



**ΠΑΝΕΠΙΣΤΗΜΙΟ
ΔΥΤΙΚΗΣ ΜΑΚΕΔΟΝΙΑΣ**

Ενσωματωμένα Συστήματα

Ενότητα: ΕΡΓΑΣΤΗΡΙΑΚΗ ΑΣΚΗΣΗ Raspberry

Απαιτούμενος Αρ. Εργατωρών: 3 (προσεγγιστικά)

Δρ. Μηνάς Δασυγένης

mdasyg@ieee.org

Τμήμα Μηχανικών Πληροφορικής και Τηλεπικοινωνιών

Εργαστήριο Ψηφιακών Συστημάτων και Αρχιτεκτονικής Υπολογιστών

<http://arch.ict.e.uowm.gr/mdasyg>

Άδειες Χρήσης

- Το παρόν εκπαιδευτικό υλικό υπόκειται σε άδειες χρήσης Creative Commons.
- Για εκπαιδευτικό υλικό, όπως εικόνες, που υπόκειται σε άλλου τύπου άδειας χρήσης, η άδεια χρήσης αναφέρεται ρητώς.



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο



ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



Περιεχόμενα

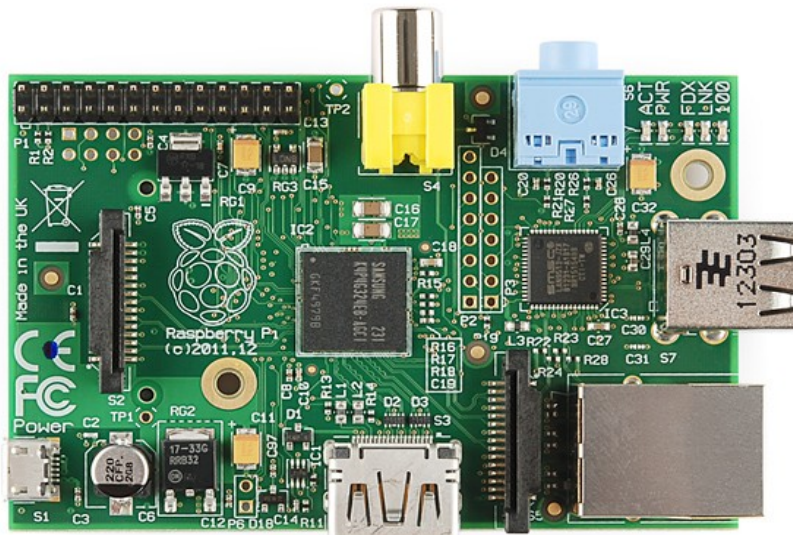
1. Σκοπός της άσκησης.....	4
2. Περιγραφή της πλακέτας.....	4
3. Μεταφόρτωση Image & Εγκατάσταση.....	5
4. Ρύθμιση.....	6
5. Χρήση python για είσοδο/έξοδο (GPIO).....	8
6. Προχωρημένα Θέματα: Σύνδεση με Arduino Slave.....	10
6.1 Ρύθμιση Arduino ως slave device.....	11
6.2 Ρύθμιση Rapsberry ως master device.....	13

1. Σκοπός της άσκησης

- Γνωριμία με την αναπτυξιακή πλακέτα Raspberry Type B
- Απαιτούμενο Υλικό: (α) αναπτυξιακή πλακέτα Raspberry Type B, (b) τροφοδοτικό USB με καλώδιο MicroUSB, (c) κάρτα μνήμης SD, (d) arduino UNO R3 + καλώδιο USB σύνδεσης.

