinii GPIO nu le poate suporta. Spre exemplu, poti controla motoare de putere mica (consumul sa nu depaseasca 0.5A/motor) sau poti controla bobinele unor relee.

Shield-ul iti permite sa comanzi ON/OFF pana la 8 sarcini independent. Consumul general la nivelul placii este de 500mA (toti pinii insumati).

Cum se utilizeaza shield-ul ?

Pentru a controla o sarcina folosind shield-ul ULN2803, va trebui sa conectezi sarcina intre unul dintre pinii marcati cu 5V (ai la dispozitie 8 pini) si unul dintre pinii de deasupra (marcat cu „G“ urmat de un numar). Pinul marcat cu „G“ functioneaza ca un intrerupator conectat intre sarcina ta si GND. Atunci cand din codul sursa activezi pinul GPIO corespunzator (vezi codul sursa de mai jos), intrerupatorul se inchide si permite trecerea curentului, inchizand circuitul si alimentand sarcina conectata. Atunci cand pinul GPIO este in LOW, intrerupatorul este deschis si curentul nu trece prin sarcina.

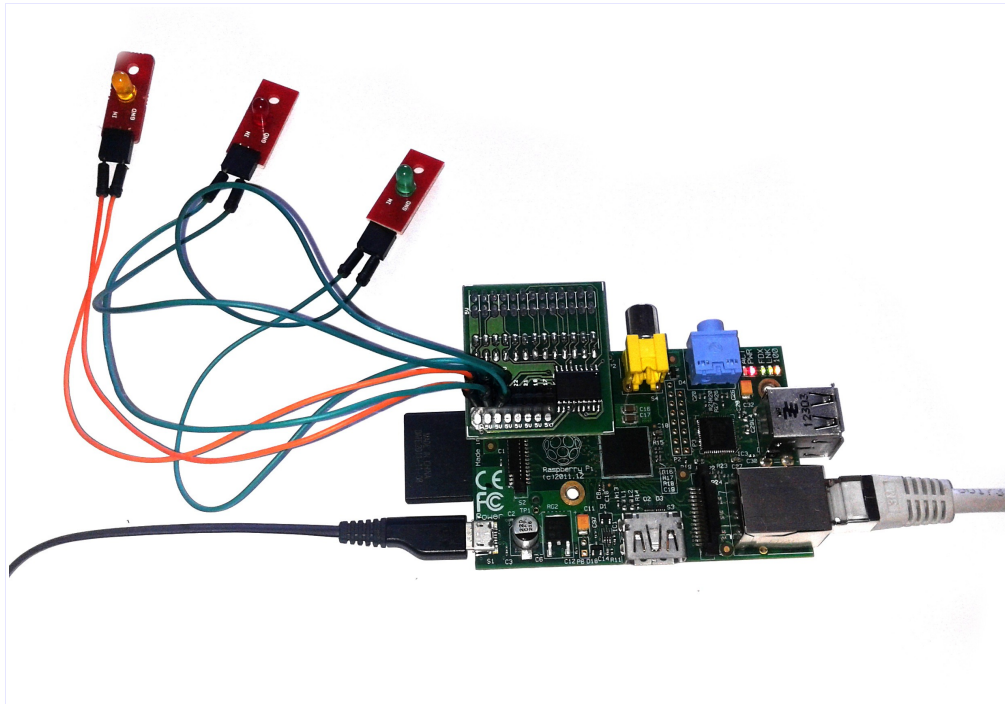
Shield-ul expune in acelasi timp si toti pinii GPIO ai placii Raspberry PI, in cazul in care vrei sa conectezi si alte componente la placa (senzori, led-uri sau alte placi de extensie).

Ce vei face in tutorialul de fata:

1. Conecteaza shield-ul la placa Raspberry PI.
2. Ca sa vedem cum functioneaza shield-ul si programul de mai jos, conecteaza 3 consumatori la pinii shield-ului. Poti conecta motoare, relee, led-uri de putere. In acest tutorial s-au conectat 3 led-uri brick la pinii urmatoari:

Led Galben Pin IN	Shield ULN2803 5V
Led Rosu Pin IN	Shield ULN2803 5V
Led Verde Pin IN	Shield ULN2803 5V
Led Galben Pin GND	Shield ULN2803 G04
Led Rosu Pin GND	Shield ULN2803 G25
Led Verde Pin GND	Shield ULN2803 G24

Vei obtine urmatoarea imagine:



3. Conecteaza placa Raspberry PI la retea si logheaza-te prin SSH.
4. Creeaza un director nou si un fisier cu urmatoarele 2 comenzi:

```
mkdir shield_uln2803
sudo nano shield.py
```

