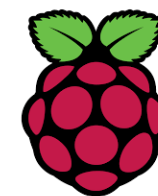




# Μικροελεγκτές Raspberry Pi

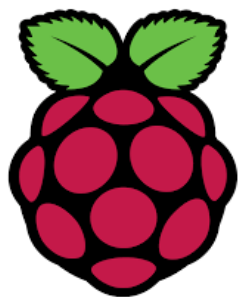
Ι. Έλληνας



Οκτώβριος 2018

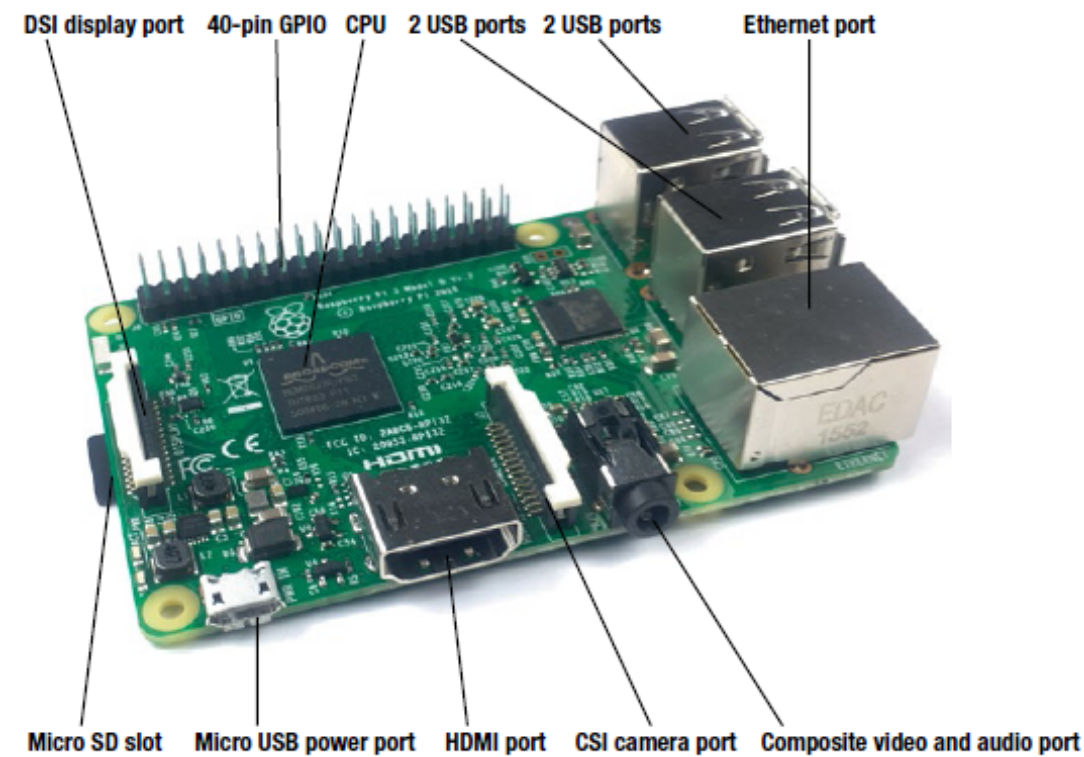
# Αναφορές

1. Σημειώσεις μαθήματος
2. Simon Monk, “Raspberry Pi Cookbook”, O’Reilly, 2016
3. Tim Cox, “Raspberry Pi for Python Programmers Cookbook, 2<sup>nd</sup> Edition 2016, Packt



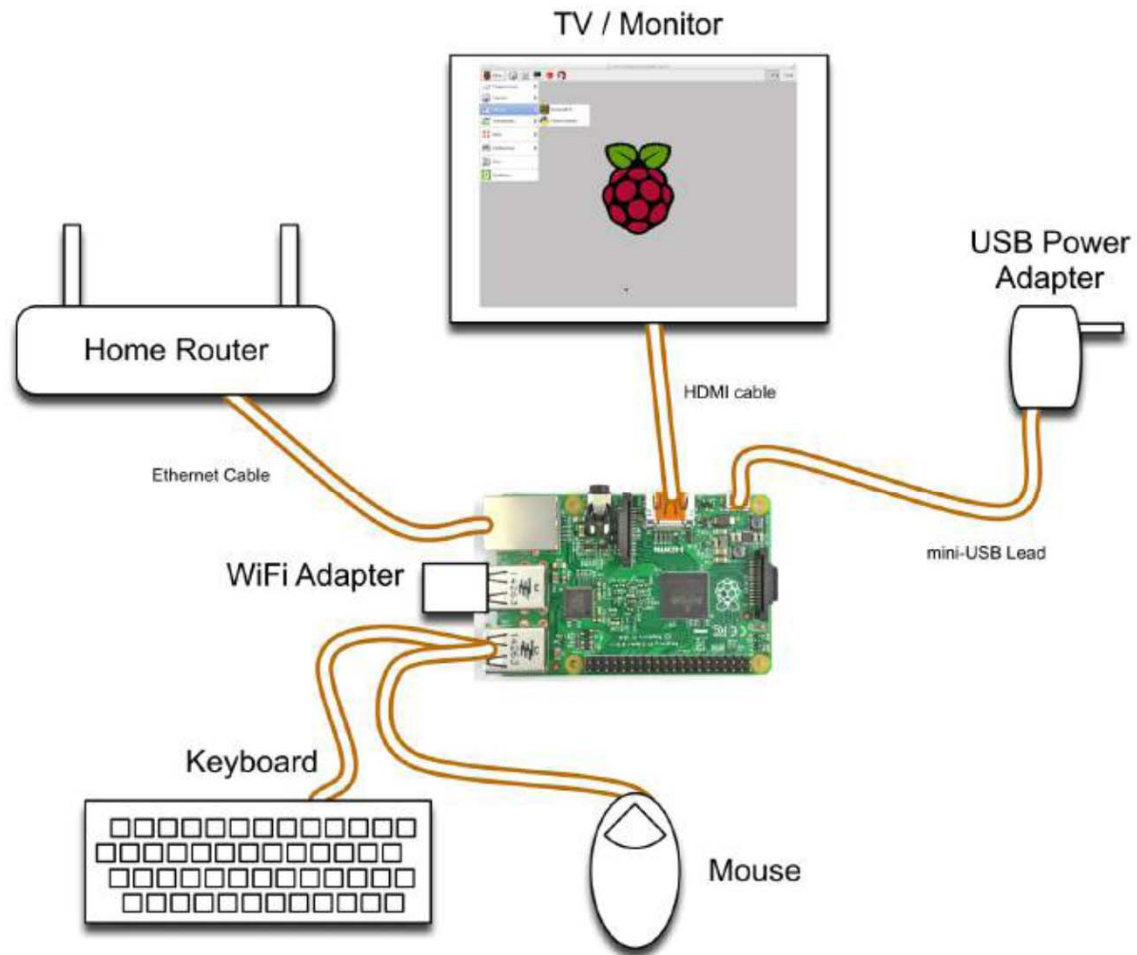
Με μια ματιά

# Raspberry Pi 3 model B



- Τετραπύρηνη CPU, 1.2 GHz
- 40-pin GPIO (General Purpose I/O)
- DSI (Display Serial Interface) για σύνδεση οθόνης (touch screen)
- Micro SD slot για τοποθέτηση λειτουργικού συστήματος
- Micro USB power port για τροφοδοσία (5V, 2.5A)
- HDMI port για σύνδεση οθόνης ή TV
- CSI (Camera Serial Interface) για σύνδεση κάμερας
- Composite video and audio output port για σύνδεση σε οθόνη και μεγάφωνα
- Τέσσερα USB ports για σύνδεση περιφερειακών (πληκτρολόγιο, ποντίκι, εκτυπωτή)
- Ethernet port 10/100 για σύνδεση με δίκτυο
- 1 GB RAM
- 802.11n Wi-Fi and Bluetooth 4.1

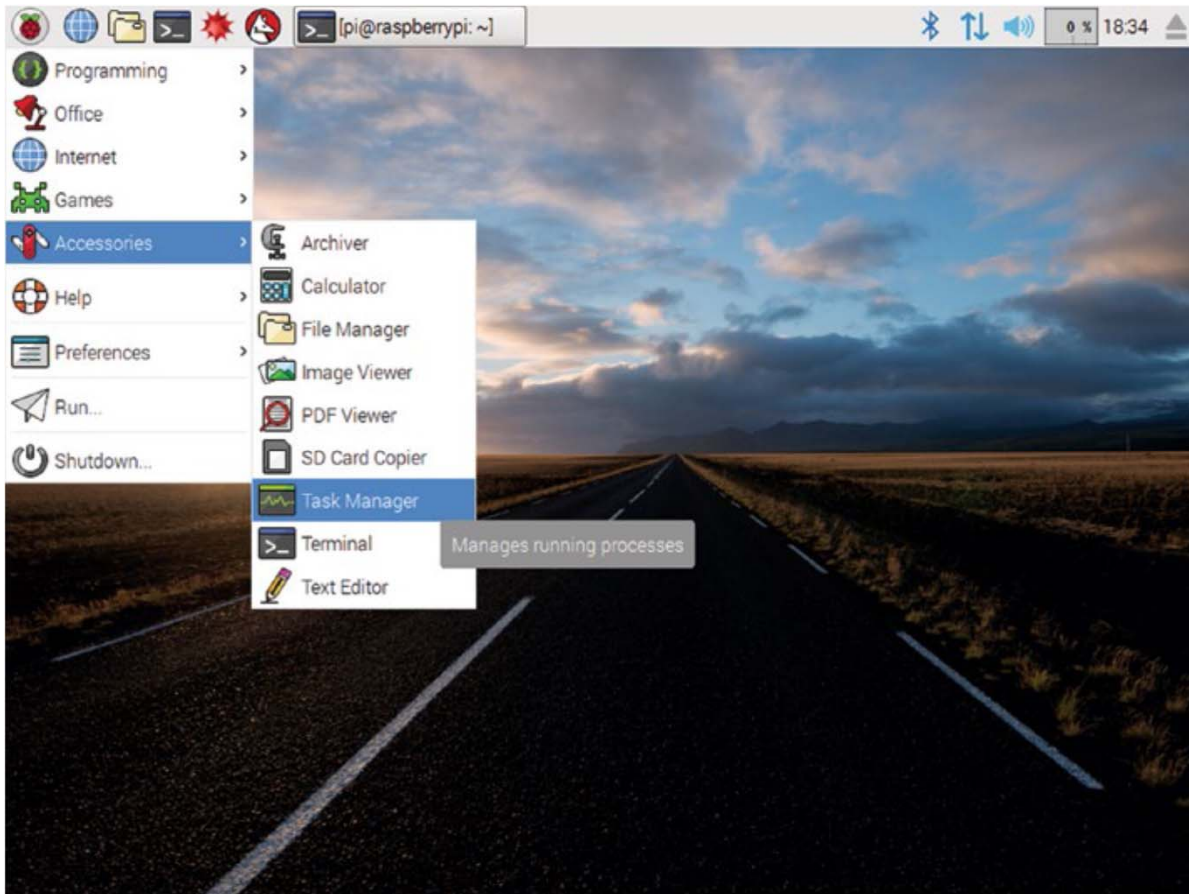
# Τυπικό σύστημα υπολογιστή



## Εγκατάσταση λειτουργικού

- Κατεβάστε 2 βασικά προγράμματα: Το **SD Formatter** ([https://www.sdcard.org/downloads/formatter\\_4/eula\\_windows/](https://www.sdcard.org/downloads/formatter_4/eula_windows/)) και το **Win32DiskImager** (<https://sourceforge.net/projects/win32diskimager/files/latest/download>). Το πρώτο μας επιτρέπει να διαμορφώνουμε κάρτες SD, ενώ το δεύτερο να γράφουμε το λειτουργικό στην κάρτα SD.
- Κατεβάστε τη διανομή **“Raspbian”** σαν image file (<https://www.raspberrypi.org/downloads/raspbian/>).
- Γράψτε το image στην κάρτα SD.
- Αν χρειαστεί να διαμορφώσετε την κάρτα SD, χρησιμοποιήστε το “SDFormatter” (Options: Format Size Adjustment=ON)
- Τοποθετήστε την κάρτα SD στην υποδοχή του Raspberry Pi.
- Ανοίξτε το τροφοδοτικό και παρακολουθήστε το boot στην οθόνη σας.
- Ικανοποιητική χωρητικότητα της κάρτας micro SD είναι 16 GB, Class 10, διαμορφωμένη σε FAT32.

