



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

Η αναφορά σε αυτό θα πρέπει να γίνεται ως εξής:

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

Εισαγωγή

Το παρόν υλικό αναπτύχθηκε προκειμένου να χρησιμοποιηθεί σε εκπαιδευτικό εργαστήριο με αντικείμενο τη γλώσσα Python και την χρήση της για τον προγραμματισμό του Raspberry Pi, στο πλαίσιο των Code Club που διοργανώνει ο Σύλλογος Εκπαιδευτικών Πληροφορικής Χίου.

Το απογευματινό εργαστήριο διεξήχθη την περίοδο Φεβρουαρίου - Απριλίου 2016 και περιελάμβανε 7 συναντήσεις των 2.5 ωρών, στις οποίες συμμετείχαν περισσότεροι από 20 μαθητές διαφόρων Λυκείων της Χίου.

Οι τρεις πρώτες συναντήσεις εστίασαν στην εξοικείωση των μαθητών με την Python 3, μέσω επιλεγμένων παραδειγμάτων από το βιβλίο «Pythonies» ¹ και σχετικών φύλλων εργασίας.

Στις επόμενες τρεις συναντήσεις οι μαθητές ήρθαν σε επαφή με το physical computing και προγραμμάτισαν τα 8 διαθέσιμα Pi σε ομάδες των 2-3 ατόμων, εφαρμόζοντας και επεκτείνοντας όσα είχαν δει μέχρι στιγμής στην Python. Λόγω του περιορισμένου χρόνου, υπήρχαν έτοιμες 8 βάσεις breadboard, με συνδεδεμένα τα απαραίτητα LEDs και Buttons και έτσι δεν δόθηκε ιδιαίτερη βαρύτητα στη συνδεσμολογία τους στο Pi.

Η τελευταία συνάντηση έδωσε τη δυνατότητα στους μαθητές να υλοποιήσουν ένα project της επιλογής τους, από μια συλλογή ιδεών, που αφορούσαν έργα είτε καθαρά σε Python, είτε στο Raspberry Pi.

Με βάση το παραπάνω περίγραμμα του εκπαιδευτικού εργαστηρίου, τα φύλλα εργασίας που αφορούν στο Raspberry Pi, και τα οποία συνοδεύονται από αντίστοιχες παρουσιάσεις, θεωρούν δεδομένο ένα στοιχειώδες επίπεδο γνώσης και χειρισμού των εντολών της Python.

Εκτός code club, το υλικό θα μπορούσε να χρησιμοποιηθεί είτε για κάποιο σχολικό project, είτε για πρακτική εξάσκηση στην Python, διαθέτοντας τα ήδη «στημένα» Pi με τα παρελκόμενά τους, λ.χ. στο μάθημα «Αρχές Προγραμματισμού Υπολογιστών» του Επαγγελματικού Λυκείου.

¹ Pythonies - Προγραμματισμός Μέσα Από Παραδείγματα στην Python: <u>http://pythonies.mysch.gr/</u>



Πίνακας περιεχομένων

0. To Raspberry Pi	5
Το Ρί με δυο λόγια	5
Προαπαιτούμενα	6
Διάγραμμα GPIO του Raspberry Pi	7
Χρήσιμες εντολές κονσόλας στο Raspbian	8
Οδηγός αναφοράς για μια γρήγορη επανάληψη στην Python	9
1. Λαμπάκια που ανάβουν και σβήνουν	10
LED	10
PWMLED	10
1ο Φύλλο εργασίας: Δραστηριότητες με LED και PWMLED	11
2. Όταν συμβεί κάτι, κάνε αυτό που θα σου πω	12
Button	12
MotionSensor	13
2ο φύλλο εργασίας: Δραστηριότητες με Buttons	14
3. Μεγαλύτερες δραστηριότητες	15
Μέρος Α: Παιχνίδια με LED και buttons	15
Μέρος Β: Δημιουργία ενός συναγερμού - ανιχνευτή κίνησης	16
4 . Πηγές	17
Πηγές και υλικό αναφοράς	17



Το Ρί με δυο λόγια

Το Raspberry Pi είναι ένας υπολογιστής σε μέγεθος πιστωτικής κάρτας, ο οποίος αναπτύσσεται στο Ηνωμένο Βασίλειο από τον μη κερδοσκοπικό οργανισμό Raspberry Pi Foundation για να προωθήσει την εκπαίδευση στην επιστήμη των υπολογιστών.

Η σημαντικότερη διαφορά του από έναν συνηθισμένο υπολογιστή δεν είναι ούτε το μικρό του μέγεθος, ούτε το χαμηλό του κόστος, αλλά το ότι διαθέτει 40 «ποδαράκια» γενικού σκοπού με τη γενική ονομασία GPIO (General Purpose Input Output) pins. Οι 40 αυτές υποδοχές επιτρέπουν τη σύνδεση ενός πολύ μεγάλου εύρους αισθητήρων και άλλων ηλεκτρονικών εξαρτημάτων και επομένως την άμεση αλληλεπίδρασή του με τον πραγματικό κόσμο και μάλιστα με χαμηλό κόστος.

Το επίσημο λειτουργικό σύστημα προς εγκατάσταση σε μια microSD κάρτα είναι το Raspbian, έκδοση του Debian Linux ειδικά προσαρμοσμένη για το Raspberry Pi. Διαθέτει πλήθος προεγκατεστημένων εφαρμογών, ενώ μπορεί να προγραμματιστεί σε διάφορες γλώσσες, από το Scratch μέχρι την Python, με την τελευταία να αποτελεί και την προτεινόμενη γλώσσα (εξ ου και η ονομασία Pi).

