

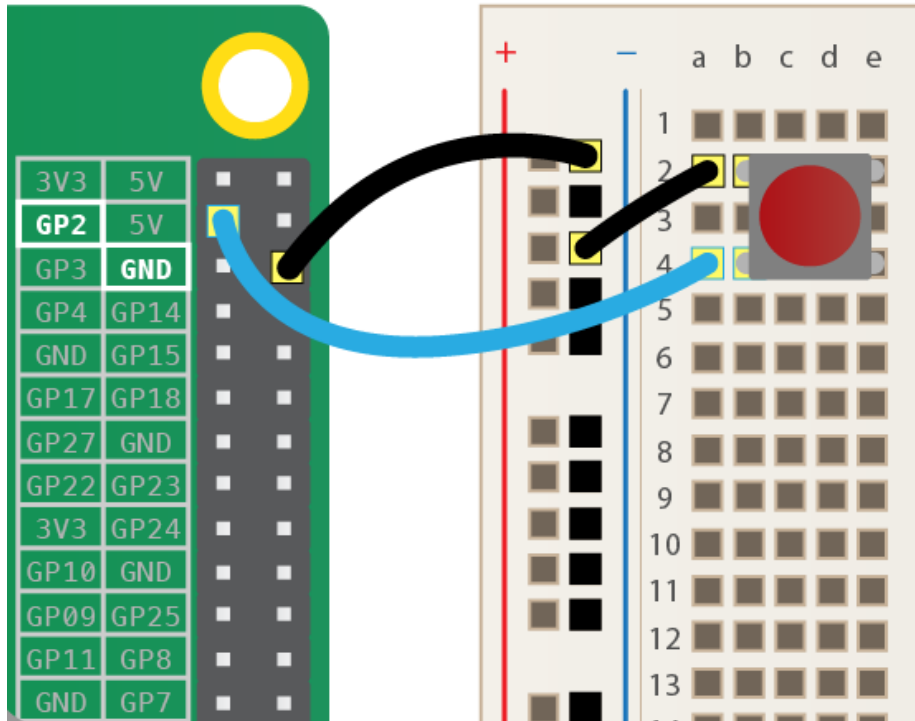
Προγραμματισμός με Python
στο Raspberry Pi

Button & MotionSensor

για το Code Club
Απρίλιος 2016

Σύλλογος Εκπαιδευτικών Πληροφορικής Χίου

Πιεζόμενα πλήκτρα (push buttons)



Πλήκτρο-διακόπτης-κουμπί:

- Συνδέεται σε οποιοδήποτε GPIO pin του Pi.
- Είσοδος αντί πληκτρολογίου.
- Ελέγχουμε εμείς τι θα συμβαίνει.

```
1 | from gpiozero import Button  
2 | b = Button(2)
```

Button



```
1  from gpiozero import LED, Button
2  from time import sleep
3  led = LED(16)
4  b = Button(12)
5  while True:
6      if b.is_pressed:
7          led.on()
8      else:
9          led.off()
10
11
12
```

Όταν και για όσο κρατάμε πατημένο ένα κουμπί, θέλουμε να ανάβει ένα led.

```
1  from gpiozero import LED, Button
2  from time import sleep
3  led = LED(16)
4  b = Button(12)
5  while True:
6      if b.is_pressed:
7          print ("Το ανάβω")           # Το led ανάβει και σβήνει όταν πρέπει, αλλά ...
8          led.on()
9      else:
10         print ("Το σβήνω")          # οι εντολές (ειδικά της else) εκτελούνται διαρκώς.
11         led.off()
12
```

Όταν και για όσο κρατάμε πατημένο ένα κουμπί, θέλουμε να ανάβει ένα led.

```
1  from gpiozero import LED, Button
2  from time import sleep
3  led = LED(16)
4  b = Button(12)
5  while True:
6      print("Σε αναμονή...")
7      b.wait_for_press()           # Περίμενε μέχρι να πατηθεί το κουμπί b.
8      print("Το ανάβω")
9      led.on()
10     b.wait_for_release()        # Περίμενε μέχρι να αφηθεί το κουμπί b.
11     led.off()
12     print("Το σβήνω")
```