2. Περιγραφή της πλακέτας

Η πλακέτα raspberry είναι μια αναπτυξιακή πλακέτα μορφής SBC, Single-board computer (=όλος ο υπολογιστής σε μια πλακέτα), και αναπτύχθηκε στο UK το 2012 προκειμένου να συμβάλει στην εκμάθηση των ΤΠΕ στα σχολεία. Για αυτό το λόγο είναι μια πλακέτα χαμηλού κόστους, που όμως μπορεί να χρησιμοποιηθεί ως πλήρης υπολογιστής, συνδέοντας μια οθόνη, ένα πληκτρολόγιο και ένα mouse. Από το 2012 έως τώρα έχουν παρουσιασθεί αρκετά μοντέλα Raspberry και πολλές άλλες πλακέτες SBC από άλλες ομάδες. Όλες έχουν αρκετά κοινά χαρακτηριστικά. Το μοντέλο που θα χρησιμοποιήσουμε σε αυτή την άσκηση είναι το Raspberry Pi - Model B (Εικόνα 1). Η πλακέτα αυτή φέρει έναν επεξεργαστή ARM1176JZFS στα 700Mhz, με chip γραφικών Videocore 4 GPU που είναι ικανό να αναπαράγει εικόνα στα 1080p30 H264. Επίσης, φέρει μια RJ45 θύρα δικτύου, μια υποδοχή για κάρτα SD η οποία φέρει το λειτουργικό σύστημα και τα αρχεία του χρήστη, δυο USB θύρες, μια HDMI έξοδο, έξοδο RCA (video) και ήχου 3.5mm, 512MB RAM, και τροφοδοτείται μέσω MicroUSB (κατ ελάχιστο με τροφοδοτικό 700ma 5V).



Εικόνα 1: Raspberry Pi - Model B

Το raspberry φέρει έναν ακροδέκτη επαφών GPIO στο οποίο υπάρχουν: 8 GPIO, UART, I2C, SPI, ακροδεκτες 3.3, 5V και γείωση.



ΠΟΛΥ ΣΗΜΑΝΤΙΚΟ: Δεν τοποθετούμε ή αφαιρούμε καλώδια στους ακροδέκτες όταν η πλακέτα τροφοδοτείται, γιατί υπάρχει σημαντική πιθανότητα να καταστραφεί το board λόγω του spike που θα δημιουργηθεί από το contact bouncing. Πρέπει να έχουμε κλείσει την πλακέτα και στη συνέχεια να τοποθετήσουμε ή να τροποποιήσουμε ή να αφαιρέσουμε τα καλώδια.

Σε αυτό το εργαστήριο θα ολοκληρώσουμε τα παρακάτω βήματα:

- Μεταφόρτωση Image & Εγκατάσταση
- Ρύθμιση
- Χρήση python για led blinking και push button input.
- Προχωρημένα θέματα: Σύνδεση Arduino ως slave για led blinking και push button input με python.

3. Μεταφόρτωση Image & Εγκατάσταση

Το raspberry υποστηρίζεται από διάφορες εκδόσεις Linux. Η έκδοση που θα χρησιμοποιήσετε σε αυτό το εργαστήριο είναι η raspbian. Από τις παρακάτω διευθύνσεις να κατεβάσετε μια τελευταία έκδοση και να την αντιγράψετε σε μια SD κάρτα με ένα πρόγραμμα disk imager (π.χ. dd ή win32diskimager αν είστε σε windows).

https://downloads.raspberrypi.org/raspbian_lite_latest

https://downloads.raspberrypi.org/raspbian_lite/images/

Τα default στοιχεία σύνδεσης για αυτά τα images ως προς username/password είναι:

Username: pi

Password: raspberry

Σημειώστε ότι το εργαστήριο αυτό ολοκληρώθηκε με επιτυχία στο image:

https://downloads.raspberrypi.org/raspbian_lite/images/raspbian_lite-2018-04-19/

ΠΡΟΣΟΧΗ: θα πρέπει να κάνετε τα παρακάτω βήματα:

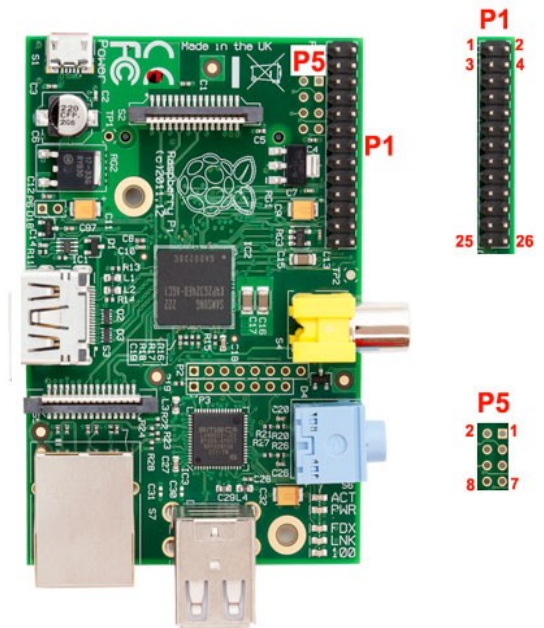
- Αποσυμπίεση του Image (προσέξτε αν εμφανιστεί error!).
- Εγγραφή σε όλο το block device (π.χ. /dev/sdb και όχι σε διαμέρισμα π.χ. /dev/sdb1).

Σημειώστε ότι σε default ρύθμιση η πλακέτα επιτρέπει σειριακή σύνδεση στην ταχύτητα 115200, και επίσης έχει ενεργοποιημένη την έξοδο RCA/video out. Επίσης, ο δαίμονας SSH είναι απενεργοποιημένος και θα πρέπει να τον ενεργοποιήσετε.

4. Ρύθμιση

Όταν θα ξεκινήσει η πλακέτα θα πρέπει να βρείτε την IP που θα έχει λάβει για να συνδεθείτε. Για τις αρχικές ρυθμίσεις μπορείτε να κάνετε κάτι από τα παρακάτω:

- Χρησιμοποιείτε ένα USB TTL μετατροπέα που συνδέεται στο USB του host υπολογιστή σας και στις γραμμές RxD, TxD και GND της πλακέτας. [στο pin 6: GND, στο pin 8: RXD, στο pin 10:TXD].
- Συνδέετε μια HDMI οθόνη και ένα USB πληκτρολόγιο.
- Συνδέετε μια τηλεόραση RCA και ένα USB πληκτρολόγιο.
- [Δεν προτείνεται] Στο HOST υπολογιστή σας κάνετε mount την κάρτα και τοποθετείτε κατάλληλο script στο cronab (δείτε φυλλάδιο beagleboard) που αναφέρει την IP που έχει σε μια web σελίδα.
- [Δεν προτείνεται] Από υπολογιστή στο ίδιο subnet (π.χ. στο ίδιο switch), ως root κάντε μια σάρωση στο τοπικό δίκτυο (έστω 192.168.70.0) για τους υπολογιστές που είναι ανοιχτοί.
`nmap -sP 192.168.70.0/24`



Εικόνα 2: Αρίθμηση ακροδεκτών

Π.χ.:

ΠΡΟΣΟΧΗ: Αν το raspberry χάνει το network connectivity (χάνονται πακέτα ping) τότε σημαίνει ότι το τροφοδοτικό δεν παρέχει τουλάχιστον 700mA στην πλακέτα και πρέπει να το αντικαταστήσετε.

Αφού συνδεθείτε στο board δώστε την εντολή `raspi-config`

και ενεργοποιήστε το SSH στο μενού interfaces. Σε περίπτωση που θέλετε να δώσετε στατική IP μπορείτε να δείτε την IP που έχετε λάβει με `ifconfig` και να την τοποθετήσετε στατικά στο board σας. Τροποποιήστε κατάλληλα τις παρακάτω ρυθμίσεις, όπως σε κάθε σύστημα Ubuntu:

```
# edit /etc/network/interfaces.d/eth0
# The primary network interface
allow-hotplug eth0
iface eth0 inet static
```

Raspberry Pi P1 Header					
PIN #	NAME			NAME	PIN #
	3.3 VDC Power	1		2	5.0 VDC Power
8	SDA0 (I2C)	3		4	DNC
9	SCL0 (I2C)	5		6	0V (Ground)
7	GPIO 7	7		8	TxD
	DNC	6		10	RxD
0	GPIO 0	11		12	GPIO1
2	GPIO2	13		14	DNC
3	GPIO3	15		16	GPIO4
	DNC	17		18	GPIO5
12	MOSI	19		20	DNC
13	MISO	21		22	GPIO6
14	SCLK	23		24	CE0
	DNC	25		26	CE1