Το Ρi διατίθεται σε διάφορες εκδόσεις, με διαφορετικά τεχνικά χαρακτηριστικά και υποδοχές σύνδεσης περιφερειακών. Στο εργαστήριο χρησιμοποιήθηκαν 8 μονάδες Ρi έκδοσης 2 Model B, έχοντας τα εξής βασικά χαρακτηριστικά:

- τετραπύρηνο ARM επεξεργαστή στα 900 MHz
- 1GB μνήμη RAM
- υποδοχή Ethernet 100 Mbps
- 4 USB2.0 θύρες για σύνδεση περιφερειακών συσκευών
- 40 pins GPIO για τη σύνδεση αισθητήρων και άλλων συσκευών
- έξοδο HDMI για σύνδεση οθόνης
- ειδική υποδοχή για τη σύνδεση της επίσημης κάμερας του Raspberry Pi
- υποδοχή microSD για την προσθήκη κάρτας με το λειτουργικό σύστημα
- υποδοχή micro USB για την τροφοδοσία του με ρεύμα.



Κάτοψη του Raspberry Pi B+

(το Β+ είναι μοντέλο υποδεέστερο σε τεχνικά χαρακτηριστικά, αλλά οι θέσεις των στοιχείων παραμένουν ίδιες).



Προαπαιτούμενα

Εάν δεν χρησιμοποιήσουμε έτοιμες πλακέτες με εξαρτήματα που διατίθενται στο εμπόριο και απλά κουμπώνουν στο Pi, θα πρέπει να συνδέσουμε μόνοι μας τα εξαρτήματα στα GPIO pins με πολύ συγκεκριμένο τρόπο, αλλιώς είτε δεν θα λειτουργήσουν, είτε κινδυνεύουμε να κάψουμε κάποια από τα pins ή και το ίδιο το Pi. Αν τα στοιχεία που σκοπεύουμε να συνδέσουμε είναι σχετικά λίγα, μπορούμε να τα συνδέσουμε απευθείας πάνω στις σχετικές υποδοχές. Αν όμως θέλουμε μεγαλύτερη ευκολία και άνεση στις συνδέσεις, τότε προτείνεται η χρήση κάποιου breadboard.

Επίσης, εάν σκοπεύουμε να χρησιμοποιήσουμε το Ρi χωρίς οθόνη και πληκτρολόγιο, θα χρειαστεί να συνδεθούμε σε αυτό με κάποιο πρόγραμμα-πελάτη απομακρυσμένης πρόσβασης (τύπου VNC ή SSH). Στη δεύτερη περίπτωση θα εργαζόμαστε σε περιβάλλον γραμμής εντολών (terminal), οπότε είναι απαραίτητο να γνωρίζουμε κάποιες στοιχειώδεις εντολές Linux.

Ως προς τις απαιτούμενες γνώσεις σε Python, αρκεί να αντιλαμβάνεστε πώς να χρησιμοποιείτε τη δομή επιλογής, τη δομή επανάληψης και υποπρογράμματα (με ή χωρίς παραμέτρους), ενώ για ελάχιστες δραστηριότητες χρήσιμη είναι και η δομή δεδομένων της λίστας (list).

Υλικό

Στη συνέχεια αναφέρονται τα υλικά που χρησιμοποιήθηκαν για τις ανάγκες του εκπαιδευτικού εργαστηρίου, τα οποία θα χρειαστείτε αν θέλετε να ασχοληθείτε περαιτέρω και μόνοι σας.

- Raspberry Pi (ιδανικά το Pi 2 Model B).
- MicroSD κάρτα (>= 4 GB για το Jessie Lite, >=8 GB για το Jessie).
- Φορτιστής micro USB >= 1000 mA.
- LEDs που να λειτουργούν και στα 3 Volt και ισάριθμες αντιστάσεις 330 Ω.
- Αισθητήρας κίνησης (PIR sensor).
- Διακόπτες/πλήκτρα (push buttons) + δύο αντιστάσεις 1 kΩ και 10 kΩ ανά διακόπτη.
- Διάφορα μικρά καλώδια.
- Μια βάση breadboard για την ευκολότερη διασύνδεση των παραπάνω υλικών.

Επιπλέον (προαιρετικό) υλικό

- Κάμερα Raspberry Pi (ή μια οποιαδήποτε usb web camera).
- Μαγνητικός διακόπτης (Door magnetic switch).

Λογισμικό

- Η διανομή Raspbian Jessie² εγκατεστημένη στην SD κάρτα.
- Η γλώσσα Python 3 (προεγκατεστημένη στο Raspbian).
- Η βιβλιοθήκη Gpiozero³ της Python (προεγκατεστημένη στο Raspbian Jessie).
- SSH client⁴ στον υπολογιστή σας, για απομακρυσμένη πρόσβαση (εάν δεν σκοπεύετε να χρησιμοποιήσετε το Pi ως αυτόνομο σταθμό εργασίας).
- Text editor/IDE⁵ με υποστήριξη της Python3, για απομακρυσμένη χρήση με X-Forwarding.

² Το λειτουργικό σύστημα Raspbian: <u>https://www.raspberrypi.org/downloads/raspbian/</u>

 $^{^3}$ Gpiozero python interface για εύκολη αλληλεπίδραση με το Pi: <u>https://gpiozero.readthedocs.org/</u>

⁴ MobaXterm (προτεινόμενος SSH client για Windows): <u>http://mobaxterm.mobatek.net/</u>

⁵ Geany (προτεινόμενος code editor): <u>https://www.geany.org/</u>



Διάγραμμα GPIO του Raspberry pi

Για τις ανάγκες του εργαστηρίου, έχουν τοποθετηθεί στις βάσεις breadboard ορισμένα κοινά για όλους εξαρτήματα, τα οποία έχουν συνδεθεί σε συγκεκριμένα pins του Pi, όπως δείχνει το παρακάτω διάγραμμα.