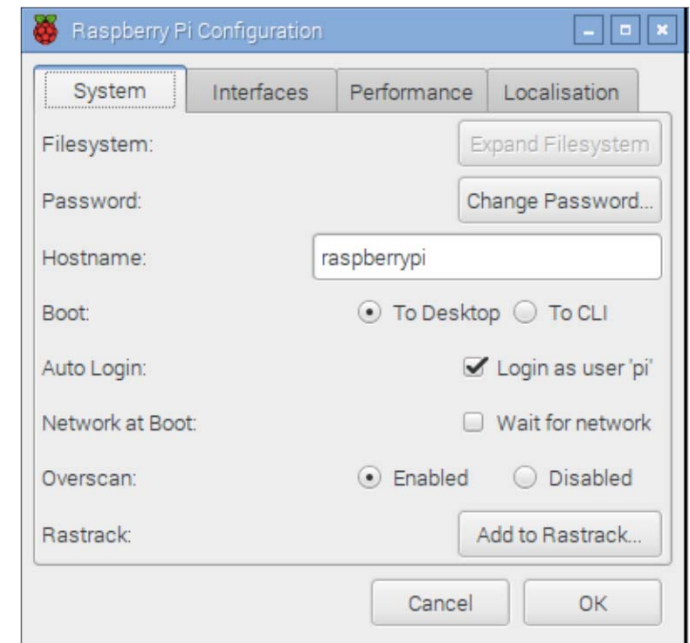
# Λειτουργικό σύστημα Raspbian



Το λειτουργικό σύστημα Raspbian που έχει ενσωματωθεί στα Pi's του μαθήματος συμπεριλαμβάνει τη βιβλιοθήκη OpenCV με τη βοήθεια της οποίας μπορούν να αναπτυχθούν real time αλγόριθμοι επεξεργασίας εικόνας και τεχνητής όρασης χρησιμοποιώντας την κάμερα.

## Ρυθμίσεις μετά την εγκατάσταση

- **Επέκταση συστήματος στην χωρητικότητα της κάρτας SD**  
\$sudo raspi-config ή Menu → Preferences → Configuration (Το τερματικό εκτέλεσης εντολών ανοίγει με CTRL+ALT+T)  
Select “Expand Filesystem”  
\$sudo reboot
- Στο “advanced options” μπορούμε να ενεργοποιήσουμε τα πρωτόκολλα SPI και I2C
- **Select Country**  
\$sudo raspi-config  
Select “Internationalisation Options”  
Select as Time Zone=Europe-Athens  
\$sudo reboot
- **Set Keyboard as US**  
Menu → Mouse and Keyboard Settings → US (English)  
\$sudo apt-get update
- **Set Date and Time**  
\$sudo date MMDDHHMMYY  
\$sudo reboot

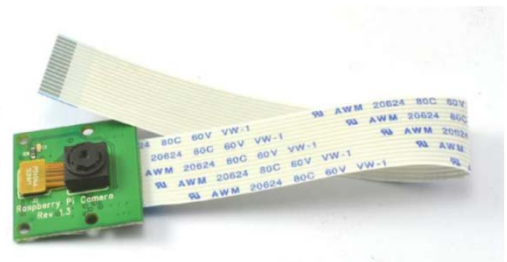


- **Εγκατάσταση RPi.GPIO**

```
$ sudo apt-get install python-dev  
$ sudo apt-get install python-rpi.gpio
```

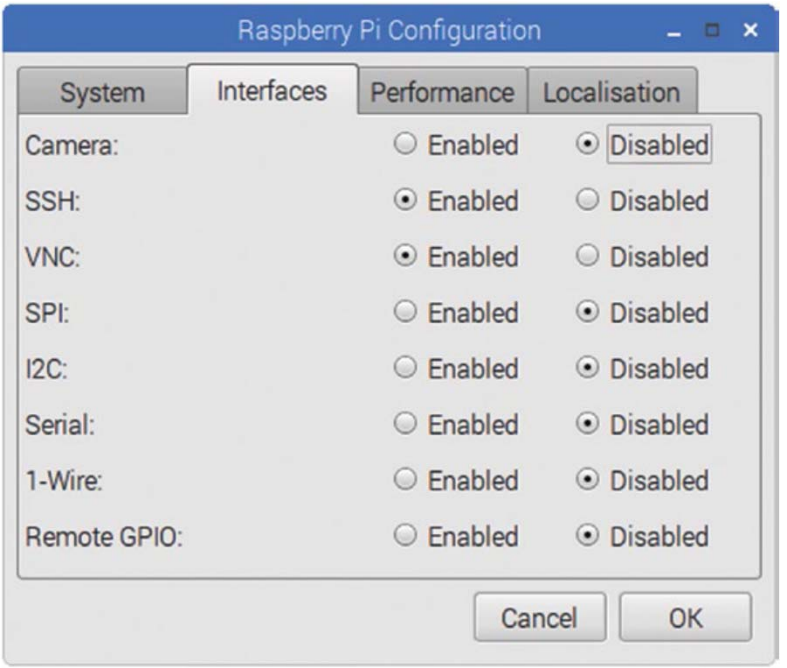
- **Εγκατάσταση κάμερας**

```
$ sudo raspi-config  
Enable Camera
```



```
$ raspistill -o image1.jpg (για λήψη μιας εικόνας)  
$ raspivid -o video.h264 -t 10000 (για λήψη βίντεο 10 sec)
```

- **SSH:** Σύνδεση στο Raspberry Pi μέσω πρωτοκόλλου SSH (Secure Shell ).
- **VNC:** Σύνδεση στο Raspberry Pi μέσω του ενσωματωμένου VNC (Virtual Network Computing) server.
- **SPI:** Το Serial Peripheral Interface (SPI) είναι ένα από τα τρία σειριακά στο GPIO.
- **I2C:** Το Inter-Integrated-Circuit (I2C) είναι το δεύτερο σειριακό στο GPIO.
- **Serial:** Το serial port είναι το τρίτο σειριακό στο GPIO.
- **1-Wire:** Το 1-Wire που βρίσκεται στα GPIO pins διακινεί δεδομένα σε μία γραμμή.
- **Remote GPIO:** Επιτρέπει την πρόσβαση στο GPIO απομακρυσμένα.



# Δικτύωση με WiFi

- Ο πιο εύκολος τρόπος χειρισμού του Raspberry Pi για ανάπτυξη εφαρμογών είναι η σύνδεσή του με μια οθόνη.
- Εάν δεν υπάρχει οθόνη, τότε είναι προτιμότερη η διασύνδεσή του με Laptop μέσω WiFi με τη βοήθεια του VNC server που είναι προ-εγκατεστημένος στη διανομή Raspbian. Το Pi αυτόματα επιτυγχάνει μια διεύθυνση IP από τον router μέσω DHCP (Dynamic Host Configuration Protocol), όμως αυτή η διεύθυνση δεν είναι σταθερή

- Η διεύθυνση σύνδεσης του Pi βρίσκεται δίνοντας την εντολή από το Terminal Window:

```
$ hostname -I  
192.168.1.16
```

- Για να δούμε όλες τις δικτυακές διασυνδέσεις του Pi (ενσύρματες και ασύρματες):

```
$ sudo ifconfig
```

```
$ sudo ifconfig  
eth0      Link encap:Ethernet  HWaddr b8:27:eb:d5:f4:8f  
          inet addr:192.168.1.16  Bcast:192.168.255.255  Mask:255.255.0.0  
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1  
          RX packets:1114 errors:0 dropped:1 overruns:0 frame:0  
          TX packets:1173 errors:0 dropped:0 overruns:0 carrier:0  
          collisions:0 txqueuelen:1000  
          RX bytes:76957 (75.1 KiB)  TX bytes:479753 (468.5 KiB)  
  
lo        Link encap:Local Loopback  
          inet addr:127.0.0.1  Mask:255.0.0.0  
          UP LOOPBACK RUNNING  MTU:16436  Metric:1  
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0  
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0  
          collisions:0 txqueuelen:0  
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)  
  
wlan0     Link encap:Ethernet  HWaddr 00:0f:53:a0:04:57  
          inet addr:192.168.1.13  Bcast:192.168.255.255  Mask:255.255.0.0  
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1  
          RX packets:38 errors:0 dropped:0 overruns:0 frame:0  
          TX packets:28 errors:0 dropped:0 overruns:0 carrier:0  
          collisions:0 txqueuelen:1000  
          RX bytes:6661 (6.5 KiB)  TX bytes:6377 (6.2 KiB)
```

## Ρύθμιση για σταθερή διεύθυνση IP

- Εκτέλεση στο Terminal Window:

```
$sudo nano /etc/network/interfaces
```

- Αλλάζουμε το **“iface wlan0 inet manual”** σε **“iface wlan0 inet static”**

- Τοποθετήστε από κάτω:

```
address 192.168.1.13      # Η στατική διεύθυνση που θέλουμε
netmask 255.255.255.0
gateway 192.168.1.1     # Η IP του router μας (στον Η/Υ μας με ipconfig-Default Gateway)
```

- Save και επανεκκίνηση με: `$sudo reboot`
- Κάθε φορά που θα γίνεται επανεκκίνηση, το Pi θα έχει αυτήν τη στατική διεύθυνση.
- Η απόδοση στατικής διεύθυνσης σε περίπτωση σύνδεσης του Pi με καλώδιο Ethernet γίνεται με όμοιο τρόπο στη γραμμή: **“iface eth0 inet manual”**.
- **Η κάρτα SD με το εγκατεστημένο raspbian έχει στατική διεύθυνση WiFi: 192.168.2.5.**
- **Στη σύνδεση του Pi με τον δικό σας Η/Υ αλλάξτε τις παραπάνω ρυθμίσεις σύμφωνα με τις δικές σας ανάγκες.**
- **Πως μπορεί να γίνει αυτό αν δεν διαθέτουμε οθόνη και πρέπει να το κάνουμε από τον Η/Υ μας (π.χ. Laptop)?**

## Σύνδεση με laptop για πρώτη φορά

- Κάνουμε boot το Raspberry Pi χωρίς να βλέπουμε κάτι, αφού δεν έχουμε διαθέσιμη οθόνη. Το WiFi του δεν μπορεί να είναι συνδεδεμένο σε κάποιο δίκτυο.
- Συνδέουμε με ένα καλώδιο Ethernet το Pi με το laptop.
- Στο Command Prompt του Η/Υ (Windows+X) εκτελούμε “arp -a”.
- Το MAC Address του Pi είναι:
  - Ethernet: b8:27:eb:1f:99:bb
  - WiFi: b8:27:eb:4a:cc:ee
- Επομένως, Raspberry Pi IP address:
  - Ethernet: 169.254.75.234
  - WiFi: 192.168.2.5
- Ανοίγουμε το VNC Viewer ή το Putty και εισάγουμε την Ethernet IP.
- Εισάγουμε username και password (pi και raspberry αντίστοιχα).
- Εμφανίζεται η οθόνη του Pi οπότε τώρα μπορούν να γίνουν οι προηγούμενες ρυθμίσεις για τη σύνδεση με WiFi και η απόδοση στατικής διεύθυνσης.

```
C:\Users\Nick>arp -a

Interface: 169.254.250.50 --- 0x2
Internet Address      Physical Address      Type
169.254.75.234        b8-27-eb-1f-99-bb    dynamic
169.254.255.255      00-ff-ff-ff-ff-ff    static
224.0.0.22           01-00-5e-00-00-16    static
224.0.0.251          01-00-5e-00-00-fb    static
224.0.0.252          01-00-5e-00-00-fc    static
239.255.255.250     01-00-5e-7f-ff-fa    static
255.255.255.255     ff-ff-ff-ff-ff-ff    static

Interface: 192.168.2.6 --- 0x3
Internet Address      Physical Address      Type
192.168.2.1          24-4d-aa-97-4d-00    dynamic
192.168.2.5          b8-27-eb-4a-cc-ee    dynamic
192.168.2.255       ff-ff-ff-ff-ff-ff    static
224.0.0.22           01-00-5e-00-00-16    static
224.0.0.251          01-00-5e-00-00-fb    static
224.0.0.252          01-00-5e-00-00-fc    static
239.255.255.250     01-00-5e-7f-ff-fa    static
255.255.255.255     ff-ff-ff-ff-ff-ff    static

Interface: 192.168.56.1 --- 0xb
Internet Address      Physical Address      Type
192.168.56.255      ff-ff-ff-ff-ff-ff    static
224.0.0.22           01-00-5e-00-00-16    static
224.0.0.251          01-00-5e-00-00-fb    static
224.0.0.252          01-00-5e-00-00-fc    static
239.255.255.250     01-00-5e-7f-ff-fa    static

Interface: 192.168.42.2 --- 0xe
Internet Address      Physical Address      Type
192.168.42.255      ff-ff-ff-ff-ff-ff    static
224.0.0.22           01-00-5e-00-00-16    static
224.0.0.251          01-00-5e-00-00-fb    static
224.0.0.252          01-00-5e-00-00-fc    static
239.255.255.250     01-00-5e-7f-ff-fa    static
```

# Ασύρματη επικοινωνία με τον Η/Υ μέσω του VNC server

- Το Pi έχει εγκατεστημένο ένα VNC server (tightvncserver ή real-vnc-server).