Εικόνα 3: P1

```
address 192.168.2.50
netmask 255.255.255.0
gateway 192.168.2.1
```

Ενημερώστε τα πακέτα του συστήματος (όπως σε κάθε Ubuntu):

```
apt-get update && apt-get upgrade
```

Σε περίπτωση που σας εμφανίσει μήνυμα size mismatch τότε δώστε τις παρακάτω εντολές για να καθαρίσετε τη cache και να τη ξαναδημιουργήσετε:

```
sudo apt-get clean
```

```
sudo rm /var/lib/apt/lists/partial/*
```

```
sudo nano /etc/apt/sources.list
```

και θα τοποθετήσετε στο αρχείο sources.list ΜΟΝΟ τα παρακάτω official repositories (για το raspbian):

```
deb http://archive.raspbian.org/raspbian wheezy main contrib
non-free
```

```
deb-src http://archive.raspbian.org/raspbian wheezy main
contrib non-free
```

```
deb http://mirrordirector.raspbian.org/raspbian/ stretch main
contrib non-free rpi
```

Επιπρόσθετα θα πρέπει να έχετε το επίσημο κλειδί που υπογράφονται τα πακέτα του raspbian, οπότε με την παρακάτω εντολή θα το μεταφορτώσετε και θα το τοποθετήσετε στο keychain του συστήματος.

```
wget http://archive.raspbian.org/raspbian.public.key -O - |
sudo apt-key add -
```

Στη συνέχεια κάντε update και upgrade

```
apt-get update && apt-get upgrade
```

Ενώ ενημερώστε το firmware του συστήματος με την εντολή:

```
rpi-update
```

=> Σε αυτό το σημείο μπορείτε να τοποθετήσετε τον κώδικα ενημέρωσης της διεύθυνσης που αναγράφεται στο φυλλάδιο beagleboard, ώστε να γνωρίζετε την IP που λαμβάνει ανά πάσα στιγμή, ΑΝ ΔΕΝ ΕΧΕΤΕ ρυθμίσει στατική IP.

5. Χρήση python για είσοδο/έξοδο (GPIO)

Οι ακροδέκτες του Raspberry φέρουν κάποια ονόματα και έχουν συγκεκριμένη αρίθμηση όπως φαίνεται στην Εικόνα 2. Ο πιο σημαντικός ακροδέκτης εισόδου εξόδου είναι ο P1, όπως φαίνεται στην Εικόνα 3. Παρατηρήστε ότι οι ακροδέκτες συνδέονται με συγκεκριμένες λειτουργίες. Π.χ. ο ακροδέκτης 8 αναγράφει TxD που σημαίνει σειριακή αποστολή δεδομένων, ενώ ο 10 RxD που σημαίνει σειριακή ανάγνωση δεδομένων. Σε περίπτωση που απαιτείται η χρήση GPIO τότε χρησιμοποιούνται τα GPIO? (general purpose input output).

Ο πιο εύκολος και δημοφιλής τρόπος να αποκτήσετε πρόσβαση στα pins είναι μέσω της python. Θα πρέπει να έχετε εγκατεστημένο κατάλληλο πακέτο για το interface. Δώστε την εντολή εγκατάστασης του βασικού πακέτου για python3 και python2.

```
apt-get -y install python3-rpi.gpio python-rpi.gpio
```

Επίσης θα χρειαστεί να εγκαταστήσετε το πρόγραμμα εγκατάστασης βιβλιοθηκών της python (pip) με τις εντολές

```
apt-get -y install python3-pip python-pip
```

Χρησιμοποιήστε ένα breadboard, μια LED και μια αντίσταση από 220Ω έως 1KΩ. Θα πρέπει να συνδέσετε την μια άκρη του LED σε ένα GPIO ακροδέκτη, μετά στην αντίσταση, και από την αντίσταση στο GND της πλακέτας. Να σημειώστε ότι η αντίσταση τοποθετείται επειδή δεν πρέπει να υπερβεί το ρεύμα τα 15mA στα 3.3 Volt (αφού υπολογίσουμε και τη πτώση τάσης στο LED). Το ίδιο ισχύει και στο Arduino και σε κάθε αναπτυξιακή πλακέτα.