Μια χρήσιμη παρένθεση

```
1 from gpiozero import LED
2 myled = LED(16)
3 myled.on()
```



```
1 from gpiozero import LED
2 myled = LED(16)
3 myled.on()
4 print ("Το led ανάβει")
```

```
1 from gpiozero import LED
2 myled = LED(16)
3 myled.on()
```

Δεν αφήνει το LED αναμμένο,
αφού η εκτέλεση τερματίζεται αμέσως

```
1 from gpiozero import LED
2 from signal import pause
3 myled = LED(16)
4 myled.on()
5 pause()
```

Το LED παραμένει αναμμένο,
αφού ο τερματισμός εμποδίζεται εσκεμμένα

Τέλος της χρήσιμης παρένθεσης

Όταν και για όσο κρατάμε πατημένο ένα κουμπί, θέλουμε να ανάβει ένα led.

```
1 from gpiozero import LED, Button
2 from signal import pause
3 led = LED(16)
4 b = Button(12)
5 def do1():
6     print ("To ανάβω")
7     led.on()
8 def do2():
9     print ("To σβήνω")
10    led.off()
11 b.when_pressed = do1
12 b.when_released = do2
13
14 pause()
```

*# Δημιουργούμε μια συνάρτηση που θα εκτελεί τις
ενέργειες που θέλουμε όταν πατηθεί το κουμπί.
Θα εκτελείται με το πάτημα του κουμπιού (βλ. γραμμή 11).*

*# Δημιουργούμε μια αντίστοιχη συνάρτηση για τις
ενέργειες που αφορούν στο άφημα του κουμπιού.*

*# Παρακολουθώ συνεχώς αν πατήθηκε ή αν αφέθηκε
το κουμπί και κάλεσε την αντίστοιχη συνάρτηση.*

Όταν φτάσεις εδώ, απλά περίμενε.



Όταν και για όσο κρατάμε πατημένο ένα κουμπί, θέλουμε να ανάβει ένα led.

```
1 from gpiozero import LED, Button
2 led = LED(16)
3 b = Button(12)
4 def do1():
5     print ("Το ανάβω")
6     led.on()
7 def do2():
8     print ("Το σβήνω")
9     led.off()
10 b.when_pressed = do1
11 b.when_released = do2
12
13 while True:
14     sleep(0.1)
```

Παρακολουθούθα συνεχώς αν πατήθηκε ή αν
αφέθηκε το κουμπί και δράσε αναλόγως.

Μια while, απλά για να μην τερματιστεί η εκτέλεση.
Η sleep έχει μπει για να πέσει η χρήση της CPU.



Button



Ένα κουμπί θα έχει διπλή λειτουργία: αν το LED ανάβει, θα το σβήνει, ενώ αν είναι σβηστό, θα το ανάβει.

```
1 from gpiozero import LED, Button
2 from signal import pause
3 led = LED(16)
4 b = Button(12)
5 # Ορίστε μια συνάρτηση που θα καλείται, όποτε πατιέται το κουμπί.
6     # Αν το led είναι αναμμένο, σβήσε το.
7     # Αν το led είναι σβηστό, ανάψε το.
8
9 # Δώστε την κατάλληλη εντολή, ώστε να
10 # παρακολουθείτε πότε πατήθηκε το Button.
11 # Σε αυτό το σημείο, απλά εμποδίστε τον τερματισμό του προγράμματος.
12
13 # Βλ. και ενδεικτική λύση 3 του Π1 (διαφάνεια 12).
```

Ένα κουμπί θα έχει διπλή λειτουργία: αν το LED ανάβει, θα το σβήνει, ενώ αν είναι σβηστό, θα το ανάβει.

```
1 from gpiozero import LED, Button
2 from time import sleep
3 led = LED(16)
4 b = Button(12)
5 def do():
6     if led.is_lit:
7         # ...
8     else:
9         # ...
10
11 b.when_pressed = do
12
13 while True:
14     sleep(0.1)
```