- Εάν δεν υπάρχει, πρέπει να εγκατασταθεί:

```
$ sudo apt-get update
```

```
$ sudo apt-get install realvnc-vnc-server
```

- Για να εγκατασταθεί η πιο πρόσφατη έκδοση:

```
$ sudo apt-get remove tightvncserver
```

```
$ sudo apt-get install realvnc-vnc-server
```

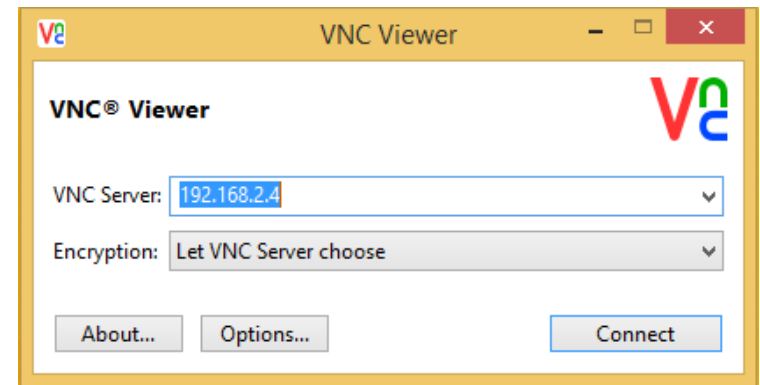
- Κάνετε reboot το Pi.

- Εγκαταστήστε στον υπολογιστή σας τον VNC Viewer (π.χ. για Windows).

- Εκτελέστε στον υπολογιστή σας τον VNC Viewer.

- Γράψτε την IP διεύθυνση και τον κωδικό σας για να συνδεθείτε (Default username=pi – password=raspberry).

- Το hostname του Pi είναι: raspberrypi.

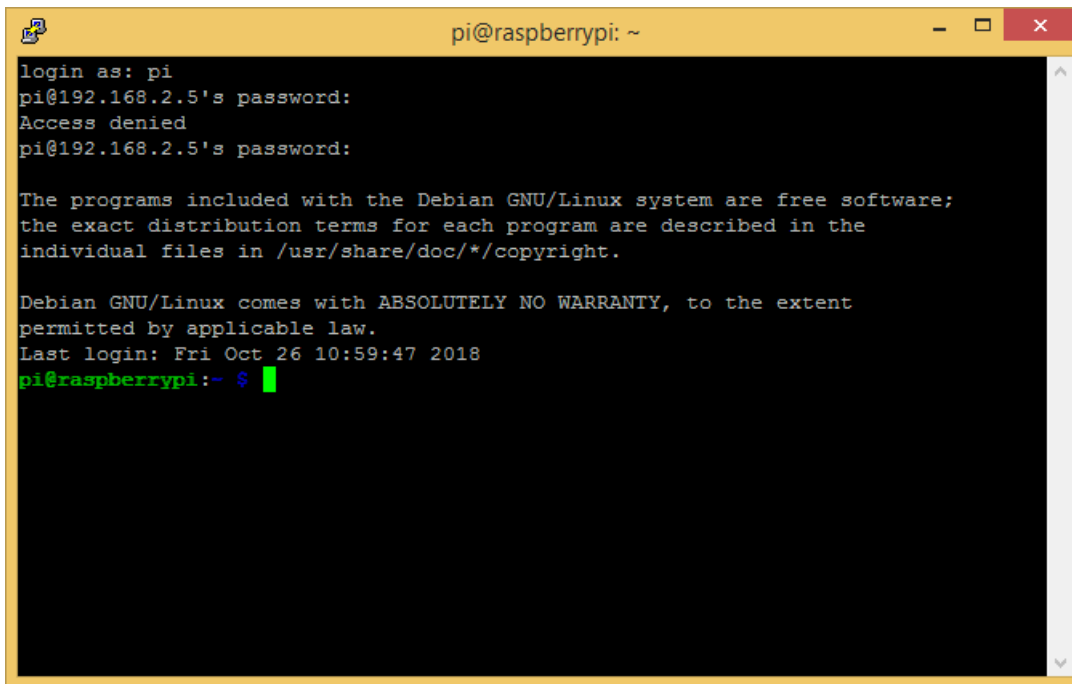


# Ασύρματη επικοινωνία με τον Η/Υ μέσω SSH

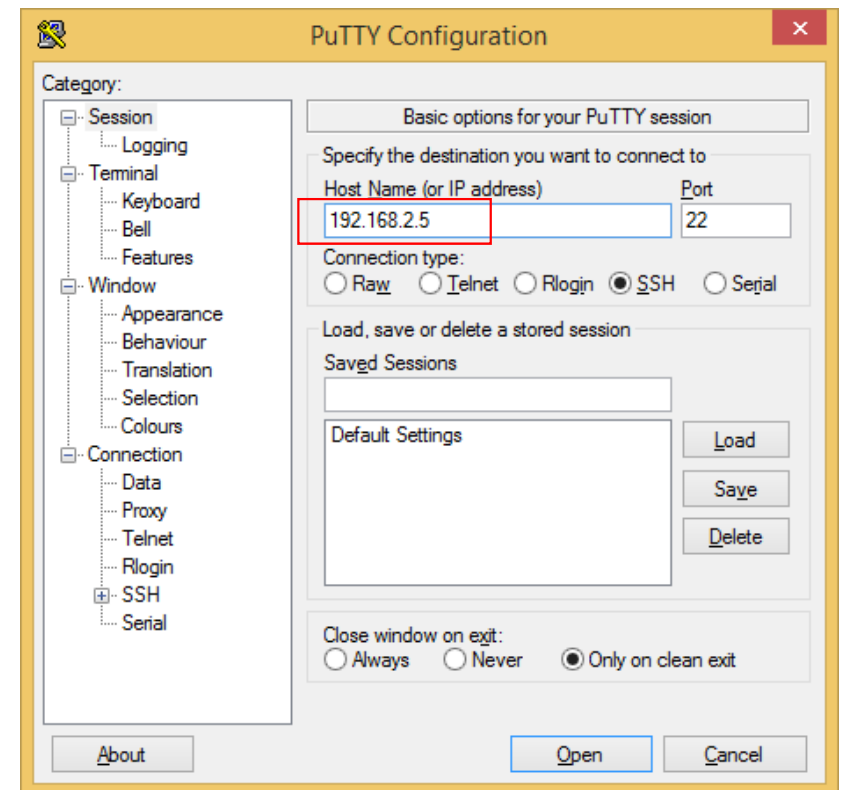
- Ενεργοποίηση του SSH με `sudo raspi-config` και επιλογή του.
- Ανοίγουμε το Putty στα Windows και δημιουργούμε ένα SSH session.
- Δίνουμε την IP διεύθυνση και στο τερματικό που ανοίγει τοποθετούμε:

login as: **pi**

pi@192.168.2.5 's password: **raspberr**y

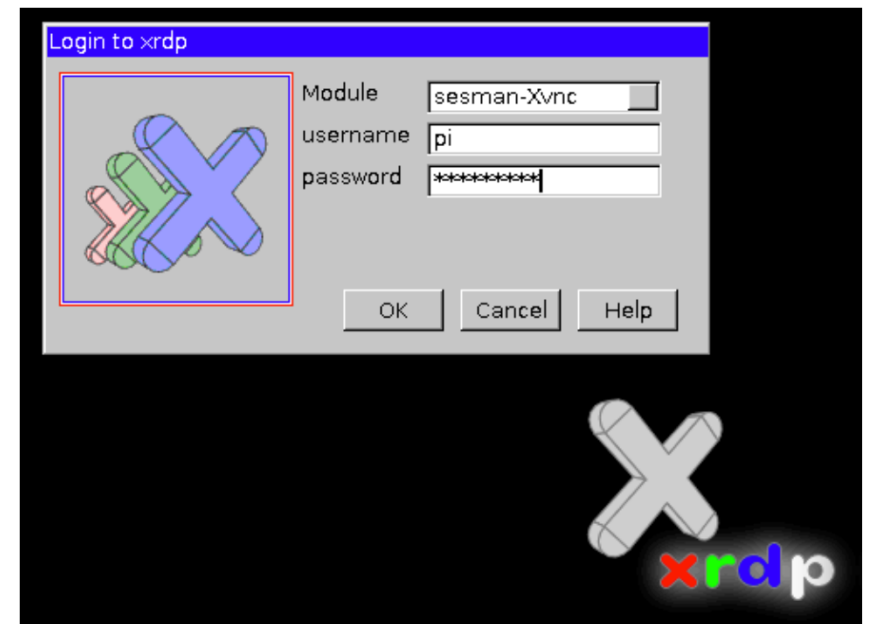


```
pi@raspberrypi: ~  
login as: pi  
pi@192.168.2.5's password:  
Access denied  
pi@192.168.2.5's password:  
  
The programs included with the Debian GNU/Linux system are free software;  
the exact distribution terms for each program are described in the  
individual files in /usr/share/doc/*/copyright.  
  
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent  
permitted by applicable law.  
Last login: Fri Oct 26 10:59:47 2018  
pi@raspberrypi:~$
```



# Ασύρματη επικοινωνία με τον Η/Υ μέσω RDP

- Η σύνδεση του Pi με υπολογιστή μέσω του RDP (Remote Desktop Protocol) προϋποθέτει την εγκατάσταση του XRDP στο Pi:  
\$ sudo apt-get update  
\$ sudo apt-get install xrdp
- Μόλις εγκατασταθεί ξεκινάει αμέσως η λειτουργία του, όπως επίσης και κάθε φορά που κάνει reboot.
- Στα Windows 7 και μετά υπάρχει εγκατεστημένος RDP client (*All Programs/Accessories/Remote Desktop Connection*)
- username=**pi** και password=**raspberrypi**
- Αυτός ο τρόπος λειτουργεί πιο αποδοτικά απ' ό,τι με τον VNC server.