Για παράδειγμα, στο pin 22 είναι συνδεδεμένο ένα button (Button1), ενώ στο pin 25 συνδέθηκε ένα led (LED5).

	3.3V	1	2	5V	
	GPIO2	3	4	5V	
	GPIO3	5	6	GND	
Buzzer	GPIO4	7	8	GPIO14	
	GND	9	10	GPIO15	
	GPIO17	11	12	GPIO18	
	GPIO27	13	14	GND	
Button1	GPIO22	15	16	GPIO23	
	3.3V	17	18	GPIO24	LED4
	GPIO10	19	20	GND	
	GPIO9	21	22	GPIO25	LED5
	GPIO11	23	24	GPIO8	
	GND	25	26	GPIO7	
	DNC	27	28	DNC	
	GPIO5	29	30	GND	
	GPIO6	31	32	GPIO12	Button2
	GPIO13	33	34	GND	
	GPIO19	35	36	GPIO16	LED1
	GPIO26	37	38	GPIO20	LED2
	GND	39	40	GPIO21	LED3

Περισσότερα: <u>http://pinout.xyz/</u>



Χρήσιμες εντολές κονσόλας στο Raspbian

Αν χρειαστεί να κάνουμε login, εισάγουμε όνομα χρήστη: **pi** και συνθηματικό: **raspberry** (όταν πληκτρολογούμε το password, δεν φαίνεται κάτι στην οθόνη).

<εντολή> help ή man <εντολή>	Εμφανίζει βοήθεια/πληροφορίες για την εντολή
pwd	Εμφανίζει τη διαδρομή του φακέλου στον οποίο βρισκόμαστε (<i>default: /home/pi</i>)
ls	Εμφανίζει τα περιεχόμενα του τρέχοντος φακέλου
mkdir <φάκελος>	Δημιουργεί ένα φάκελο με το αναφερόμενο όνομα
cd <όνομα φακέλου ή διαδρομή>	Μας μεταφέρει στον αναφερόμενο φάκελο. Π.χ.: cd videos ή cd /home/pi/videos
cp <αρχείο1> <διαδρομή ή αρχείο2>	Αντιγράφει το αρχείο1 στην αναφερόμενη διαδρομή ή φτιάχνει ένα αντίγραφο με το όνομα αρχείο2. Π.χ.: cp ex3.py /home/pi/exercises/ ή cp ex1.py ex2.py
nano	Ανοίγει τον επεξεργαστή κειμένου nano
nano <όνομα_αρχείου.py>	Ανοίγει το αναφερόμενο αρχείο στο nano
Συνδυασμός πλήκτρων Ctrl+O	Αποθηκεύει το ανοιχτό αρχείο (<i>αποθήκευση ως</i>)
Συνδυασμός πλήκτρων Ctrl+X	Κλείνει το ανοιχτό αρχείο και το nano
python3 <όνομα_αρχείου.py>	Εκτελεί το συγκεκριμένο αρχείο στην python 3
Συνδυασμός πλήκτρων Ctrl+C	Επιβάλλει τον τερματισμό του προγράμματος/script που εκτελείται
sudo <εντολή>	Εκτελεί την αναφερόμενη εντολή με δικαιώματα διαχειριστή
sudo reboot	Κάνει επανεκκίνηση του Ρί
sudo poweroff	Τερματίζει τη λειτουργία του Ρί
sudo apt-get update	Ενημερώνει τη λίστα των πακέτων του Raspbian
sudo apt-get upgrade	Αναβαθμίζει το Raspbian
sudo apt-get install <πακέτο>	Εγκαθιστά το αναφερόμενο πακέτο/εφαρμογή
sudo apt-get upgrade <πακέτο>	Αναβαθμίζει το αναφερόμενο πακέτο/εφαρμογή
sudo raspi-config	Ξεκινάει το κύριο διαχειριστικό εργαλείο του Raspbian (για αλλαγή password, hostname, προσθήκη γλωσσών, ακόμα και overclocking)
hostname	Εμφανίζει το hostname του Pi
hostname - I	Εμφανίζει την ΙΡ διεύθυνση του Ρί
geany &	Ανοίγει σε ξεχωριστό παράθυρο το πρόγραμμα επεξεργασίας κώδικα Geany (εφόσον είναι εγκατεστημένο)

Περισσότερα: https://www.raspberrypi.org/documentation/linux/





Οδηγός αναφοράς για μια γρήγορη επανάληψη στην Python

Πριν από την αναφορά στα αντικείμενα της βιβλιοθήκης gpiozero, γίνεται μια σύντομη ανασκόπηση των εντολών και λειτουργιών της Python που είναι απαραίτητες για την υλοποίηση των συνοδευτικών δραστηριοτήτων.