Button



Παιχνίδι με ένα LED που ελέγχει την αντίδραση του παίκτη.

```
1  ...                                     # Εισαγωγή των LED και Button, της sleep και της randint.
2  ...                                     # Δήλωση του led και του Button που θα χρησιμοποιηθούν.
3  score = 0                               # Αρχικοποίηση των μεταβλητών για το σκορ και τις ζωές.
4  lives = 5
5
6  def check():                             # Ορισμός της συνάρτησης που θα ελέγχει κατά πόσο
7      global score, lives                 # το κουμπί πατήθηκε με αναμμένο ή σβηστό led και
8      ...                                  # θα ενημερώνει κατάλληλα το σκορ ή τις ζωές.
9      ...                                  # Οι μεταβλητές score και lives δηλώνονται ως global,
10                                         # δηλαδή ισχύουν εντός αλλά και εκτός της συνάρτησης.
11  b.when_pressed = check                 # Παρακολουθούθα συνεχώς αν πατήθηκε το Button.
...                                       # Σε περίπτωση που το Button πατηθεί,
                                         # κάλεσε τη συνάρτηση check.
```

Παιχνίδι με ένα LED που ελέγχει την αντίδραση του παίκτη.

```
...
12 while ... :
13     ...
14     ...
15
16 print ("Τέλος παιχνιδιού")
17 print ("Τελικό σκορ:", score)
```

Όσο δεν συντρέχει λόγος να τελειώσει το παιχνίδι:
Άναψε το led για τυχαίο διάστημα, π.χ. για 0.5 έως 2.5 sec.
Σβήσε το led για τυχαίο διάστημα, π.χ. για 1 έως 2 sec.

Ενημέρωσε τον παίκτη για τις επιδόσεις του.

Button



Εφαρμογή που μετράει την ταχύτητα αντίδρασης του χρήστη.

```
1  ... # Εισαγωγή LED & Button, time & sleep και randint.
2  ... # Δήλωση του led και του Button που θα χρησιμοποιηθούν.
3  good = 0 # Αρχικοποίηση του μετρητή των συνεχόμενων
4           # ικανοποιητικών αντιδράσεων.
5  def check():
6      global steal # Ορισμός της συνάρτησης που θα ελέγχει αν το κουμπί
7      ... # πατήθηκε με σβηστό led, ώστε να ενημερώνει κατάλληλα
8      ... # μια λογική μεταβλητή steal.
9
10 b.when_pressed = check # Παρακολουθώ αν πατήθηκε το Button.
... # Σε περίπτωση που το Button πατηθεί,
    # κάλεσε τη συνάρτηση check.
```

Εφαρμογή που μετράει την ταχύτητα αντίδρασης του χρήστη.

```

...
11 while ... :
12     ...
13     ...
14     ...
15     ...
16     ...
17     ...
18     ...
19     ...
20 print ("Συγχαρητήρια!")

```

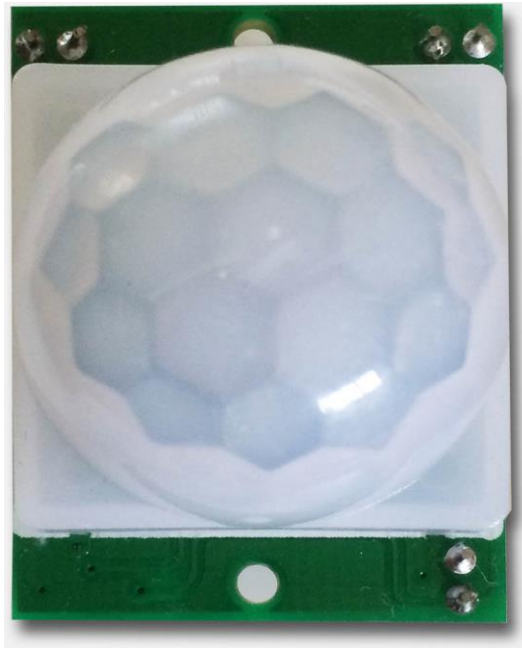
Όσο δεν ικανοποιείται η συνθήκη τερματισμού:

- # Άναψε το led μετά από τυχαίο διάστημα, π.χ. 1 έως 3 sec.
- # Κατάγραψε πότε άναψε (τη χρονική στιγμή).
- # Περίμενε να πατηθεί το κουμπί.
- # Κατάγραψε πότε πατήθηκε το κουμπί (τη χρονική στιγμή).
- # Υπολόγισε το χρόνο αντίδρασης και σβήσε το led.
- # Έλεγξε αν τυχόν ο χρήστης έκλεψε:
 - # Αν έκλεψε, μηδένισε το μετρητή.
 - # Αν όχι, εμφάνισε το χρόνο του και
 - # έλεγξε κατά πόσο ο χρόνος ήταν ικανοποιητικός.