## Κοινή χρήση του φακέλου Home

- Είναι επιθυμητή η κοινή χρήση του φακέλου Home και η μεταφορά αρχείων μεταξύ αυτού και μιας μνήμης USB. Αυτό απαιτεί το πρωτόκολλο SMB (Server Message Block) .
- Εγκαθιστούμε και ρυθμίζουμε το πρόγραμμα Samba:  
\$ sudo apt-get install samba  
\$ sudo apt-get install samba-common-bin
- Εάν τοποθετηθεί μνήμη USB, τα περιεχόμενά της είναι:  
\$ cd /media  
\$ ls
- Ρύθμιση του Samba για κοινή χρήση:
- Πρώτα προσθέτουμε έναν χρήστη samba (pi):  
\$ sudo smbpasswd -a pi  
New SMB password:  
Retype new SMB password:  
Added user pi.
- Στο αρχείο `"/etc/samba/smb.conf"` κάνουμε τα εξής:  
\$ sudo nano /etc/samba/smb.conf  
workgroup = HOME (να συμφωνεί με το αντίστοιχο των Windows)  
Στη γραμμή `# security = user` αφαιρούμε το `"#"` για ενεργοποίηση της ασφάλειας.  
Στο τέλος του αρχείου προσθέτουμε τα εξής:

[USB]

path = /media/NAS

comment = NAS Drive

valid users = pi

writable = yes

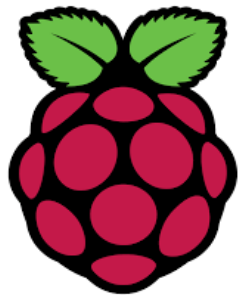
browseable = yes

create mask = 0777

public = yes

- Σώζουμε το αρχείο και επανεκινούμε το πρόγραμμα:  
\$ sudo /etc/init.d/samba restart
- Η μνήμη USB έχει κοινή χρήση.

**Σημείωση:** Μπορεί να χρησιμοποιηθεί η δυνατότητα File Transfer του VNC.



GPIO

## Raspberry Pi 3 GPIO Header

Pin#	NAME		NAME	Pin#
01	3.3v DC Power		DC Power 5v	02
03	GPIO02 (SDA1 , I <sup>2</sup> C)		DC Power 5v	04
05	GPIO03 (SCL1 , I <sup>2</sup> C)		Ground	06
07	GPIO04 (GPIO_GCLK)		(TXD0) GPIO14	08
09	Ground		(RXD0) GPIO15	10
11	GPIO17 (GPIO_GEN0)		(GPIO_GEN1) GPIO18	12
13	GPIO27 (GPIO_GEN2)		Ground	14
15	GPIO22 (GPIO_GEN3)		(GPIO_GEN4) GPIO23	16
17	3.3v DC Power		(GPIO_GEN5) GPIO24	18
19	GPIO10 (SPI_MOSI)		Ground	20
21	GPIO09 (SPI_MISO)		(GPIO_GEN6) GPIO25	22
23	GPIO11 (SPI_CLK)		(SPI_CE0_N) GPIO08	24
25	Ground		(SPI_CE1_N) GPIO07	26
27	ID_SD (I <sup>2</sup> C ID EEPROM)		(I <sup>2</sup> C ID EEPROM) ID_SC	28
29	GPIO05		Ground	30
31	GPIO06		GPIO12	32
33	GPIO13		Ground	34
35	GPIO19		GPIO16	36
37	GPIO26		GPIO20	38
39	Ground		GPIO21	40

Rev. 2  
29/02/2016

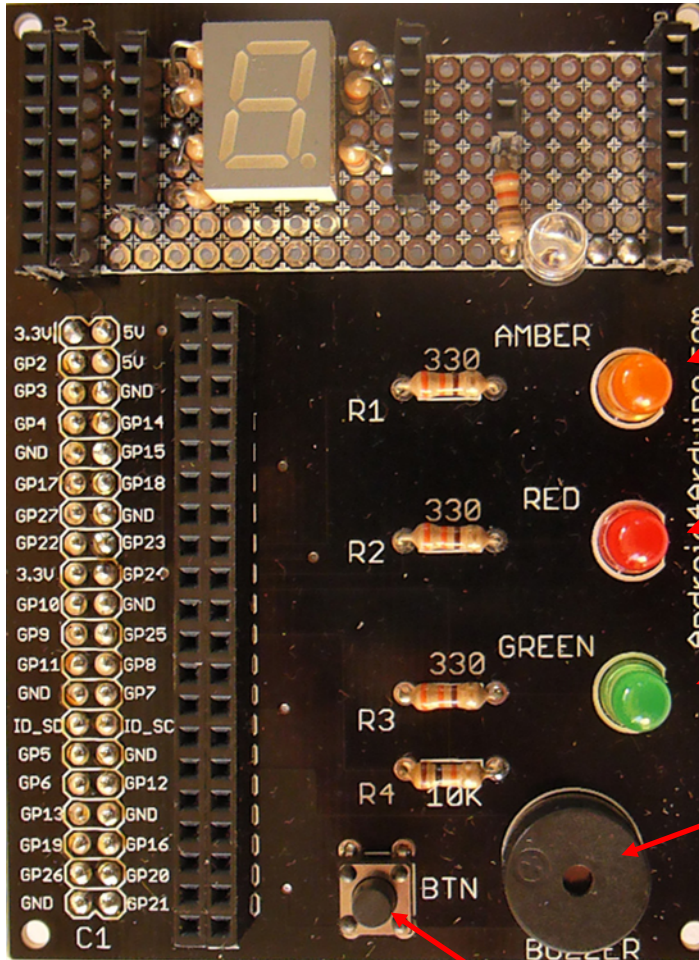
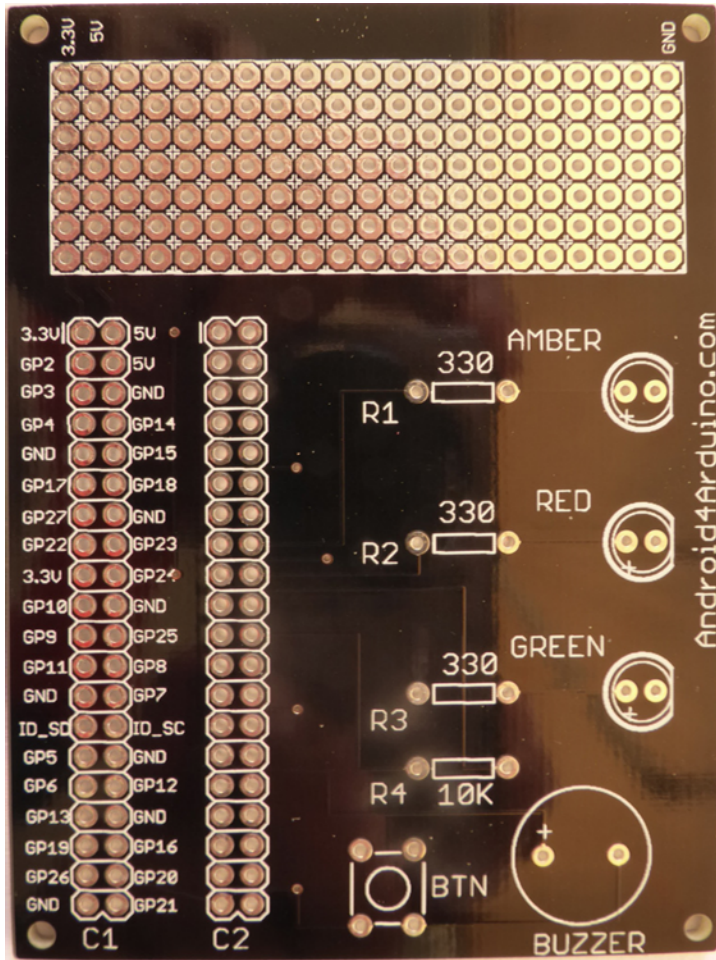
[www.element14.com/RaspberryPi](http://www.element14.com/RaspberryPi)

### Εγκατάσταση βιβλιοθήκης GPIO

```
$ sudo apt-get install python-dev
$ sudo apt-get install python-rpi.gpio
```

(Υπάρχει στη διανομή Raspbian)

# Traffic Hat



GPIO-16

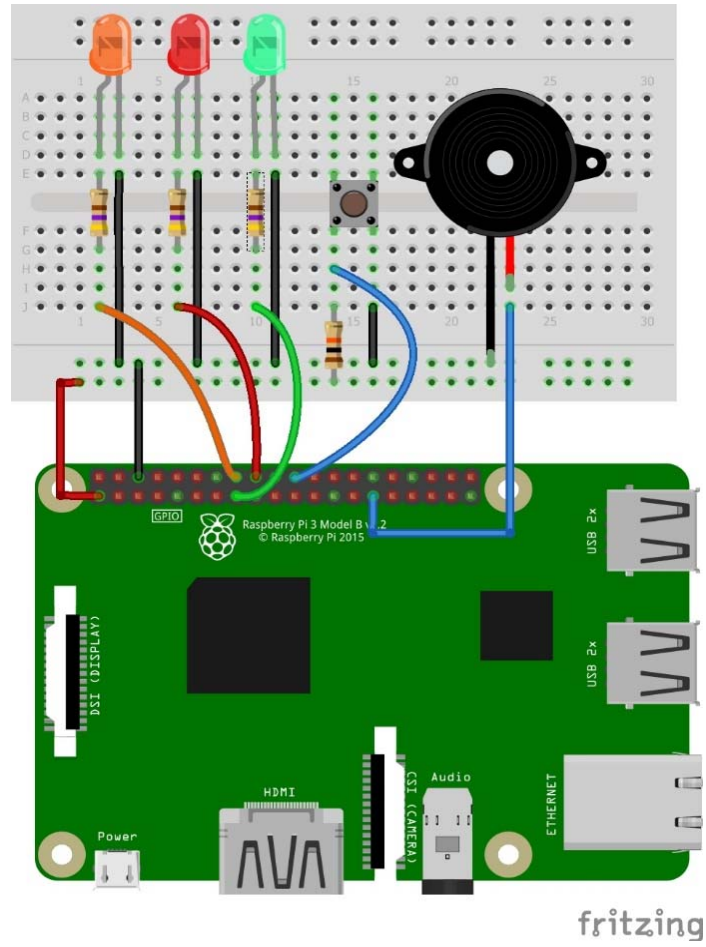
GPIO-18

GPIO-15

GPIO-29

Σημειώσεις στο αρχείο:  
Traffic Hat for Raspberry pi.pdf

GPIO-22

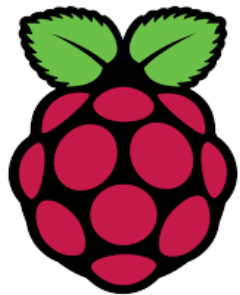


fritzing

- Βιβλιοθήκη και αντικείμενο GPIO:
- Σχήμα αριθμοδότησης ακροδεκτών GPIO:
- Operation mode ενός pin σαν είσοδος ή έξοδος:
- Έξοδος σε ένα digital pin:  
LOW = “0” ή “False” and HIGH = “1” ή “True”.
- Είσοδος από ένα digital pin:  
Επιστρέφει “False” όταν η είσοδος είναι 0V και “True” όταν η είσοδος είναι +3.3V.
- Τα pins εισόδου μπορούν να έχουν internal pull-up ή pull-down αντιστάσεις, με τις εντολές:  
**GPIO.setup(pin, GPIO.IN, pull\_up\_down=GPIO.PUD\_UP) or**  
**GPIO.setup(pin, GPIO.IN, pull\_up\_down=GPIO.PUD\_DOWN)**
- Το pin 12 (PWM0) μπορεί να δημιουργήσει αναλογική τάση εξόδου σε μορφή PWM. Πρώτα ορίζουμε μια μεταβλητή με τη συχνότητα PWM και μετά δημιουργούμε το PWM με το επιθυμητό Duty Cycle:  
**pwm=GPIO.PWM(pin, frequency)**  
**pwm.start(% Duty Cycle)**  
Αλλαγή Duty Cycle: **pwm.ChangeDutyCycle(% Duty Cycle)**  
Παύση της εξόδου PWM: **pwm.stop()**
- Για χρονοκαθυστερήσεις χρησιμοποιούμε τη βιβλιοθήκη “time”:  
**include time**  
**time.sleep(delay in seconds)**

```
import RPi.GPIO as GPIO
GPIO.setmode(GPIO.BOARD)
GPIO.setup(pin, GPIO.IN or GPIO.OUT)
GPIO.output(pin, GPIO.LOW or GPIO.HIGH)
```

```
GPIO.input(pin)
```