Αριθμητικοί τελεστές	+, -, *, /, **, //, %	
Συγκριτικοί τελεστές	>, >=, <, <= , ==, !=	
Λογικοί τελεστές	and, or, not	
Λογικές τιμές	True, False	
Εκχώρηση τιμής σε μεταβλητή	i = 2 ** 3 # ύψωση σε δύναμη (σχόλιο μιας γραμμής)	
Εμφάνιση μηνύματος στην οθόνη	print ("Διάλεξε ένα από τα", i, "Ρi μας!")	
Είσοδος από το πληκτοολόγιο	name - input("Dura de Véve: ")	\
	hame = input(noc or keve,)	
και απευθείας εκχωρηση	$b = int (input(1)old \chi povid y e$	ννηθηκες;))
Επιλογή με την if	if συνθήκη1:	
	# εντολές1	
	elif συνθήκη2:	
	# εντολές2	
	else:	
	#εντολές3	
	$\# \text{SVTO} \hat{s} c \Lambda$ (to té) oc the if $oo/7c$	εται μέσω της στοίνισης)
Επανάληψη με τη while	while συνθήκη:	
	# εντολές	
Επανάληψη με τη for	for μεταβλητή in range(αρχή,	τέλος , βήμα) :
	# εντολές	
	Παραδείγματα range:	
	range(10):	# 0, 1,, 9
	range(100, 0, -10):	# 100, 90,, 10
Λίστες	pins = [12, 20, 21, 19, 26]	# ορισμός λίστας
	pins[0] = 16	# αλλαγή του 1ου στοιχείου
	for i in range(len(pins)):	# διάσχιση στοιχείων λίστας
	print(pins[i], end=" ")	# εμφάνιση 16 20 21 19 26
Ορισμός συνάρτησης	def myfunction():	
	# εντολές	





Λαμπάκια που ανάβουν και σβήνουν LEDs που τα ελέγχουμε εμείς

LED

Μας επιτρέπει να διαχειριστούμε ένα απλό LED με λειτουργία on/off.

Α. Εισαγωγή στο πρόγραμμα:

from gpiozero import LED

Β. Δημιουργία αντικειμένου τύπου LED:

led1 = LED(20) # Ονομάσαμε led1 ένα αντικείμενο τύπου LED που συνδέσαμε στο pin 20.

Γ. Διαθέσιμες εντολές:

Μπορούμε να ανάψουμε ή να σβήσουμε ένα LED:

led1.on()	# Ανάβουμε το led1.
led1.off()	# Σβήνουμε το led1.

Μπορούμε να ελέγξουμε την κατάσταση ενός LED:

led1.is_lit # Λογική τιμή (True ή False) που περιγράφει την τρέχουσα κατάσταση του LED. # Π.χ. όταν το led1 ανάβει, η συνθήκη led1.is_lit έχει την τιμή True (αληθής).

PWMLED

Παρόμοια με τη LED, αλλά με την επιπλέον δυνατότητα να ορίζουμε τη φωτεινότητα του LED.

Α. Εισαγωγή στο πρόγραμμα:

from gpiozero import PWMLED

Β. Δημιουργία αντικειμένου τύπου PWMLED:

led2 = PWMLED(25) # Ονομάσαμε led2 ένα αντικείμενο τύπου PWMLED που συνδέσαμε στο pin 25.

Γ. Διαθέσιμες εντολές:

Μπορούμε να ανάψουμε ή να σβήσουμε ένα LED:

led2.on()	# Ανάβουμε το led2.
led2.off()	# Σβήνουμε το led2.

Μπορούμε να ελέγξουμε την κατάσταση ενός LED:

led2.is_lit	# Λογική τιμή (True ή False) που περιγράφει την τρέχουσα κατάσταση του LED.
	# Π.χ. όταν το led ανάβει σε πλήρη φωτεινότητα, η led2.is_lit έχει την τιμή True.

Μπορούμε να ελέγξουμε την τιμή της φωτεινότητας ή να δώσουμε μια πραγματική τιμή (float) μεταξύ 0 και 1 ώστε να ορίσουμε **άμεσα** τη φωτεινότητα ενός LED:

led2.value	# Π.χ. αν το led ανάβει πλήρως, η print (led2.value) θα εμφανίσει 1.
	# Π.χ. αν δώσουμε led2.value = 0.25 ορίζουμε τη φωτεινότητα στο 25%.



1° Φύλλο εργασίας: Δραστηριότητες με LED και PWMLED

1.1 Γράψτε ένα πρόγραμμα που θα αναβοσβήνει ένα LED, αφού πρώτα ρωτήσει το χρήστη πόσες φορές και για πόση διάρκεια θα ανάβει και θα σβήνει.

Το πρόγραμμα που ακολουθεί, υποθέτοντας ότι έχουμε συνδέσει ένα LED στο pin 16 του Raspberry Pi, αναβοσβήνει συνεχώς αυτό το LED για 1 sec, μέχρι να πατήσουμε το συνδυασμό πλήκτρων Ctrl+C. Βασιστείτε σε αυτό και τροποποιήστε το κατάλληλα για να πετύχετε το σκοπό σας.

```
from gpiozero import LED
from time import sleep
led = LED(16)
while True:
    led.on()
    sleep(1)
    led.off()
    sleep(1)
```

1.2 Κάντε ένα LED να μοιάζει με ένα κερί που τρεμοπαίζει (το led ανάβει σε τυχαία ένταση και για τυχαίο χρονικό διάστημα).

Αντί της LED, εισάγετε στο πρόγραμμά σας την **PWMLED** από την gpiozero, η οποία μας επιτρέπει να ορίζουμε πόσο έντονα θα ανάβει το led, μέσω της ιδιότητας **value**. Έτσι αξιοποιώντας εντολές όπως η παρακάτω, μπορούμε να ορίσουμε άμεσα και ακριβώς τη φωτεινότητα ενός led σε οποιαδήποτε πραγματική τιμή από 0 μέχρι και 1.

led.value = 0.25 # ορίζει τη φωτεινότητα του led σε 25%

Θα χρειαστείτε επίσης τη συνάρτηση **randint** της βιβλιοθήκης random, η οποία παίρνει ως παράμετρο ένα εύρος **ακέραιων** αριθμών και παράγει έναν τυχαίο ακέραιο εντός του εύρους αυτού. Π.χ.:

```
from random import randint
z = randint(1,6)
print (z)  # εμφανίζει μια τυχαία ζαριά
```

Για ένα πιο ρεαλιστικό εφέ, προτείνουμε η φωτεινότητα να κυμαίνεται μεταξύ 10% και 100% και να αλλάζει κάθε 0 μέχρι 0.1 sec, αλλά μπορείτε να πειραματιστείτε ελεύθερα με ό,τι τιμές θέλετε.