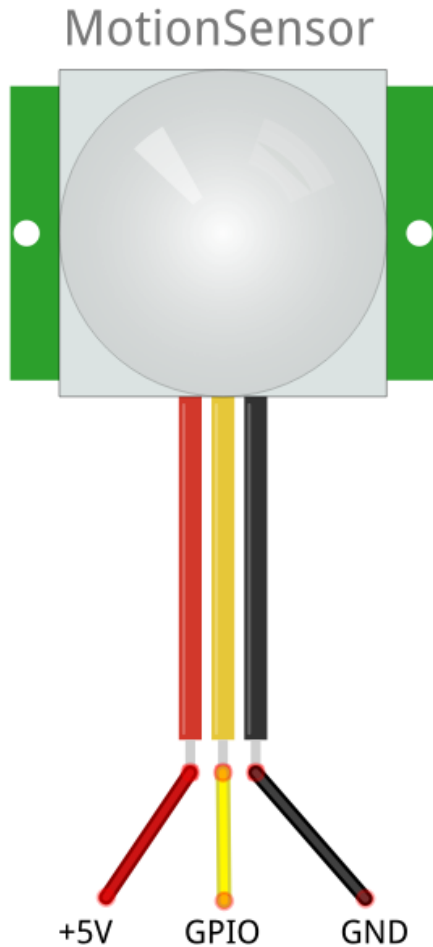
MotionSensor

Αισθητήρας ανίχνευσης κίνησης (PIR sensor)

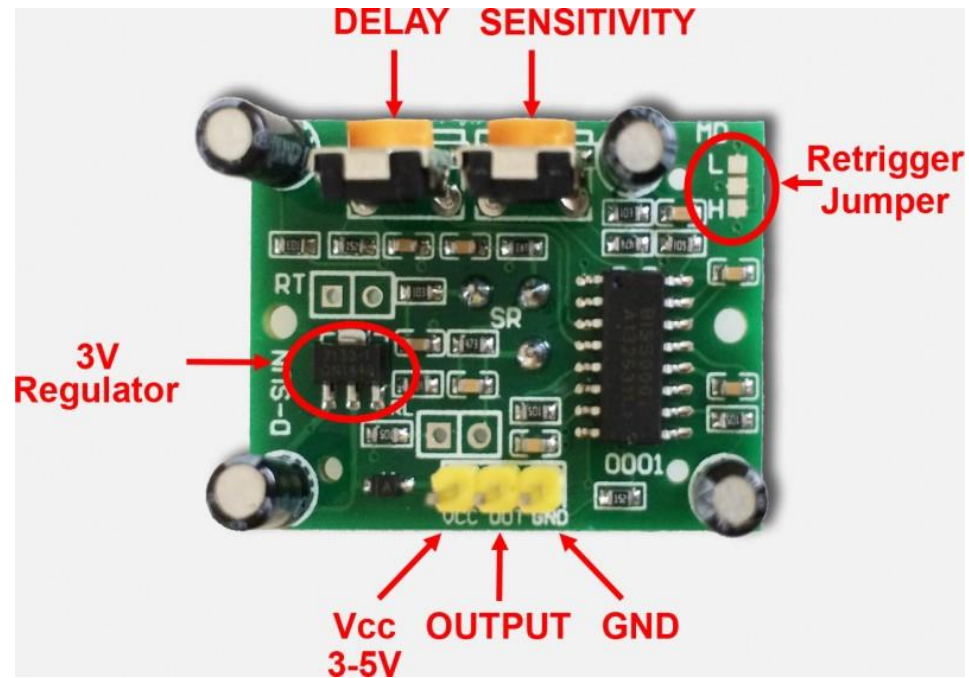
Αισθητήρας ανίχνευσης κίνησης (Passive InfraRed Sensor)



Αισθητήρας ανίχνευσης κίνησης (Passive InfraRed Sensor)



```
1 | from gpiozero import MotionSensor  
2 | p = MotionSensor(18)
```



MotionSensor



Όταν και για όσο ανιχνεύεται κίνηση, να ανάβει ένα led.

```
1 from gpiozero import LED, MotionSensor
2 from time import sleep
3 led = LED(16)
4 pir = MotionSensor(18)
5 while True:
6     print("Σε αναμονή...")
7     pir.wait_for_motion()
8     print("Ανιχνεύτηκε κίνηση")
9     led.on()
10    pir.wait_for_no_motion()
11    led.off()
12    print("Η κίνηση σταμάτησε")
```