# Παραδείγματα

# Οδήγηση LED

Να γίνει πρόγραμμα το οποίο θα ανάβει κατά σειρά τα 3 LED στο Traffic Hat με ενδιάμεση χρονοκαθυστέρηση 2 sec. Όταν πατηθεί CTRL+C το πρόγραμμα θα τερματίζεται.

```
0 import RPi.GPIO as GPIO
1 import time
2
3 #pin definitions
4 ledAmber=16
5 ledRed=18
6 ledGreen=15
7
8 #numbering scheme
9 GPIO.setmode(GPIO.BOARD)
10 #make pins Outputs
11 GPIO.setup(ledAmber, GPIO.OUT)
12 GPIO.setup(ledRed, GPIO.OUT)
13 GPIO.setup(ledGreen, GPIO.OUT)
14 #initial state of outputs
15 GPIO.output(ledAmber, GPIO.LOW)
16 GPIO.output(ledRed, GPIO.LOW)
17 GPIO.output(ledGreen, GPIO.LOW)
18
```

```
19 try:
20     while True:
21         print("Press CTRL+C to exit")
22         GPIO.output(ledAmber, GPIO.HIGH)
23         GPIO.output(ledRed, GPIO.HIGH)
24         GPIO.output(ledGreen, GPIO.HIGH)
25         time.sleep(2)
26         GPIO.output(ledAmber, GPIO.LOW)
27         GPIO.output(ledRed, GPIO.LOW)
28         GPIO.output(ledGreen, GPIO.LOW)
29         GPIO.output(buzzerPin, GPIO.LOW)
30         time.sleep(2)
31 except KeyboardInterrupt:
32     GPIO.cleanup()
```

# Οδήγηση Πλήκτρου και LED

Να γίνει πρόγραμμα το οποίο θα ενεργοποιεί τα LED και το buzzer για 5 sec όταν υπάρξει ένα πλήρες πάτημα του πλήκτρου (κάτω-επάνω). Με CTRL+C πραγματοποιείται έξοδος.

```
1 import Rpi.GPIO as GPIO
2 import time
3
4 #pin definitions
5 btnPin=22
6 ledAmber=16
7 ledRed=18
8 ledGreen=15
9 buzzerPin=29
10
11 GPIO.setmode(GPIO.BOARD)
12 #Button has an external pull-up resistor
13 #GPIO.setup(btnPin, GPIO.IN)
14 #Button has no external pull-up resistor
15 GPIO.setup(btnPin, GPIO.IN,
pull_up_down=GPIO.PUD_UP)
16 GPIO.setup(ledAmber, GPIO.OUT)
17 GPIO.setup(ledRed, GPIO.OUT)
18 GPIO.setup(ledGreen, GPIO.OUT)
19 GPIO.setup(buzzerPin, GPIO.OUT)
20
21 #initial state of outputs
22 GPIO.output(ledAmber, GPIO.LOW)
```

```
23 GPIO.output(ledRed, GPIO.LOW)
24 GPIO.output(ledGreen, GPIO.LOW)
25 GPIO.output(buzzerPin, GPIO.LOW)
26
27 try:
28     while True:
29         print("Press button to start. CTRL+C to exit")
30         if not GPIO.input(GPIO.btnPin)
31             while not GPIO.input(GPIO.btnPin)
32                 print("release the button")
33             else:
34                 GPIO.output(ledAmber, GPIO.HIGH)
35                 GPIO.output(ledRed, GPIO.HIGH)
36                 GPIO.output(ledGreen, GPIO.HIGH)
37                 GPIO.output(buzzerPin, GPIO.HIGH)
38                 time.sleep(5)
39                 GPIO.output(ledAmber, GPIO.LOW)
40                 GPIO.output(ledRed, GPIO.LOW)
41                 GPIO.output(ledGreen, GPIO.LOW)
42                 GPIO.output(buzzerPin, GPIO.LOW)
43 except KeyboardInterrupt:
44     GPIO.cleanup()
```

# Οδήγηση LED με PWM

Να γίνει πρόγραμμα το οποίο θα ενεργοποιεί το κόκκινο LED με παλμό PWM και Duty Cycle=100% όταν το πλήκτρο είναι πατημένο και με Duty Cycle=10% όταν το πλήκτρο είναι ελεύθερο. Με CTRL+C πραγματοποιείται έξοδος.

```
1 import Rpi.GPIO as GPIO
2 import time
3
4 #pin definitions
5 btnPin=22
6 ledPin=18
7
8 #set PWM duty cycle
9 dc_released=10
10 dc_pressed=100
11
12 GPIO.setmode(GPIO.BOARD)
13 #Button has an external pull-up resistor
14 GPIO.setup(btnPin, GPIO.IN)
15 GPIO.setup(ledPin, GPIO.OUT)
16 pwm=GPIO.PWM(ledPin, 100)
17
```

```
18 #initial state of LED
19 pwm.start(dc_released)
20
21 print("Press button to start. CTRL+C to exit")
22 try:
23     while True:
24         if not GPIO.input(GPIO.btnPin)
25             pwm.ChangeDutyCycle(dc_pressed)
26             time.sleep(0.05)
27         else
28             pwm.ChangeDutyCycle(dc_released)
29             time.sleep(0.05)
30 except KeyboardInterrupt:
31     GPIO.cleanup()
```

## Χρήση της βιβλιοθήκης gpiozero και της κλάσης Traffic Hat

Να γίνει πρόγραμμα το οποίο θα ενεργοποιεί τα LED και το buzzer όταν το πλήκτρο είναι πατημένο και θα τα απενεργοποιεί όταν είναι ελεύθερο.

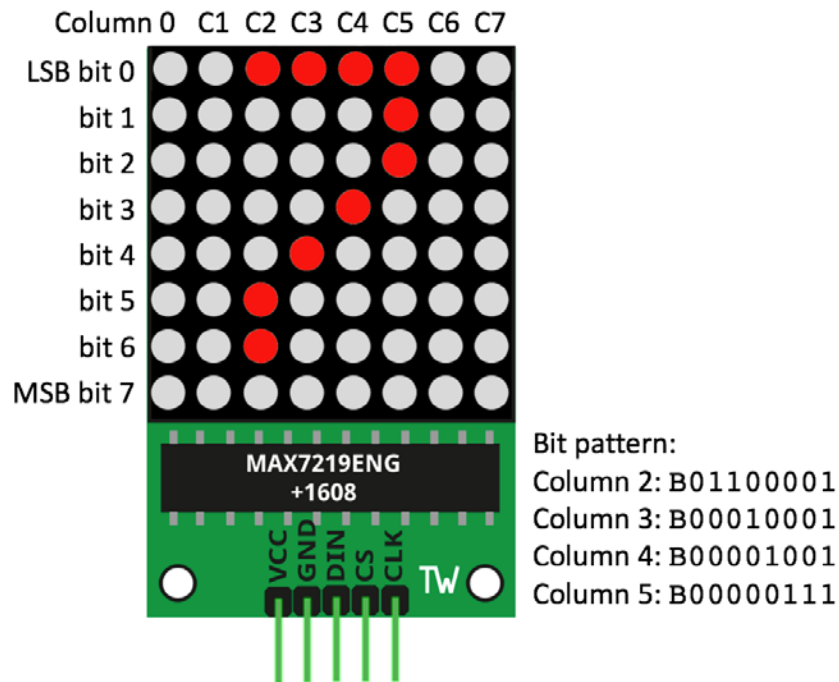
```
1 from gpiozero import TrafficHat
2 import time
3 from signal import pause
4
5 hat=TrafficHat()
6
7 hat.button.wait_for_press()
8 hat.lights.on()
9 hat.buzzer.on()
10 hat.button.wait_for_release()
11 hat.lights.off()
12 hat.buzzer.off()
13
14 pause()
```

## Χρήση της βιβλιοθήκης gpiozero και της κλάσης Traffic Hat

Να γίνει πρόγραμμα το οποίο θα ενεργοποιεί τα LED με PWM (DC=100%) όταν το πλήκτρο είναι πατημένο και θα τα ενεργοποιεί με PWM (DC=10%) όταν είναι ελεύθερο.

```
1 from gpiozero import TrafficHat
2 from gpiozero import PWMLED
3 import time
4
5 hat=TrafficHat(pwm=True)
6
7 while True:
8     hat.button.wait_for_press()
9     hat.lights.amber.value(1)
10    hat.lights.red.value(1)
11    hat.lights.green.value(1)
12    time.sleep(0.05)
13    hat.button.wait_for_release()
14    hat.lights.amber.value(0.1)
15    hat.lights.amber.value(0.1)
16    hat.lights.amber.value(0.1)
17    time.sleep(0.05)
```

# Οδήγηση LED matrix 8X8



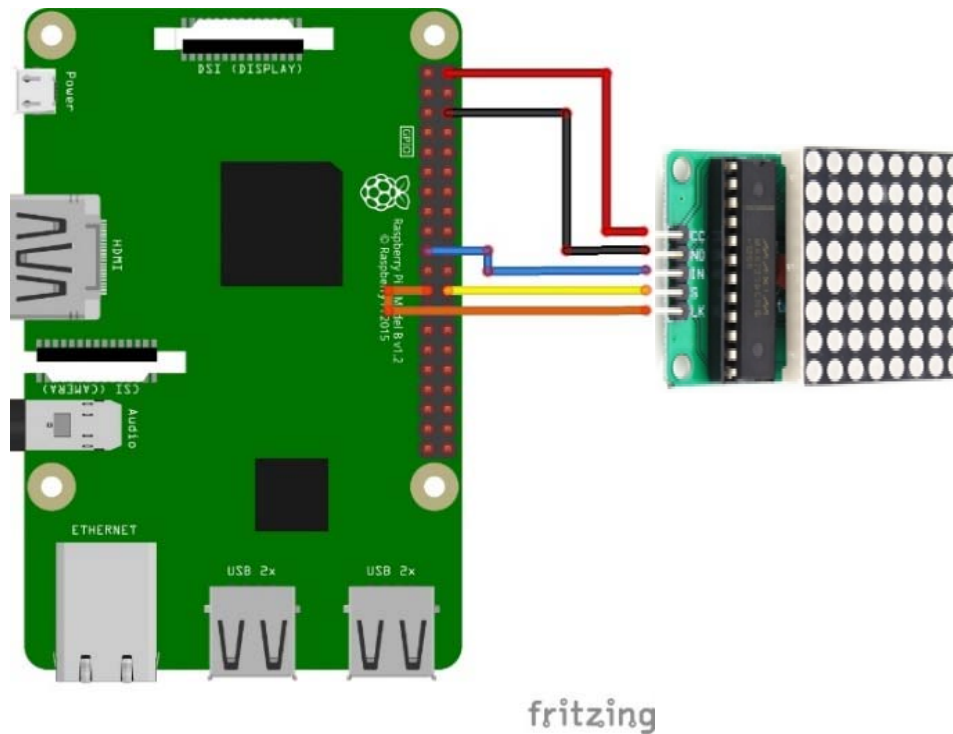
Το Raspberry Pi μπορεί να συνδεθεί με ένα τέτοιο module (κατά μέγιστο μέχρι 8):

- Χρησιμοποιώντας pins από το GPIO και προσομοιώνοντας το SPI interface του MAX7219.
- Χρησιμοποιώντας τα SPI pins από το GPIO.