Η αρίθμηση των pins στο raspberry (python) μπορεί να γίνει με 2 τρόπους:

1. Αρίθμηση του board, δηλαδή σύμφωνα με τους αριθμούς που φαίνονται στο μέσω της Εικόνα 2. Σε αυτή την περίπτωση στο python θα έχουμε μετά τη γραμμή `import RPi.GPIO as GPIO` τη γραμμή `GPIO.setmode(GPIO.BOARD)`.
2. Αρίθμηση της εταιρίας broadcom (BCM). Αυτοί οι αριθμοί φαίνονται στις πλευρές της Εικόνα 2. Σε αυτή την περίπτωση στο python θα έχουμε μετά τη γραμμή `import RPi.GPIO as GPIO` τη γραμμή `GPIO.setmode(GPIO.BCM)`.

Χρησιμοποιήστε τον παρακάτω κώδικα (και τροποποιήστε τον αν χρειαστεί) για να αναβοσβήσετε το LED:


```

import RPi.GPIO as GPIO      # calling header file for GPIO's of PI
import time                  # calling for time to provide delays in program

GPIO.setmode (GPIO.BOARD)   # programming the GPIO by BOARD pin numbers, GPIO21 is
called as PIN40
GPIO.setup(40,GPIO.OUT)      # initialize digital pin40 as an output.
GPIO.output(40,1)           # turn the LED on (making the voltage level HIGH)
time.sleep(1)               # sleep for a second
GPIO.cleanup()              # turn the LED off (making all the output pins LOW)
time.sleep(1)               #sleep for a second

#loop is executed second time
GPIO.setmode (GPIO.BOARD)
GPIO.setup(40,GPIO.OUT)
GPIO.output(40,1)
time.sleep(1)
GPIO.cleanup()
time.sleep(1)

#loop is executed third time
GPIO.setmode (GPIO.BOARD)
GPIO.setup(40,GPIO.OUT)
GPIO.output(40,1)
time.sleep(1)
GPIO.cleanup()
time.sleep(1)

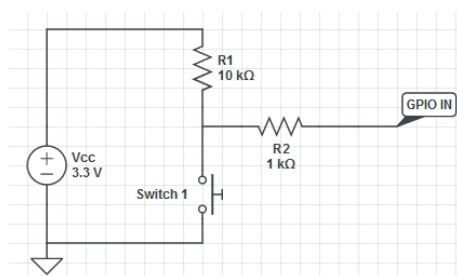
```

Για τη σύνδεση με ένα κουμπί, θα πρέπει να το συνδέσετε με τέτοιο τρόπο ώστε όταν πατηθεί θα συνδεθεί ένα GPIO στη γείωση (GND). Για λόγους προστασίας ή θα χρησιμοποιήσετε μια εξωτερική αντίσταση ή θα ενεργοποιήσετε την εσωτερική αντίσταση PULL-UP στο pin του επεξεργαστή. Η αντίσταση PULL-UP σημαίνει ότι όταν το pin δεν έχει συνδεθεί (floating) τότε η PULL-UP το έλκει προς το VCC, οπότε η τιμή που θα διαβάσει κάποιος είναι '1'. Κάποιοι επεξεργαστές έχουν αντιστάσεις PULL-DOWN δηλαδή τα floating pins τα έλκουν στο GND και η κανονική κατάσταση είναι 0. Μπορείτε να χρησιμοποιήσετε το PULL-DOWN αν θέλετε, αλλά η συνδεσμολογία του κουμπιού θα είναι διαφορετική (θα συνδεθεί στο VCC).