Όταν και για όσο ανιχνεύεται κίνηση, να ανάβει ένα led.

```
1 from gpiozero import LED, MotionSensor
2 from signal import pause
3 led = LED(16)
4 pir = MotionSensor(18)
5 def alarm_on():
6     print ("Ανιχνεύτηκε κίνηση")
7     led.on()
8 def alarm_off():
9     print ("Η κίνηση σταμάτησε")
10    led.off()
11 pir.when_motion = alarm_on
12 pir.when_no_motion = alarm_off
13 pause()
```

Όταν και για όσο ανιχνεύεται κίνηση, να ανάβει ένα led.

A. Μετρήστε για πόση ώρα παραμένει ενεργή η έξοδος του PIR sensor.

```
1 from time import sleep, time           # Εισαγωγή της time.
2 t1 = time()                             # Καταγραφή της χρονικής στιγμής.
3 sleep(5)
4 t2 = time()                             # Καταγραφή της χρονικής στιγμής.
5 print ("Χρόνος: ", round(t2-t1, 2) )    # Εμφάνιση του χρόνου που πέρασε,
                                           # στρογγυλοποιημένου στα 2 δεκαδικά ψηφία.
```

B. Καταγράψτε και εμφανίστε την ημερομηνία και ώρα που ανιχνεύθηκε κίνηση.

```
1 from datetime import datetime          # Εισαγωγή της datetime.
2 t = datetime.now()                   # Καταγραφή της τρέχουσας ημ/νίας και ώρας.
3 print ("Ημερομηνία και ώρα: ", t )   # Εμφάνιση με τη μορφή 2016-03-25 21:10:40.984092.
```

Παράρτημα

Ενδεικτικές λύσεις

Ένα κουμπί θα έχει διπλή λειτουργία: αν το LED ανάβει, θα το σβήνει, ενώ αν είναι σβηστό, θα το ανάβει.

```
1 from gpiozero import LED, Button
2 from signal import pause
3 led = LED(16)
4 b = Button(12)
5 def do():
6     if led.is_lit:
7         print ("Το σβήνω")
8         led.off()
9     else:
10        print ("Το ανάβω")
11        led.on()
12 b.when_pressed = do
13 pause()
```

```
# Η συνάρτηση do θα καλείται όταν πατιέται
# το κουμπί (βλ. γραμμή 12) και εκτελεί τις
# ενέργειες που θέλουμε σε συνδυασμό με
# το αν το led ανάβει ή όχι.
```

```
# Παρακολουθώ πότε πατιέται το κουμπί.
# Η pause αποτρέπει τον τερματισμό της
# εκτέλεσης του προγράμματος.
```

Ένα κουμπί θα έχει διπλή λειτουργία: αν το LED ανάβει, θα το σβήνει, ενώ αν είναι σβηστό, θα το ανάβει.

```
1 from gpiozero import LED, Button
2 led = LED(16)
3 b = Button(12)
4 while True:
5     b.wait_for_press()
6     if led.is_lit == False:
7         led.on()
8     else:
9         led.off()
10    b.wait_for_release()
```


Ένα κουμπί θα έχει διπλή λειτουργία: αν το LED ανάβει, θα το σβήνει, ενώ αν είναι σβηστό, θα το ανάβει.

```
1 from gpiozero import LED, Button
2 led = LED(16)
3 b = Button(12)
4 state = 0
5 while True:
6     b.wait_for_press()
7     if state == 0:
8         led.on()
9         state = 1
10    else:
11        led.off()
12        state = 0
13    b.wait_for_release()
```

Έστω state η τρέχουσα κατάσταση του led.
Αρχικά το led είναι σβηστό (θεωρούμε state = 0).

Ενημέρωσε την τιμή της state.

Ενημέρωσε την τιμή της state.