GPIO PINS	8x8 MATRIX	SPI PINS
2 (+5V)	Vcc	2 (+5V)
6 (GND)	GND	6 (GND)
31 (GPIO06)	DIN	19 (GPIO10)
33 (GPIO13)	CS	24 (GPIO08)
29 (GPIO05)	CLK	23 (GPIO11)

## Οδήγηση LED matrix 8X8 με SPI

Να γίνει πρόγραμμα το οποίο θα εμφανίζει στην οθόνη ένα μήνυμα χρησιμοποιώντας το SPI του Raspberry Pi και τη βιβλιοθήκη “spidev”.



```

1 #8X8 LED MATRIX USING SPI
2 import spidev
3 import RPi.GPIO as GPIO
4 from time import sleep
5 from signal import pause
6
7 #MAX7219 registers
8 MAX7219_NOOP = 0x00
9 MAX7219_DIGIT0 = 0x01
10 MAX7219_DIGIT1 = 0x02
11 MAX7219_DIGIT2 = 0x03
12 MAX7219_DIGIT3 = 0x04
13 MAX7219_DIGIT4 = 0x05
14 MAX7219_DIGIT5 = 0x06
15 MAX7219_DIGIT6 = 0x07
16 MAX7219_DIGIT7 = 0x08
17 MAX7219_DECODEMODE = 0x09
18 MAX7219_INTENSITY = 0x0A
19 MAX7219_SCANLIMIT = 0x0B
20 MAX7219_SHUTDOWN = 0x0C
21 MAX7219_TEST = 0x0F
22
23 text='THIS EXAMPLE IS ABOUT DRIVING A LED
      MATRIX WITH MAX7219 WITH SPI'
24

```

```

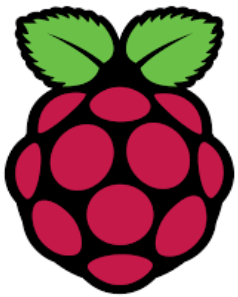
25 cp437_font = {'0':( 0x3E, 0x7F, 0x71, 0x59, 0x4D, 0x7F, 0x3E, 0x00 ), # '0'
26               '1':( 0x40, 0x42, 0x7F, 0x7F, 0x40, 0x40, 0x00, 0x00 ), # '1'
27               '2':( 0x62, 0x73, 0x59, 0x49, 0x6F, 0x66, 0x00, 0x00 ), # '2'
28               '3':( 0x22, 0x63, 0x49, 0x49, 0x7F, 0x36, 0x00, 0x00 ), # '3'
29               '4':( 0x18, 0x1C, 0x16, 0x53, 0x7F, 0x7F, 0x50, 0x00 ), # '4'
30               '5':( 0x27, 0x67, 0x45, 0x45, 0x7D, 0x39, 0x00, 0x00 ), # '5'
31               '6':( 0x3C, 0x7E, 0x4B, 0x49, 0x79, 0x30, 0x00, 0x00 ), # '6'
32               '7':( 0x03, 0x03, 0x71, 0x79, 0x0F, 0x07, 0x00, 0x00 ), # '7'
33               '8':( 0x36, 0x7F, 0x49, 0x49, 0x7F, 0x36, 0x00, 0x00 ), # '8'
34               '9':( 0x06, 0x4F, 0x49, 0x69, 0x3F, 0x1E, 0x00, 0x00 ), # '9'
35               ' ': ( 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00 ), # ' '
36               'A':( 0x7C, 0x7E, 0x13, 0x13, 0x7E, 0x7C, 0x00, 0x00 ), # 'A'
37               'B':( 0x41, 0x7F, 0x7F, 0x49, 0x49, 0x7F, 0x36, 0x00 ), # 'B'
38               'C':( 0x1C, 0x3E, 0x63, 0x41, 0x41, 0x63, 0x22, 0x00 ), # 'C'
39               'D':( 0x41, 0x7F, 0x7F, 0x41, 0x63, 0x3E, 0x1C, 0x00 ), # 'D'
40               'E':( 0x41, 0x7F, 0x7F, 0x49, 0x5D, 0x41, 0x63, 0x00 ), # 'E'
41               'F':( 0x41, 0x7F, 0x7F, 0x49, 0x1D, 0x01, 0x03, 0x00 ), # 'F'
42               'G':( 0x1C, 0x3E, 0x63, 0x41, 0x51, 0x73, 0x72, 0x00 ), # 'G'
43               'H':( 0x7F, 0x7F, 0x08, 0x08, 0x7F, 0x7F, 0x00, 0x00 ), # 'H'
44               'I':( 0x00, 0x41, 0x7F, 0x7F, 0x41, 0x00, 0x00, 0x00 ), # 'I'
45               'J':( 0x30, 0x70, 0x40, 0x41, 0x7F, 0x3F, 0x01, 0x00 ), # 'J'
46               'K':( 0x41, 0x7F, 0x7F, 0x08, 0x1C, 0x77, 0x63, 0x00 ), # 'K'
47               'L':( 0x41, 0x7F, 0x7F, 0x41, 0x40, 0x60, 0x70, 0x00 ), # 'L'
48               'M':( 0x7F, 0x7F, 0x0E, 0x1C, 0x0E, 0x7F, 0x7F, 0x00 ), # 'M'
49               'N':( 0x7F, 0x7F, 0x06, 0x0C, 0x18, 0x7F, 0x7F, 0x00 ), # 'N'
50               'O':( 0x1C, 0x3E, 0x63, 0x41, 0x63, 0x3E, 0x1C, 0x00 ), # 'O'
51               'P':( 0x41, 0x7F, 0x7F, 0x49, 0x09, 0x0F, 0x06, 0x00 ), # 'P'

```

```

52         'Q':( 0x1E, 0x3F, 0x21, 0x71, 0x7F, 0x5E, 0x00, 0x00 ), # 'Q'
53         'R':( 0x41, 0x7F, 0x7F, 0x09, 0x19, 0x7F, 0x66, 0x00 ), # 'R'
54         'S':( 0x26, 0x6F, 0x4D, 0x59, 0x73, 0x32, 0x00, 0x00 ), # 'S'
55         'T':( 0x03, 0x41, 0x7F, 0x7F, 0x41, 0x03, 0x00, 0x00 ), # 'T'
56         'U':( 0x7F, 0x7F, 0x40, 0x40, 0x7F, 0x7F, 0x00, 0x00 ), # 'U'
57         'V':( 0x1F, 0x3F, 0x60, 0x60, 0x3F, 0x1F, 0x00, 0x00 ), # 'V'
58         'W':( 0x7F, 0x7F, 0x30, 0x18, 0x30, 0x7F, 0x7F, 0x00 ), # 'W'
59         'X':( 0x43, 0x67, 0x3C, 0x18, 0x3C, 0x67, 0x43, 0x00 ), # 'X'
60         'Y':( 0x07, 0x4F, 0x78, 0x78, 0x4F, 0x07, 0x00, 0x00 ), # 'Y'
61         'Z':( 0x47, 0x63, 0x71, 0x59, 0x4D, 0x67, 0x73, 0x00 ), # 'Z'
62     }
63
64 def sendSPI(address, data):
65     spi.xfer([address, data]) #Send address&data to MAX7219
66
67 GPIO.setmode(GPIO.BOARD)
68 spi = spidev.SpiDev()         #create spi object
69 spi.open(0, 0)               #open spi port 0, device (CE)0
70
71 #MAX7219 initialization
72 sendSPI(MAX7219_TEST, 0x00)   #Finish test
73 sendSPI(MAX7219_DECODEMODE, 0x00) #Disable BCD mode
74 sendSPI(MAX7219_INTENSITY, 0x03) #Lowest intensity
75 sendSPI(MAX7219_SCANLIMIT, 0x07) #Scan all digits
76 sendSPI(MAX7219_SHUTDOWN, 0x01) #Turn on chip
77
78 for i in range(len(text)):
79     for j in range(0, 8):
80         sendSPI(j+1, cp437_font[text[i]][j])
81     sleep(1)
82
83 sendSPI(MAX7219_SHUTDOWN, 0x00) #Turn off chip
84 spi.close()
85 pause()

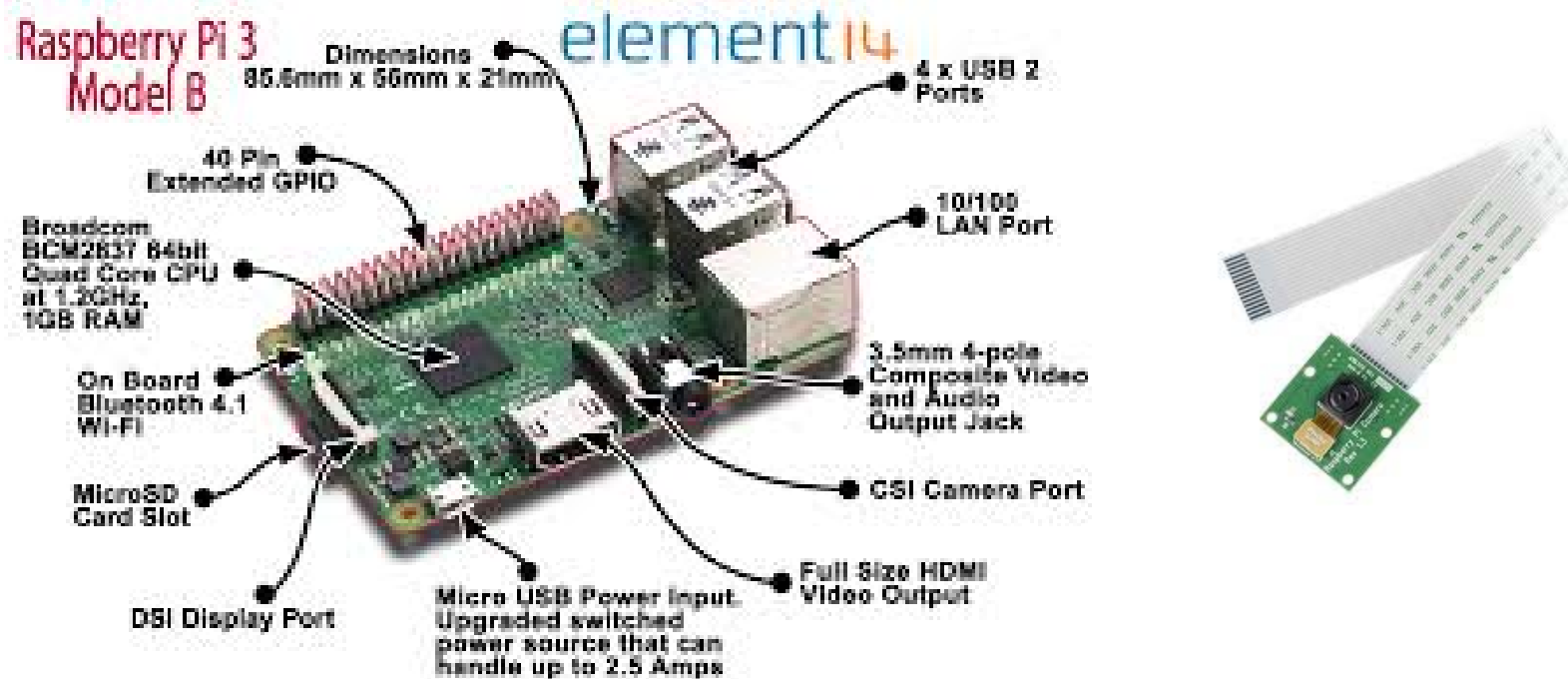
```



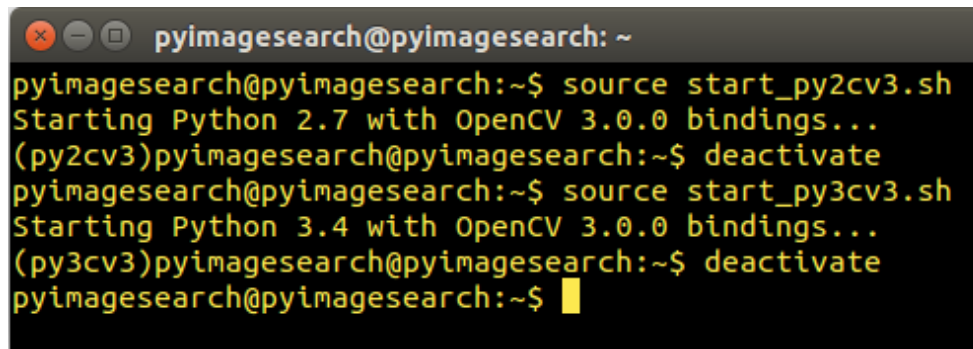
# Επεξεργασία εικόνων

# Raspberry Pi-Python-OpenCV

Η OpenCV έχει C++, C, Python και Java interfaces και υποστηρίζει Windows, Linux, Mac OS, iOS και Android. Η OpenCV έχει σχεδιαστεί για υπολογιστική αποδοτικότητα και κυρίως για εφαρμογές πραγματικού χρόνου.