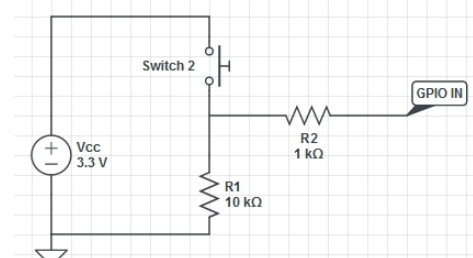
Η Εικόνα 4 δείχνει ένα αναλυτικό κύκλωμα για τη συνδεσμολογία pull-up, όπου φαίνεται ότι το floating GPIO pin (διακόπτης switch 1 δεν έχει πατηθεί) συνδέεται στο VCC και άρα στην κανονική κατάσταση διαβάζει '1'.

Η Εικόνα 5 δείχνει ένα αναλυτικό κύκλωμα για τη συνδεσμολογία pull-down, όπου φαίνεται ότι το floating GPIO pin συνδέεται στο GND στην κανονική κατάσταση (όταν δεν έχει πατηθεί το switch 1) και άρα διαβάζει '0'.

Αρκετοί επεξεργαστές (όπως και ο ARM που έχει το Raspberry) έχουν εσωτερικά κυκλώματα pull-up και pull-down. Δε χρειάζεται να υλοποιήσετε όλο το κύκλωμα που φαίνεται στις 2 αυτές εικόνες.



Εικόνα 4: Pull-UP



Εικόνα 5: Pull-Down

- **Επιλέξτε τη συνδεσμολογία Pull-up.** Ρυθμίστε κατάλληλα το GPIO μέσω `python`, συνδέστε ένα κουμπί ανάμεσα στο GPIO και στο GND και επιβεβαιώστε την ορθή εκτέλεση. Να εμφανίζεται το μήνυμα αν το κουμπί έχει πατηθεί ή όχι.
- **Επιλέξτε τη συνδεσμολογία Pull-down.** Ρυθμίστε κατάλληλα το GPIO μέσω `python`, συνδέστε ένα κουμπί ανάμεσα στο VCC (3.3V) και στο GPIO και επιβεβαιώστε την ορθή εκτέλεση. Να εμφανίζεται το μήνυμα αν το κουμπί έχει πατηθεί ή όχι.

Να υλοποιήσετε και τις 2 συνδεσμολογίες για να κατανοήσετε τη διαφορά που έχουν.

Σας δίνεται το παρακάτω κομμάτι κώδικα που θα σας βοηθήσει. Θα χρειαστεί να βρείτε σε ποιο από τα 2 tasks αναφέρεται και σε πιο GPIO, και να το τροποποιήσετε για το επόμενο task.

```
import time
import RPi.GPIO as GPIO
GPIO.setmode(GPIO.BCM)
GPIO.setwarnings(False)

button = 4
GPIO.setup(button, GPIO.IN, GPIO.PUD_UP)

while True:
    button_state = GPIO.input(button)
    if button_state == GPIO.HIGH:
        print ("HIGH")
    else:
        print ("LOW")
    time.sleep(0.5)
```

6. Προχωρημένα Θέματα: Σύνδεση με Arduino Slave

Υπάρχουν περιπτώσεις που τα GPIO του raspberry δεν είναι αρκετά για τις διασυνδέσεις που απαιτούνται. Σε αυτή την περίπτωση μπορούμε να χρησιμοποιήσουμε μια επιπρόσθετη πλακέτα (π.χ. arduino) την οποία θα συνδέσουμε μέσω καλωδίου USB. Μπορούμε να χρησιμοποιήσουμε κάποιο πακέτο, το οποίο θα φορτώσει ένα slave κώδικα στο arduino προγραμματιζόμενο κατά το

χρόνο εκτέλεσης. Από το raspberry μέσω python θα δίνονται εντολές για το κάθε pin είτε να τεθεί σε λειτουργία εισόδου ή εξόδου και επίσης αντίστοιχα να διαβάζει ή να γράφει μια κατάσταση.

Μπορούμε να υλοποιήσουμε μια δικιά μας βιβλιοθήκη, αλλά καλύτερα να χρησιμοποιήσουμε κάποια έτοιμη. Μια τέτοια βιβλιοθήκη είναι το nanpy (<https://pypi.org/project/nanpy/>).