1.3 Αξιοποιήστε ξανά την PWMLED για να κάνετε ένα led να αναβοσβήνει αργά, ώστε να μοιάζει σαν να αναπνέει, δηλαδή να αυξάνει σταδιακά τη φωτεινότητά του μέχρι να ανάψει πλήρως, έπειτα να μειώνει σταδιακά τη φωτεινότητά του μέχρι να σβήσει για λίγο, και πάλι από την αρχή. Επεκτείνετε τον κώδικά σας, ώστε ένα ακόμα led να αναβοσβήνει αντίστροφα από το άλλο, δηλαδή όταν π.χ. η φωτεινότητα του ενός led είναι στο 10%, του άλλου να είναι στο 90%.

Γενική συμβουλή για τις δραστηριότητες

Μια καλή πρακτική, χρήσιμη και για διαγνωστικούς λόγους (όταν π.χ. κάτι δεν λειτουργεί όπως θα θέλαμε και προσπαθούμε να εντοπίσουμε τι φταίει) είναι να εισάγουμε, σε επιλεγμένα σημεία, εντολές εκτύπωσης στην οθόνη. Παραδείγματα:

```
- μια εντολή του τύπου led.on() μπορεί να συνδυαστεί με μια αντίστοιχη print ("το led ανάβει"),
```

```
- μια εντολή led.value = z με μια print ("φωτεινότητα του led:", z),
```

```
- στο τέλος μιας επαναληπτικής δομής while βάζουμε μια print ("τέλος επανάληψης").
```







Όταν συμβεί κάτι, κάνε αυτό που θα σου πω Είσοδος στα προγράμματά μας από συνδεμένα πλήκτρα ή αισθητήρες

Button

Μας επιτρέπει να διαχειριστούμε ένα απλό πλήκτρο-διακόπτη (push button).

Α. Εισαγωγή στο πρόγραμμα:

from gpiozero import Button

Β. Δημιουργία αντικειμένου τύπου Button:

 $\mathbf{b} = Button(12)$ # Ονομάσαμε b ένα αντικείμενο τύπου Button που συνδέσαμε στο pin 12.

Γ. Διαθέσιμες εντολές:

Μπορούμε να πούμε στο πρόγραμμά μας να περιμένει μέχρι να πατηθεί/αφεθεί ένα button:

```
b.wait for press() # Το πρόγραμμά μας θα περιμένει μέχρι να πατηθεί το κουμπί b.
b.wait for release() # Το πρόγραμμά μας θα περιμένει μέχρι να αφεθεί το κουμπί b.
```

Μπορούμε να ελέγξουμε την κατάσταση ενός button:

b.is pressed # Λογική τιμή (True ή False) που περιγράφει την τρέχουσα κατάσταση του button.

Μπορούμε να παρακολουθούμε συνεχώς εάν πατήθηκε ή αφέθηκε ένα κουμπί και να ορίσουμε τι θα γίνεται κάθε φορά που συμβαίνει αυτό το γεγονός:

b.when_pressed	<pre># Παρακολουθεί συνεχώς την κατάσταση του κουμπιού b. Από μόνη της αυτή η # εντολή δεν κάνει κάτι. Μπορούμε όμως να τη συνδυάσουμε με μια # συνάρτηση που θα κάνει αυτά που επιθυμούμε. Π.χ. έστω η συνάρτηση: def lights_on(): print("Το κουμπί πατήθηκε, ανάβω τα 2 led!") led1.on() led2.on() # Η παρακάτω εντολή θα καλεί τη συνάρτηση lights_on όποτε πατιέται το κουμπί b. b.when_pressed = lights_on</pre>
b.when_released	<pre># Αντιστοίχως, όποτε το κουμπί b αφεθεί, μπορούμε να καλούμε μια συνάρτηση # π.χ. lights_off με παρόμοιο τρόπο: def lights_off(): print("Το κουμπί αφέθηκε, τα led σβήνουν!") led1.off() led2.off() # Η παρακάτω εντολή θα καλεί και θα εκτελεί τη συνάρτηση lights_off όποτε # πατιέται το κουμπί b. b.when released = lights off</pre>

Μια χρήσιμη παρατήρηση

Οι εντολές ανίχνευσης συμβάντων when_pressed και when_released αρκεί να εκτελεστούν μια φορά και παρακολουθούν συνεχώς το αντικείμενο στο οποίο αφορούν. Αυτό σημαίνει ότι δεν χρειάζεται να τις βάλουμε εντός κάποιας επανάληψης. Έτσι, μπορούμε να αποτρέψουμε τον τερματισμό της εκτέλεσης του προγράμματος τοποθετώντας την εντολή pause() από τη βιβιοθήκη signal στο τέλος του. Π.χ.:

from signal import pause	# Εισάγουμε την pause.
	# Εντολές (και δηλώσεις συναρτήσεων)
b .when_pressed = lights_on	# Παρακολουθούμε πότε πατιέται το b.
	# Κι άλλες εντολές
pause()	# Η pause αποτελεί την τελευταία εντολή.