Παιχνίδι με ένα LED που ελέγχει την αντίδραση του παίκτη.

```
1 from gpiozero import LED, Button
2 from time import sleep
3 from random import randint
4
5 led = LED(25)
6 b = Button(12)
7 score = 0
8 lives = 5
9 duration = 10
10 passed = 0
11
12 def check():
13     global score, lives
14     if led.is_lit:
15         print ("Μπράβο!")
16         score = score + 1
17     else:
18         print ("Ουπς...")
19         lives = lives - 1
20 b.when_pressed = check
...
```

Παιχνίδι με ένα LED που ελέγχει την αντίδραση του παίκτη.

```
...
21 while lives > 0 and passed <= duration:
22     led.on()
23     t1 = randint(50, 250) / 100
24     passed = passed + t1
25     sleep(t1)
26     led.off()
27     t2 = randint(100, 200) / 100
28     sleep(t2)
29
30 print ("Το παιχνίδι τελείωσε!")
31 print ("Τελικό σκορ:", score)
32 if lives > 0:
33     print ("Απέμεναν και", lives, "ζωές.")
```

Εφαρμογή που μετράει την ταχύτητα αντίδρασης του χρήστη.

```
1 from gpiozero import LED, Button
2 from time import sleep, time
3 from random import randint
4
5 led = LED(16)
6 b = Button(12)
7 good = 0
8
9 def check():
10     global steal
11     if not led.is_lit:
12         steal = True
13
14 b.when_pressed = check
15 while good < 3:
16     steal = False
17     print ("Έτοιμος;")
18     t = randint(10, 30) / 10
19     sleep(t)
20     led.on()
21     time1 = time()
22     b.wait_for_press()
23     time2 = time()
24     reaction_time = time2 - time1
25     led.off()
...

```

Εφαρμογή που μετράει την ταχύτητα αντίδρασης του χρήστη.

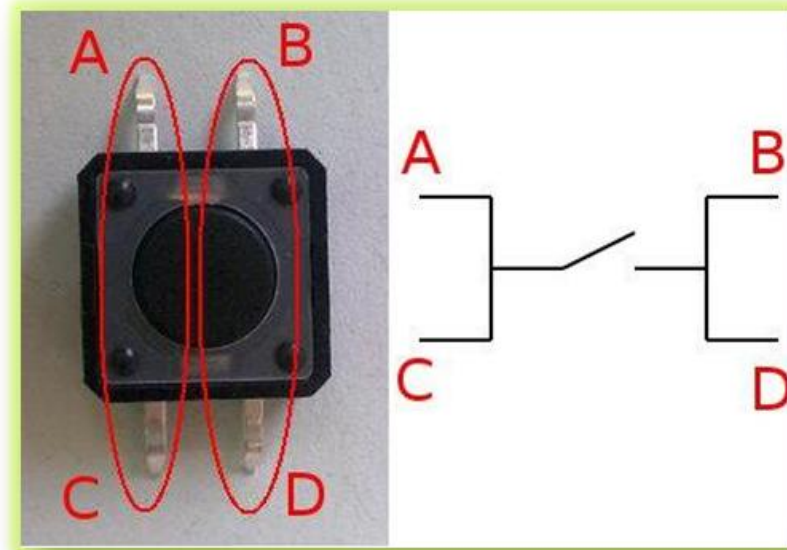
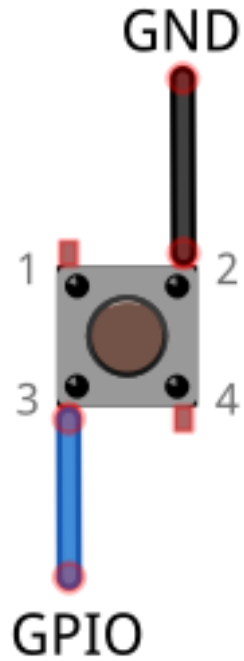
```
...
26     if steal:
27         print ("Μήπως κλέβεις;")
28         good = 0
29     else:
30         reaction_time = round (reaction_time, 3)
31         if reaction_time <= 0.25:
32             good = good + 1
33             print ("Μπράβο γρήγορε! Χρόνος:", reaction_time)
34         else:
35             good = 0
36             print ("Μπορείς και καλύτερα! Χρόνος:", reaction_time)
37
38     print ("\nΣυγχαρητήρια, τα κατάφερες!")
```