6.1 Ρύθμιση Arduino ως slave device

Κάντε compile και γράψτε τον αντικειμενικό κώδικα του nanpy σε ένα arduino και συνδέστε το στο raspberry. Δε θα χρειαστεί να κάνετε κάτι άλλο στο arduino. Κατεβάστε τον κώδικα από: <https://pypi.python.org/pypi/nanpy>

Μπορείτε να κάνετε compile τον κώδικα είτε στα Windows, είτε στο Linux. Προτείνεται να εγκαταστήσετε τα απαραίτητα πακέτα στο raspberry που έχει συνδεθεί στο arduino, ώστε να μη χρειαστείτε άλλο υπολογιστή. Οι οδηγίες που ακολουθούν μπορεί να γίνουν σε ένα οποιοδήποτε Ubuntu ή Raspbian (raspberry) σύστημα.

Εγκαταστήστε τα απαραίτητα πακέτα:

```
apt-get install avrdude git arduino-core arduino-mk
```

Μεταφορτώστε το αναπτυξιακό πακέτο του arduino για την αρχιτεκτονική του υπολογιστή που έχετε (μπορείτε να κατεβάσετε την τελευταία έκδοση από <https://www.arduino.cc/en/Main/Software>). Υπενθυμίζεται ότι η αρχιτεκτονική σας εμφανίζεται με την εντολή `uname -a`

(A) Για raspberry ή άλλη πλακέτα με επεξεργαστή APM:

```
cd /tmp && wget https://downloads.arduino.cc/arduino-1.8.5-linuxarm.tar.xz
```

(B) Για επεξεργαστή 32bit Intel/AMD:

```
cd /tmp && wget https://downloads.arduino.cc/arduino-1.8.7-linux32.tar.xz
```

(C) Για επεξεργαστή 64bit Intel:

```
cd /tmp && wget https://downloads.arduino.cc/arduino-1.8.7-linux64.tar.xz
```

Αφού τις κατεβάσετε τις αποσυμπίεζετε.

```
mkdir /opt ; cd /opt ; tar -xvJf /tmp/arduino-*
```

```
ln -s /opt/arduino-1.* /opt/arduino
```

```
ln -s /opt/arduino /usr/share
```

```
ln -s /usr/share/arduino/hardware/arduino/avr/boards.txt /usr/share/arduino/hardware/arduino/
```

Μεταφέρετε το firmware στο σύστημά σας:

```
cd
git clone https://github.com/nanpy/nanpy-firmware.git
cd nanpy-firmware
./configure.sh
cd Nanpy
```

Στη συνέχεια συνδέστε το Arduino UNO και με την εντολή `dmesg` βρείτε σε ποια θύρα έχει συνδεθεί (π.χ. `ttyACM0` , `ttyUSB0` κτλ).

Δημιουργούμε το αρχείο Makefile (και τροποποιήστε το PORT αν είναι διαφορετικό):

```
cat > Makefile << EOM
ARDUINO_DIR = /usr/share/arduino
#ARDUINO_LIBS = Ethernet Ethernet/utility SPI
BOARD_TAG = uno
ARDUINO_PORT = /dev/ttyACM0
include /usr/share/arduino/Arduino.mk
EOM
```

Μπορείτε να τροποποιήσετε το αρχείο `Nanpy/cfg.h` αν θέλετε να συμπεριλάβετε υποστήριξη για επιπρόσθετο hardware (δε χρειάζεται για αυτό το εργαστήριο).