5. Copiaza codul sursa de mai jos. Salveaza-l cu CTRL X si Y.

```
GNU nano 2.2.6 File: shield.py
#!/usr/bin/env python
import sys
import time
import os
import RPi.GPIO as GPIO

GPIO.setwarnings(False)
GPIO.setmode(GPIO.BCM)
GPIO.setup(int(sys.argv[1]), GPIO.OUT)
GPIO.output(int(sys.argv[1]), int(sys.argv[2]))
print "Starea pinului " + sys.argv[1] + " a fost setata pe " + sys.argv[2] + " logic"
```

6. Executa programul cu urmatoarele comenzi:

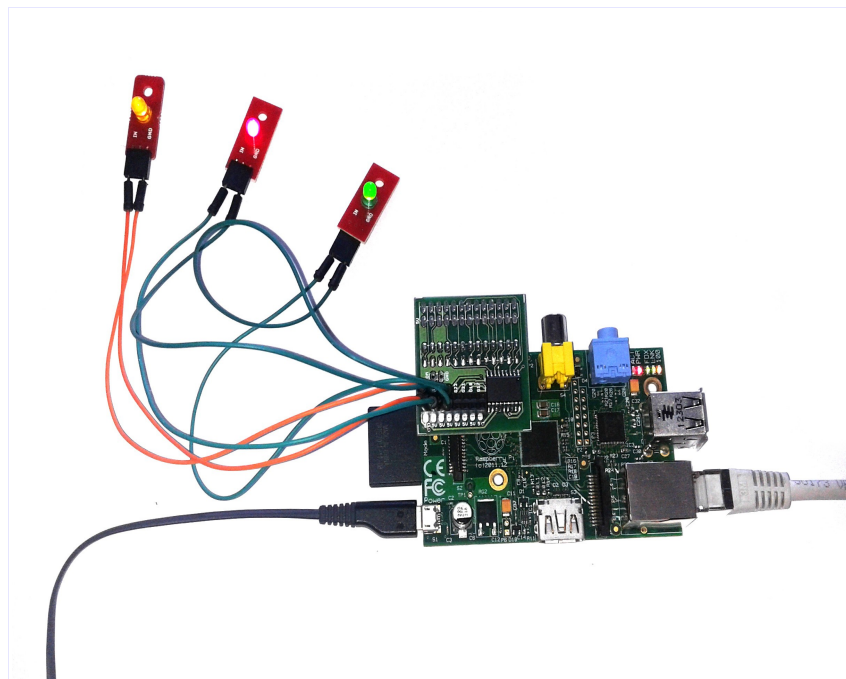
```
sudo python shield.py 4 1
sudo python shield.py 25 1
sudo python shield.py 24 1

sudo python shield.py 4 0
sudo python shield.py 25 0
sudo python shield.py 24 0
```

```
pi@raspberrypi ~/shield_uln2803 $ sudo python shield.py 4 1
Starea pinului 4 a fost setata pe 1 logic
pi@raspberrypi ~/shield_uln2803 $ sudo python shield.py 25 1
Starea pinului 25 a fost setata pe 1 logic
pi@raspberrypi ~/shield_uln2803 $ sudo python shield.py 24 1
Starea pinului 24 a fost setata pe 1 logic
pi@raspberrypi ~/shield_uln2803 $ sudo python shield.py 4 0
Starea pinului 4 a fost setata pe 0 logic
pi@raspberrypi ~/shield_uln2803 $ sudo python shield.py 25 0
Starea pinului 25 a fost setata pe 0 logic
pi@raspberrypi ~/shield_uln2803 $ sudo python shield.py 24 0
Starea pinului 24 a fost setata pe 0 logic
```

Cum functioneaza programul ?

Atunci cand apelezi programul, trebuie sa ii dai 2 argumente. Primul argument este pinul GPIO care poate fi 4, 24 sau 25 iar cel de-al doilea argument este starea pinului, care poate fi „1” sau „0”, adica pentru cele 3 led-uri „1” inseamna aprins iar „0” inseamna stins.



```
#!/usr/bin/env python
import sys
import time
import os
import RPi.GPIO as GPIO

GPIO.setwarnings(False)
GPIO.setmode(GPIO.BCM)
GPIO.setup(int(sys.argv[1]), GPIO.OUT)
GPIO.output(int(sys.argv[1]), int(sys.argv[2]))
print "Starea pinului " + sys.argv[1] + " a fost setata pe " +
sys.argv[2] + " logic"
```