- Το Raspberry Pi με την Pi camera χρησιμοποιούνται για εφαρμογές πραγματικού χρόνου. Το λειτουργικό σύστημα είναι το Raspbian Jessie στο οποίο έχει ενσωματωθεί η OpenCV.
- Μπορούν να δημιουργηθούν δύο εικονικά περιβάλλοντα λειτουργίας:
  - ❑ **(py2cv3)** –Python 2.7 και OpenCV 3. Ξεκινά εκτελώντας σε ένα Terminal Window:  
*source start\_py2cv3.sh*
  - ❑ **(py3cv3)** –Python 3.4 και OpenCV 3. Ξεκινά εκτελώντας σε ένα Terminal Window:  
*source start\_py3cv3.sh*

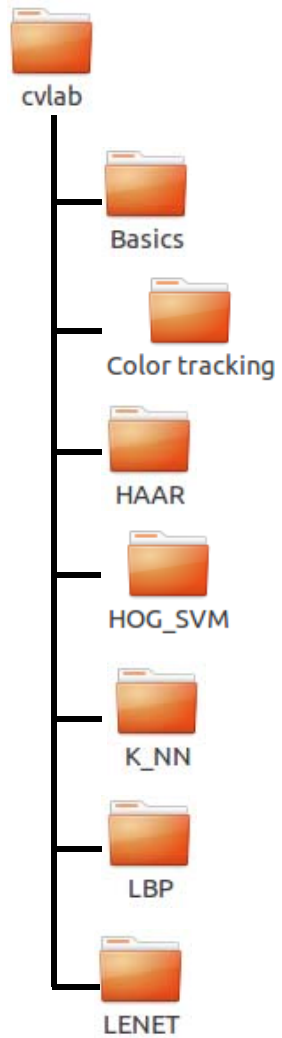


```
pyimagesearch@pyimagesearch: ~
pyimagesearch@pyimagesearch:~$ source start_py2cv3.sh
Starting Python 2.7 with OpenCV 3.0.0 bindings...
(py2cv3)pyimagesearch@pyimagesearch:~$ deactivate
pyimagesearch@pyimagesearch:~$ source start_py3cv3.sh
Starting Python 3.4 with OpenCV 3.0.0 bindings...
(py3cv3)pyimagesearch@pyimagesearch:~$ deactivate
pyimagesearch@pyimagesearch:~$ █
```

Terminal window calling: CTRL+ALT+T

Terminal window exiting: exit

- Φάκελος παραδειγμάτων:



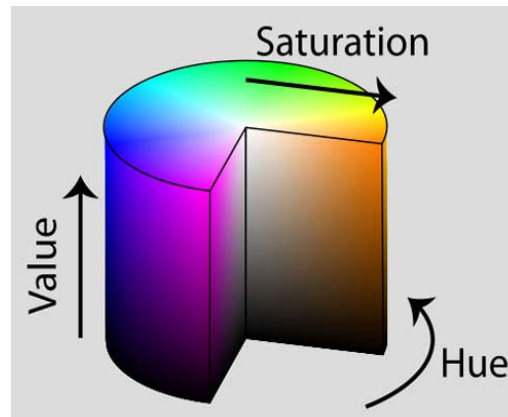
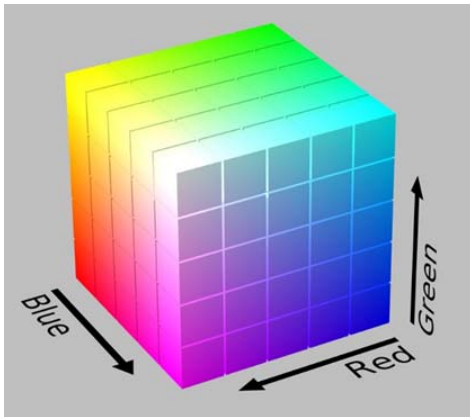
- Ανοίγουμε ένα Terminal Window
- Δημιουργούμε εικονικό περιβάλλον (py2cv3)
- Πηγαίνουμε στον επιθυμητό φάκελο με εντολή “cd”
- Εκτελούμε ένα πρόγραμμα rython
- Εξερχόμαστε του εικονικού περιβάλλοντος με “deactivate”
- Εξερχόμαστε του Terminal Window με “exit”

## Επεξεργασία εικόνων

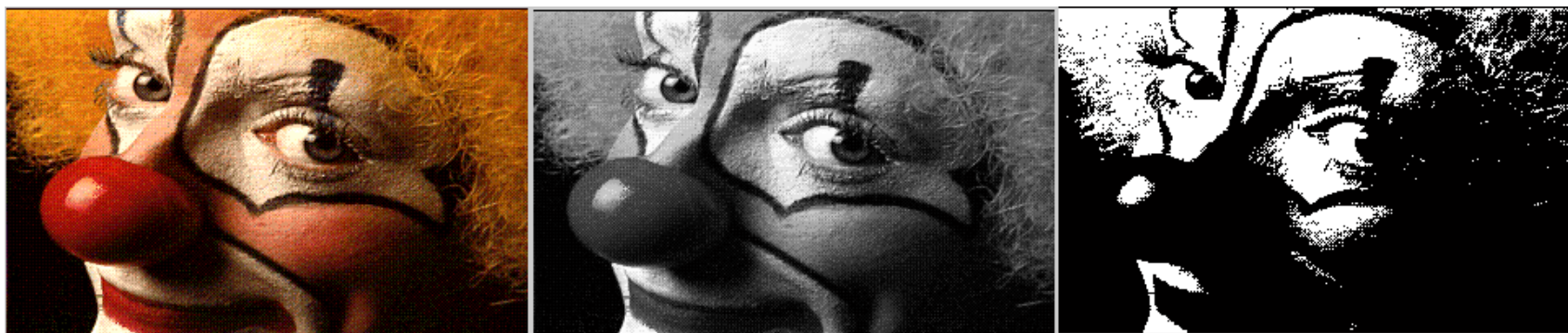
- Ο χρωματικός χώρος RGB υλοποιεί τις φωτεινότητες των εικονοστοιχείων σε τρία συστατικά: Κόκκινο - Πράσινο - Μπλε. Οι έγχρωμες εικόνες είναι συστοιχίες 3-D όπου κάθε μία αντιστοιχεί σε ένα έγχρωμο στοιχείο.

$$I(i, j) = [I_R(I, J), I_G(I, J), I_B(I, J)]$$

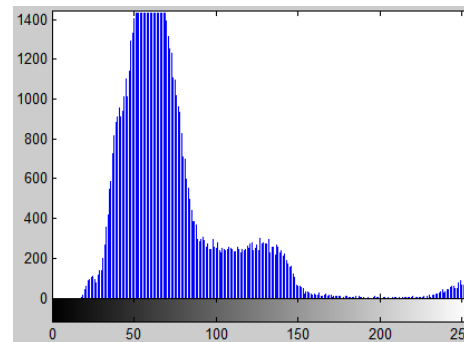
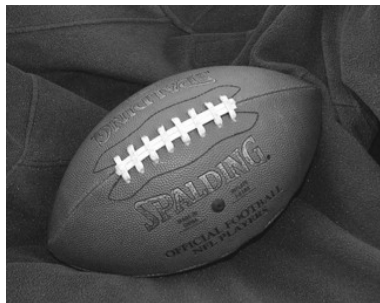
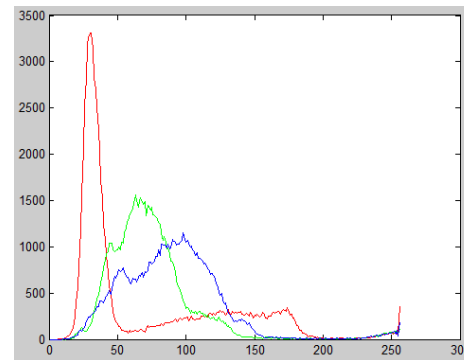
- Ενώ οι τιμές RGB είναι πολύ κατανοητές, το σύστημα RGB αποτυγχάνει να μιμηθεί τον τρόπο με τον οποίο οι άνθρωποι αντιλαμβάνονται το χρώμα. Μερικές φορές είναι δύσκολο να επεξεργαστούμε εικόνες RGB και γι' αυτόν το λόγο χρησιμοποιούμε το χρωματικό σύστημα HSV.
- Ο χρωματικός χώρος HSV εντάσσει τις φωτεινότητες των εικονοστοιχείων σε έναν κύλινδρο. Ο HSV διαχωρίζει τη φωτεινότητα ή την ένταση της εικόνας από το χρώμα ή τις πληροφορίες χρώματος.



- Οι γκρι εικόνες (εικόνες φωτεινότητας ή Gray Scale) είναι αριθμητικοί πίνακες 2-D με τιμές είτε uint8 στην περιοχή [0 255] είτε double στην περιοχή [0 1].
- Οι δυαδικές εικόνες (binary) είναι αριθμητικοί πίνακες 2-D με λογικές τιμές είτε 0 (για μαύρο χρώμα) είτε 1 (για λευκό χρώμα). Οι δυαδικές εικόνες χρησιμοποιούνται σε μεγάλο βαθμό στην όραση υπολογιστών, όπου πρέπει να εντοπιστούν αντικείμενα. Έτσι, οι έγχρωμες εικόνες μετατρέπονται σε εικόνες γκρι, οι οποίες με τη σειρά τους μετατρέπονται σε δυαδικές εικόνες με μία μόνο τιμή κατωφλίου στην περιοχή [0 255]. Η επιλογή αυτού του κατώτατου ορίου είναι δύσκολη και έχουν γίνει πολλές έρευνες σχετικά με τις βέλτιστες ρυθμίσεις. Ο αλγόριθμος του Otsu παρέχει μία από τις καλύτερες προσεγγίσεις σε αυτό.
- Συνήθως, μια εικόνα γίνεται πρώτα θολή (blurring) και μετά μετατρέπεται σε δυαδική. Η μέθοδος Gaussian blurring είναι πολύ αποτελεσματική για το σκοπό αυτό. Το θόλωμα εξομαλύνει την εικόνα, αλλά οι ακμές περιγράφουν ακόμα τα αντικείμενα.



Το **ιστόγραμμα** μιας εικόνας είναι η κατανομή φωτεινότητας ή η απεικόνιση της φωτεινότητας των εικονοστοιχείων της ή η συχνότητα εμφάνισής τους. Εάν οι τιμές ενός ιστογράμματος διαιρεθούν με το συνολικό αριθμό των εικονοστοιχείων, το αποτέλεσμα είναι το κανονικοποιημένο ιστόγραμμα που δείχνει την πιθανότητα εμφάνισης για κάθε φωτεινότητα.



Το **φιλτράρισμα** μιας εικόνας στοχεύει στην απομάκρυνση του θορύβου, την ανίχνευση ακμών (Sobel, Canny), την οξύτητα (sharpening) των ακμών (Laplacian), την εξομάλυνσή τους (smoothing) κ.λπ.



Smoothing με φίλτρο Gaussian, μάσκα (kernel) 5x5 και  $\sigma=5$ .



Διαδική εικόνα με λιγότερη υφή (texture).

- Το φίλτρο Sobel ή φίλτρο πρώτης παραγώγου εφαρμόζεται για την ανίχνευση ακμών. Η μάσκα για οριζόντιες ακμές (πρώτη παράγωγος κατά μήκος του άξονα  $y$ ) ή κάθετες ακμές (πρώτη παράγωγος κατά μήκος του άξονα  $x$ ) είναι:

$$\text{mask} = \begin{matrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{matrix}$$

$$\text{mask} = \begin{matrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{matrix}$$

- Ο αλγόριθμος Canny είναι πιο αποτελεσματικός από τον Sobel. Ορίζει δύο τιμές κατωφλίου και εξετάζει τρεις περιπτώσεις. Ένα εικονοστοιχείο ανήκει σε μια ακμή αν η φωτεινότητά του είναι μεγαλύτερη από την υψηλή τιμή κατωφλίου. Δεν ανήκει σε ακμή αν είναι μικρότερη από την χαμηλή τιμή κατωφλίου. Εάν η φωτεινότητά του είναι μεταξύ των δύο τιμών κατωφλίου, τότε ταξινομείται σύμφωνα με τα εικονοστοιχεία που το περιβάλλουν.