MotionSensor

Αισθητήρας ανίχνευσης κίνησης (Passive IR sensor).

Α. Εισαγωγή στο πρόγραμμα:

from gpiozero import MotionSensor

Β. Δημιουργία αντικειμένου τύπου MotionSensor:

p = MotionSensor(18) # Δηλώνουμε έναν αισθητήρα με όνομα p συνδεδεμένο στο pin 18.

Γ. Διαθέσιμες εντολές:

Μπορούμε να πούμε στο πρόγραμμά μας να περιμένει μέχρι να ανιχνευτεί/σταματήσει η κίνηση:

```
p.wait_for_motion() # Το πρόγραμμά μας θα περιμένει μέχρι ο αισθητήρας p να ανιχνεύσει κίνηση.
p.wait_for_no_motion() # Το πρόγραμμά μας θα περιμένει μέχρι να σταματήσει η κίνηση.
```

Μπορούμε να ελέγξουμε την κατάσταση ενός αισθητήρα:

p.motion_detected # Λογική τιμή που περιγράφει την τρέχουσα κατάσταση του αισθητήρα.

Μπορούμε να παρακολουθούμε συνεχώς εάν ανιχνεύεται κίνηση ή αν αυτή η κίνηση παύει να υφίσταται και να ορίσουμε τι θα γίνεται κάθε φορά που συμβαίνουν αυτά τα γεγονότα:

p.when_motion	# Παρακολουθεί συνεχώς την κατάσταση του αισθητήρα p. Όπως και στην # περίπτωση του button, συνδυάζουμε την εντολή αυτή με μια συνάρτηση, η οποία # μπορεί (μεταξύ άλλων) να καλεί και άλλες συναρτήσεις. Π.χ.:
	def alarm(): print("Ανίχνευση κίνησης!") take_picture() play_alarm_sound()
	# Η παρακάτω εντολή θα καλεί και θα εκτελεί τη συνάρτηση alarm όποτε # ανιχνεύεται κίνηση.
	p.when_motion = alarm
p.when_no_motion	# Αντιστοίχως, όποτε σταματάει η κίνηση, καλούμε μια άλλη συνάρτηση. Π.χ.:
	def standby(): print("Σε αναμονή") led1.on()
	# Η παρακάτω εντολή θα καλεί και θα εκτελεί τη συνάρτηση standby όποτε # σταματάει η κίνηση που ενεργοποίησε τον αισθητήρα. p.when no motion = standby

Δ. Σύνδεση ενός PIR sensor στο Pi:

- Οι αισθητήρες κίνησης διαθέτουν συνήθως 3 ποδαράκια τα οποία συνδέονται στο Pi ως εξής:
 VCC (ή +5V): σε ένα από τα (2) pins του Pi που παρέχουν σταθερά 5 Volt.
 - GND: σε οποιοδήποτε pin του Pi με την ένδειξη GND.
 - ΟUT: σε οποιοδήποτε GPIO pin του Pi.
- Ορισμένοι αισθητήρες διαθέτουν 2 μικρούς ρυθμιστές (ποτενσιόμετρα):
 - Sx (Sensitivity): καθορίζει την ευαισθησία του αισθητήρα
 - Τx (Time): καθορίζει το χρόνο που παραμένει ενεργή η έξοδος του αισθητήρα από τη στιγμή που ανιχνεύεται κίνηση (συνήθως από 2.5 sec μέχρι μερικά λεπτά).





2° φύλλο εργασίας: Δραστηριότητες με Buttons

2.1 Γράψτε ένα πρόγραμμα που το πάτημα ενός διακόπτη επιτελεί διπλή λειτουργία για ένα LED (αν το LED είναι σβηστό το ανάβει, ενώ αν ανάβει το σβήνει).

- Αξιοποιήστε, είτε τις wait_for_press() και wait_for_release(), είτε εναλλακτικά φτιάξτε μια συνάρτηση και συνδέστε τη με το γεγονός πατήματος του κουμπιού (when_pressed).
- Χρησιμοποιήστε την ιδιότητα Led.is_Lit που έχει την τιμή True ή False ανάλογα με την κατάσταση του led, ώστε να συμπεραίνετε ποια ενέργεια πρέπει να γίνει.

2.2 Φτιάξτε παιχνίδι στο οποίο ένας παίκτης συγκεντρώνει πόντους όταν αντιδράει σωστά και χάνει ζωές όταν κάνει λάθος. Συγκεκριμένα, ένα led ανάβει μετά από τυχαίο χρονικό διάστημα και για τυχαίο χρονικό διάστημα. Όσες πιο πολλές φορές προλάβει ο παίκτης να πατήσει το κουμπί του με το led αναμμένο, τόσο περισσότερους πόντους μαζεύει, ενώ αντιθέτως χάνει μια ζωή για κάθε πάτημα με το led σβηστό. Το παιχνίδι τελειώνει όταν ο παίκτης δεν έχει άλλες ζωές.

Κάντε το παιχνίδι πιο ανταγωνιστικό, μετρώντας το συνολικό χρόνο που το led μένει αναμμένο (αυτός είναι και ο χρόνος στον οποίο δίνεται πρακτικά στον παίκτη η ευκαιρία να σκοράρει), ώστε το παιχνίδι να τερματίζεται και όταν παρέλθει αυτός ο χρόνος.