Για να γίνει compile και να δημιουργηθεί το `.hex` που θα γραφεί στη flash μνήμη του arduino, δίνετε μέσα στον κατάλογο του Nanpy:

```
make
```

Σε περίπτωση που δε δημιουργηθεί το αρχείο αλλά σας αναφέρει κάποιο πρόβλημα με `"undefined reference to `countPulseASM"` τότε αυτό σημαίνει ότι το Arduino SDK συνεχίζει και σε αυτή την έκδοση να έχει πρόβλημα με τα αρχεία `.S1` και θα πρέπει είτε να χρησιμοποιήσετε μια παλαιότερη έκδοση, είτε να τοποθετήσετε στο τέλος του αρχείου:

```
/opt/arduino/hardware/arduino/avr/cores/arduino/wiring\_pulse.c
```

τον παρακάτω κώδικα:

¹<https://github.com/francoiscampbell/CLionArduinoPlugin/issues/23>

```

#include <Arduino.h>
#include <stdlib.h>

unsigned long countPulseASM(volatile uint8_t *port, uint8_t bit, uint8_t stateMask, unsigned long
maxloops)
{
    unsigned long width = 0;
    // wait for any previous pulse to end
    while ((*port & bit) == stateMask)
        if (--maxloops == 0)
            return 0;

    // wait for the pulse to start
    while ((*port & bit) != stateMask)
        if (--maxloops == 0)
            return 0;

    // wait for the pulse to stop
    while ((*port & bit) == stateMask) {
        if (++width == maxloops)
            return 0;
    }
    return width;
}

```

Για να το μεταφορτώσετε στο συνδεδεμένο arduino δώστε:

```
make upload
```

Διαβάστε προσεκτικά τη γραμμή που αναφέρει ότι ολοκληρώθηκε η εγγραφή και ότι δεν αναφέρθηκε κάποιο πρόβλημα.

Μπορείτε να ανεβάσετε το firmware και χειρονακτικά με την εντολή:

```
avrdude -c /usr/share/arduino/hardware/tools/avr/etc/avrdude.conf -p atmega328p -c
arduino -P /dev/ttyACM0 -b 57600 -D -Uflash:w:buiild-uno/Nanpy.hex
```

(αν δε λειτουργεί για 57600 δοκιμάστε 115200).

6.2 Ρύθμιση Rapsberry ως master device

Εγκαταστήστε το πακέτο nanpy για python στο raspberry μέσω των εντολών:

```
pip install nanpy
```

```
pip3 install nanpy .
```

Στο raspberry για να συνδεθείτε στο arduino δώστε μέσα στο περιβάλλον της python3:

```
from nanpy import ArduinoApi
```

```
from nanpy import SerialManager
```

```
from nanpy.arduinotree import ArduinoTree
```

```
connection = SerialManager()
```

```
a = ArduinoApi(connection=connection)
```

Αν εμφανιστεί πρόβλημα στην τελευταία εντολή, απλώς κάντε reset το arduino γιατί δεν έχει ξεκινήσει το νέο firmware.

Στη συνέχεια συνδέστε ένα κουμπί και ένα LED (μπορείτε να χρησιμοποιήσετε το ενσωματωμένο LED πάνω στην πλακέτα που συνδέεται στο PIN13) σε κατάλληλα pins στο arduino (ΕΚΤΟΣ των 0 και 1 που είναι τα pin επικοινωνίας του USB).

Θέστε το pin σε

Κατάσταση εισόδου:

```
a.pinMode(11, a.INPUT)
```

ή αν απαιτείται και PULL-UP

```
a.pinMode(11, a.INPUT_PULLUP)
```

Κατάσταση εξόδου

```
a.pinMode(13, a.OUTPUT)
```

Εγγραφή σε pin του Arduino

Αν το pin είναι κατάσταση εξόδου, γράψτε μια τιμή με την εντολή:

```
a.digitalWrite(13, a.HIGH)
```

```
a.digitalWrite(13, a.LOW)
```

Ανάγνωση από pin του Arduino

Αν είναι σε κατάσταση εισόδου, διαβάστε την τιμή με την εντολή:

```
a.digitalRead(13)
```

Να δημιουργήσετε πρόγραμμα σε python στο raspberry που θα ελέγχει ένα arduino. Στο arduino θα έχει συνδεθεί ένα LED και ένα push button. Όταν πατηθεί το button θα πρέπει να αναβοσβήσει μια φορά το LED. Κάθε φορά που θα πατιέται το button θα αναβοσβήνει το LED τόσες φορές όσο ο αριθμός των φορών που έχει πατηθεί από την αρχή ως εκείνο το σημείο.