Sobel



Canny



# Load & Save

```
# python load_display_pi.py

import time
import cv2
from picamera.array import PiRGBArray
from picamera import PiCamera

cam=PiCamera()
rawCapture=PiRGBArray(cam)

time.sleep(0.1)

# capture a frame
cam.capture(rawCapture, format="bgr")
frame=rawCapture.array

# show some basic information
print("width: {} pixels".format(frame.shape[1]))
print("height: {} pixels".format(frame.shape[0]))
print("channels: {}".format(frame.shape[2]))

cv2.imshow("FRAME",frame)
cv2.waitKey(0)

# Save the image
cv2.imwrite("newimage.jpg", frame)
```

# Capture video

```
# python video_pi.py

import time
import cv2
from picamera.array import PiRGBArray
from picamera import PiCamera

cam=PiCamera()
cam.resolution = (640, 480)
cam.framerate = 32
rawCapture=PiRGBArray(cam, size=(640, 480))

time.sleep(0.1)

# capture video
while True:
    cam.capture(rawCapture, format="bgr", use_video_port=True)
    frame=rawCapture.array
    rawCapture.truncate(0)

    cv2.imshow("FRAME", frame)
    key = cv2.waitKey(1) & 0xFF
    if key == ord("q"):
        break
```

- Define the resolution and the frame rate
- Define size of memory

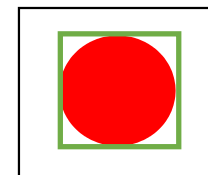
← Clear the current frame

← Exit with "q"

# Color tracking

Ανίχνευση χρώματος βρίσκοντας το περίγραμμα του αντικειμένου

- Λήψη εικόνας και μετατροπή της σε δυαδική εικόνα με βάση κατώφλια χρώματος
- Ορισμός μάσκας 37X37 pixels και απομάκρυνση θορύβου με “erosion & dilate”
- Blurring
- Εύρεση περιγραμμάτων αντικειμένου. Επιλογή του μεγαλύτερου. Σχεδίαση ορθογώνιου περιγράμματος



**Erosion & Dilation είναι μορφολογικοί χειρισμοί:**

**Erosion:** Ένα pixel σε μια εικόνα είναι “1” μόνο εάν όλα τα pixels της μάσκας γύρω από αυτό είναι “1”, διαφορετικά απαλείφεται (γίνεται “0”).

**Dilation:** Ένα pixel σε μια εικόνα είναι “1” εάν τουλάχιστον ένα pixel της μάσκας γύρω από αυτό είναι “1”.



# Ανίχνευση αντικειμένων κόκκινων ή μπλε

```
#capture image with picamera and track color objects
#using contouring - draw all contours
#rejecting noise with errosion & dilation

import numpy as np
import time
import cv2
from picamera.array import PiRGBArray
from picamera import PiCamera

cam=PiCamera()
cam.resolution=(640,480)
cam.framerate=32
rawCapture=PiRGBArray(cam,size=(640,480))

time.sleep(1)

key=raw_input('Press r=RED or b=BLUE:  ')

if key=="r":
    #red
    lower=np.array([160, 50, 50])
    upper=np.array([180, 255, 255])
elif key=="b":
    #dark blue
    #lower=np.array([110, 50, 50])
    #upper=np.array([130, 255, 255])
    #light blue
    lower=np.array([80, 50, 50])
    upper=np.array([110, 255, 255])
```

Εισάγουμε "r" ή "b"

Τιμές κατωφλίου για κόκκινο και μπλε  
χρώμα σε χρωματικό σύστημα HSV

Blurring

Περιγράμματα  
(Contours)

```
while True:
    cam.capture(rawCapture, format="bgr", use_video_port=True)
    frame=rawCapture.array
    rawCapture.truncate(0)

    hsv=cv2.cvtColor(frame,cv2.COLOR_BGR2HSV)
    mask=cv2.inRange(hsv, lower, upper)
    kernel=np.ones((9,9), np.uint8)
    erosion=cv2.erode(mask,kernel,iterations=1)
    dilation=cv2.dilate(erosion,kernel,iterations=1)
    mask=cv2.GaussianBlur(dilation, (3,3), 0)

    # find contours in the image
    (_, cnts, _) = cv2.findContours(mask.copy(), cv2.RETR_EXTERNAL,
                                    cv2.CHAIN_APPROX_SIMPLE)
```

Μετατροπή RGB σε HSV

Διαδική εικόνα

Μάσκα 9X9

Συρρίκνωση (erosion)

Διαστολή (dilation)

```

# draw all contours (-1) with green color line of width=2
#cv2.drawContours(frame,cnts, -1, (0, 255, 0), 2)

# check to see if any contours were found
if len(cnts) > 0:
    # sort the contours and find the largest one
    cnt = sorted(cnts, key = cv2.contourArea, reverse = True)[0]
    # compute the bounding box of contour and draw it
    rect = np.int32(cv2.boxPoints(cv2.minAreaRect(cnt)))
    cv2.drawContours(frame, [rect], -1, (0, 255, 0), 2)

# show the frame and the binary image
cv2.imshow("FRAME", frame)
cv2.imshow("MASK", mask)
time.sleep(0.025)

key=cv2.waitKey(5) & 0xFF
if key==ord("q"):
    break

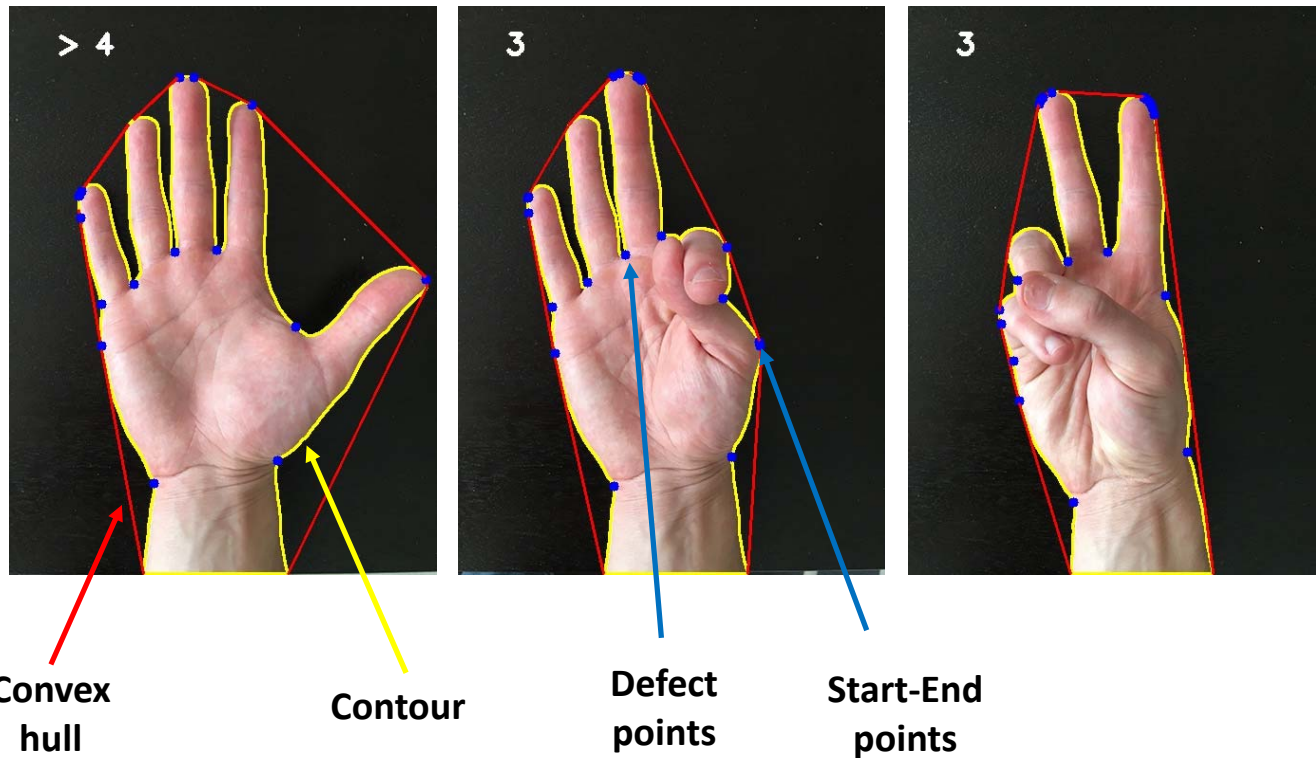
cv2.destroyAllWindows()

```

Σχεδίαση ορθογώνιου περιγράμματος  
γύρω από το αντικείμενο με τη  
μεγαλύτερη επιφάνεια

# Gesture recognition – Ανίχνευση χειρονομιών

Η αναγνώριση χειρονομιών μπορεί να γίνει με τη σχεδίαση του περιγράμματος του χεριού, του convex hull και την εύρεση των defect points.



Ταξινόμηση: Αριθμός defect points με γωνία  $< 90^\circ$

```
# python gestures_static.py
```

```
import cv2  
import math  
import numpy as np
```

```
# load image, convert to grayscale and blur it  
image = cv2.imread('images/hand_03.png')  
gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)  
gray = cv2.GaussianBlur(gray, (5, 5), 0)
```

```
# threshold the image, then perform a series of erosions +  
# dilations to remove any small regions of noise  
thresh = cv2.threshold(gray, 45, 255, cv2.THRESH_BINARY)[1]  
thresh = cv2.erode(thresh, None, iterations=2)  
thresh = cv2.dilate(thresh, None, iterations=2)
```

```
# get the largest contour  
cnts = cv2.findContours(thresh.copy(), cv2.RETR_EXTERNAL,  
                       cv2.CHAIN_APPROX_SIMPLE)  
cnts = cnts[1]  
cnt = max(cnts, key=cv2.contourArea)
```

```
# draw outermost contour and extreme points  
cv2.drawContours(image, [cnt], -1, (0, 255, 255), 2)
```

Μετατροπή RGB σε gray scale και blurring

Μετατροπή σε δυαδική

Απομάκρυνση θορύβου με erosion-dilation

Εύρεση περιγραμμάτων

Μέγιστο περίγραμμα

Ορθογώνιο περίγραμμα αντικειμένου

```

hull=cv2.convexHull(cnt, returnPoints=False)
defects=cv2.convexityDefects(cnt,hull)
counter=0
for i in range(defects.shape[0]):
    s, e, f, d = defects[i, 0]
    start = tuple(cnt[s][0])
    end = tuple(cnt[e][0])
    far = tuple(cnt[f][0])
    a=math.sqrt((end[0]-start[0])**2+(end[1]-start[1])**2)
    b=math.sqrt((far[0]-start[0])**2+(far[1]-start[1])**2)
    c=math.sqrt((end[0]-far[0])**2+(end[1]-far[1])**2)
    angle=math.acos((b**2+c**2-a**2)/(2*b*c))*57
    if angle < 90:
        counter += 1
        cv2.line(image, start, end, [0,0,255], 2)
        cv2.circle(image, far, 5, [255, 0, 0], -1)
if counter == 0:
    cv2.putText(image, "0", (50,50), cv2.FONT_HERSHEY_SIMPLEX, 1, (255,255,255),3)
if counter == 1:
    cv2.putText(image, "1", (50,50), cv2.FONT_HERSHEY_SIMPLEX, 1, (255,255,255),3)
elif counter == 2:
    cv2.putText(image, "2", (50,50), cv2.FONT_HERSHEY_SIMPLEX, 1, (255,255,255),3)
elif counter == 3:
    cv2.putText(image, "3", (50,50), cv2.FONT_HERSHEY_SIMPLEX, 1, (255,255,255),3)
elif counter == 4:
    cv2.putText(image, "4", (50,50), cv2.FONT_HERSHEY_SIMPLEX, 1, (255,255,255),3)
else:
    cv2.putText(image, "> 4", (50,50), cv2.FONT_HERSHEY_SIMPLEX, 1, (255,255,255),3)

# show the output image
cv2.imshow("Image", image)
cv2.waitKey(0)
cv2.destroyAllWindows()

```

Convex Hull και defect points

Υπολογισμός γωνιών στα defect points

Σχεδιασμός σημείων και γραμμών

Ταξινόμηση χειραψιών