- Χρησιμοποιήστε τη δυνατότητα συνεχούς παρακολούθησης της κατάστασης του κουμπιού που παρέχει η when_pressed, ώστε να καλείτε μια συνάρτηση (την οποία και θα κατασκευάσετε) για να ελέγχετε κατά πόσο το γεγονός του πατήματος του κουμπιού συνέβη με το led αναμμένο ή όχι.
- Η συνάρτηση θα χρειαστεί να ενημερώνει το σκορ και τις ζωές του παίκτη και να επικοινωνεί με το υπόλοιπο πρόγραμμα, ώστε λ.χ. το παιχνίδι να σταματάει όταν τελειώνουν οι ζωές. Για αυτό το λόγο, οι μεταβλητές της συνάρτησης που θέλουμε να επικοινωνούν με το κυρίως πρόγραμμα δεν αρκεί να έχουν το ίδιο όνομα, αλλά πρέπει και να δηλώνονται σε αυτή με τον όρο global (καθολικές).
- Για την καταμέτρηση του χρόνου, αρκεί να αθροίζετε τα τυχαία διαστήματα που ανάβει το led.

2.3 Φτιάξτε εφαρμογή η οποία μετράει επαναληπτικά την ταχύτητα αντίδρασης του χρήστη. Συγκεκριμένα, ο χρόνος αρχίζει να μετράει από την τυχαία χρονική στιγμή που ένα led ανάβει, μέχρι ο χρήστης να πατήσει το κουμπί, οπότε το led σβήνει και η ταχύτητά του εμφανίζεται στην οθόνη. Αυτή η διαδικασία επαναλαμβάνεται μέχρι π.χ. να καταγραφούν τρεις συνεχόμενες αντιδράσεις χρόνου μικρότερου των 0.2 sec.

- Η εντολή wait_for_press() για το button θα επιτρέψει στο πρόγραμμά σας να περιμένει (υπομονετικά ⁽ⁱⁱⁱⁱ⁾) μέχρι να πατηθεί το κουμπί.
- Αξιοποιήστε την time() από τη βιβλιοθήκη time, για να αποθηκεύετε σε δυο μεταβλητές τη χρονική στιγμή που άναψε το led και τη χρονική στιγμή που πατήθηκε το πλήκτρο αντίστοιχα, ώστε να υπολογίζετε τη διαφορά τους.
- Μπορείτε (προαιρετικά) να στρογγυλοποιήσετε την ταχύτητα αντίδρασης του χρήστη (πρόκειται για την τιμή μιας μεταβλητής τύπου float) σε όσα δεκαδικά ψηφία θέλετε με την round. Π.χ. av t = 0.477762, η εντολή t = round(t, 2) θα εκχωρήσει στην t την τιμή 0.48, ενώ η εντολή t = round(t, 1) θα εκχωρήσει στην t την τιμή 0.5.
- Η εφαρμογή μας αυτή τη στιγμή παρουσιάζει ένα μικρό αλλά σημαντικό πρόβλημα: ο χρήστης μπορεί να κλέψει, κρατώντας μόνιμα πατημένο το κουμπί του, οπότε θα φαντάζει ταχύτατος! Διορθώστε το πρόβλημα, παρακολουθώντας το πάτημα του κουμπιού συνεχώς, οπότε αν το κουμπί πατηθεί πριν ανάψει το led, ο χρήστης να «τιμωρείται», π.χ. με ακύρωση της προσπάθειάς του ή και μηδενισμό των μέχρι στιγμής επιτυχημένων προσπαθειών του.





3 Μεγαλύτερες δραστηριότητες Ιδέες για projects

Μέρος Α: Παιχνίδια με LED και buttons



3.1 [QUICK REACTION MULTI-PLAYER GAME]

Φτιάξτε ένα παιχνίδι δύο (ή περισσότερων) παικτών στο οποίο κερδίζει ο παίκτης με την ταχύτερη αντίδραση. Ένα led ανάβει μετά από τυχαίο χρονικό διάστημα και ο πρώτος που πατάει το κουμπί του κερδίζει. Ξεκινήστε φτιάχνοντας μια απλή εκδοχή ενός γύρου. Στη συνέχεια, μπορείτε να επεκτείνετε το παιχνίδι, διαβάζοντας τα ονόματα των παικτών, αυξάνοντας τον αριθμό των γύρων και εμφανίζοντας το σκορ κάθε παίκτη στην οθόνη. Φυσικά, καλό θα ήταν να προβλέψετε τι θα συμβαίνει όταν κάποιος παίκτης επιχειρεί να κλέψει, πατώντας το κουμπί του πριν ανάψει το led.

3.2 [LED IN THE MIDDLE GAME]

Φτιάξτε ένα παιχνίδι με πολλά LEDs που ανάβουν σε σειρά το ένα μετά το άλλο. Πιο συγκεκριμένα, μόνο ένα led ανάβει κάθε φορά και μοιάζει σαν να πηγαινοέρχεται μπρος πίσω από άκρη σε άκρη (κάτι σαν τη φωτεινή μπάρα που είχε ο KITT, το θρυλικό αυτοκίνητο του Ιππότη της Ασφάλτου). Σκοπός του παιχνιδιού είναι ο παίκτης να πατήσει το κουμπί όταν ανάβει το μεσαίο LED. Για κάθε επιτυχημένη (ή και αποτυχημένη) προσπάθεια η ταχύτητα του εφέ αυξάνεται και το παιχνίδι δυσκολεύει. Μπορούν να προστίθενται πόντοι για κάθε επιτυχημένη.