Παράρτημα Β

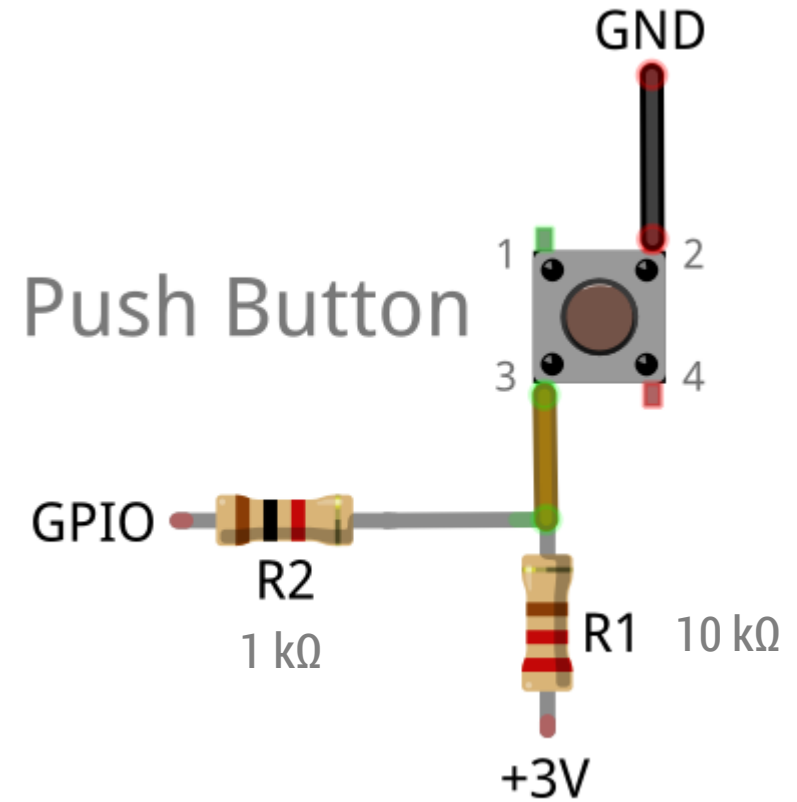
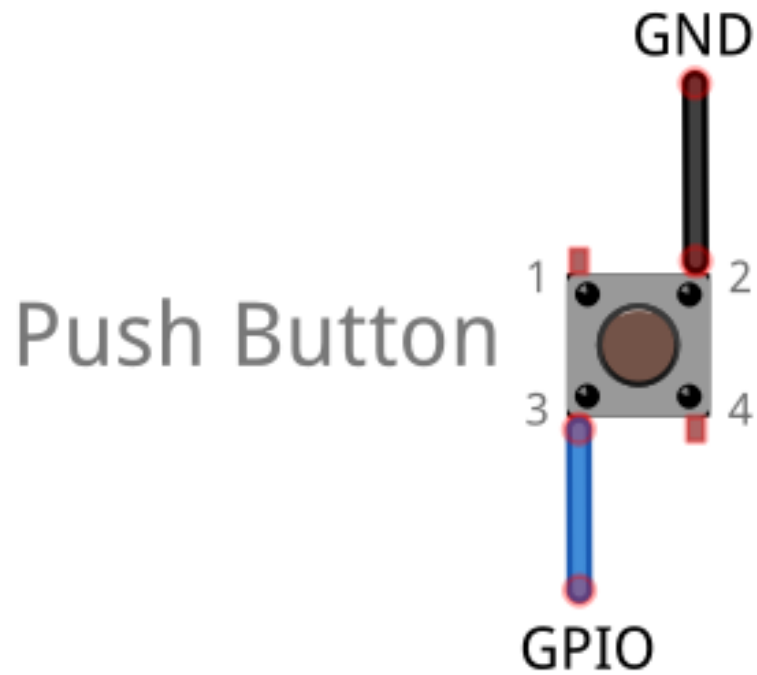
Περισσότερα στοιχεία για τα buttons

Διακόπτης - πλήκτρο

Push Button



Για μεγαλύτερη ασφάλεια





Το υλικό αυτό διατίθεται με άδεια Creative Commons Αναφορά Δημιουργού - Παρόμοια Διανομή 3.0.
<http://creativecommons.org/licenses/by-sa/3.0/gr/>

Η αναφορά σε αυτό θα πρέπει να γίνεται ως εξής:

*Προγραμματισμός με Python στο Raspberry Pi. Γιώργος Χατζηνικολάκης,
Σύλλογος Εκπαιδευτικών Πληροφορικής Χίου, Απρίλιος 2016.*