3.3 [LADDER GAME]

Φτιάξτε ένα παιχνίδι με μια σειρά από LEDs (όσο πιο πολλά, τόσο το καλύτερο) που θυμίζει μια σκάλα την οποία καλείται να ανέβει ο παίκτης. Στην αρχή αναβοσβήνει το πρώτο led της σκάλας και ο παίκτης πρέπει να πατήσει το κουμπί του όταν το led ανάβει, ώστε να ανέβει στο επόμενο σκαλοπάτι και να αρχίσει να αναβοσβήνει το επόμενο led της σκάλας (ενώ το προηγούμενο led θα παραμένει σταθερά αναμμένο). Για κάθε λανθασμένο πάτημα του κουμπιού του, ο παίκτης επιστρέφει στο πρώτο σκαλοπάτι, ενώ παράλληλα αυξάνεται ο ρυθμός με τον οποίο αναβοσβήνει το εκάστοτε led - σκαλοπάτι - στόχος. Το παιχνίδι ολοκληρώνεται όταν ο παίκτης καταφέρει να φτάσει στο τελευταίο σκαλοπάτι (και του ανακοινώνεται ο χρόνος που χρειάστηκε για να ανέβει τη σκάλα) ή όταν περάσει κάποιο χρονικό διάστημα της επιλογής σας ή όταν τελειώσουν οι ζωές του παίκτη.





Μέρος Β: Δημιουργία ενός συναγερμού - ανιχνευτή κίνησης



3.4 [INTRUDER DETECTION ALARM]

Συνδέστε έναν αισθητήρα κίνησης (PIR Sensor) στο Ρί και φτιάξτε ένα πρόγραμμα-συναγερμό στο οποίο, όταν εκτελείται:

α) ένα κουμπί ενεργοποιεί το συναγερμό και ένα άλλο τον απενεργοποιεί προσωρινά.

β) όσο ο συναγερμός είναι ενεργός, αναβοσβήνει ένα LED. Εναλλακτικά, εάν προτιμάτε το σύστημα παρακολούθησης να είναι πιο «διακριτικό», μπορείτε να απενεργοποιήσετε προσωρινά όλα τα leds, ακόμα και αυτά που βρίσκονται πάνω στο Pi.

γ) όταν ανιχνεύεται κίνηση, προτείνονται τα παρακάτω:

- 1. ανάβει ένα άλλο LED που ειδοποιεί οπτικά για την ανίχνευση.
- 2. αν διαθέτετε ηχεία, αναπαράγεται ένας ήχος (wav ή mp3) της επιλογής σας.
- 3. αν διαθέτετε κάμερα, λαμβάνεται φωτογραφία ή βίντεο του «εισβολέα».
- 4. αποστέλλεται σχετικό e-mail ή ειδοποίηση (notification) στον υπολογιστή ή στο κινητό σας και επισυνάπτεται στο μήνυμα αντίστοιχη φωτογραφία (ή βίντεο).

Χρήσιμα

A. Εκτέλεση εντολών συστήματος μέσω της Python

```
from os import system
system("omxplayer /home/pi/Sounds/PanicAlarm.mp3") # αναπαραγωγή ενός mp3
```

Β. Απενεργοποίηση των ενσωματωμένων LED του Pi



- Το Ρi έχει 2 μικρά LED σε μια γωνία του, τα οποία ανάβουν ως εξής:
- ACT: αναβοσβήνει (πράσινο) όταν υπάρχει πρόσβαση στην SD κάρτα.

- **PWR**: ανάβει σταθερά (κόκκινο) όταν το Ρί τροφοδοτείται με ικανοποιητικό ρεύμα.

Παρόλο που αυτά τα δύο LED χρησιμοποιούνται ήδη για συγκεκριμένο σκοπό, μπορούμε να τα ελέγξουμε, γνωρίζοντας ότι το ACT συνδέεται στο pin **35** και το PWR στο pin **47**. <u>https://gpiozero.readthedocs.org/en/v1.2.0/recipes.html#controlling-the-pi-s-own-leds</u>

Γ. Λήψη φωτογραφιών

- Για την Raspberry Pi Camera: <u>http://picamera.readthedocs.org/</u>
- Για USB Web Camera: https://www.raspberrypi.org/documentation/usage/webcams/README.md

Δ. Αποστολή push notification σε smartphone ή υπολογιστή με το Pushbullet

Εγκατάσταση και χρήση απλού Pushbullet python client: <u>https://pypi.python.org/pypi/pushbullet.py</u>

E. Αποστολή email με το Yagmail

Εύκολη αποστολή μηνυμάτων e-mail από gmail: https://github.com/kootenpv/yagmail





Πηγές Και χρήσιμοι σύνδεσμοι

Πηγές και υλικό αναφοράς

- Official recources to teach, learn and make with Raspberry Pi. <u>https://www.raspberrypi.org/resources/</u>
- Gpiozero interface package
 <u>http://gpiozero.readthedocs.org/</u>
- Προγραμματίζοντας με τον μικροελεγκτή Arduino. Ε. Πουλάκης, Ηράκλειο, 2015. http://users.sch.gr/manpoul/index.php/arduino
- Pythonies. Προγραμματισμός μέσα από παραδείγματα στην Python. <u>http://pythonies.mysch.gr/</u>
- CamJam Edukit worksheets. <u>http://camjam.me/</u>
- Joseph Haig Reaction Game.
 <u>https://github.com/jrmhaig/rpi_projects</u>
- StopIt! LED Game (for Arduino).
 <u>http://www.instructables.com/id/StopIt-LED-Game-powered-by-arduino/</u>
- Ladder Game. https://projects.drogon.net/raspberry-pi/gpio-examples/ladder-game/
- Picamera interface package. https://picamera.readthedocs.org
- Fswebcam simple webcam app. http://www.sanslogic.co.uk/fswebcam/
- yagmail Yet Another GMAIL/SMTP client. https://github.com/kootenpv/yagmail
- Simple Python client for PushBullet. https://pypi.python.org/pypi/pushbullet.py