



## ΑΠΟΜΑΚΡΥΣΜΕΝΟΣ ΕΛΕΓΧΟΣ ΗΛΕΚΤΡΙΚΩΝ ΣΥΣΚΕΥΩΝ-ΙΟΤ

Τμήμα Μηχανικών Βιομηχανικής  
Σχεδίασης & Παραγωγής



Συγγραφέας: ΙΓΝΑΤΙΟΥ ΜΙΧΑΗΛ-ΑΓΓΕΛΟΣ

Αριθμός μητρώου: 41693

Επιβλ. Καθηγητής :ΔΡΟΣΟΣ ΧΡΗΣΤΟΣ

Ημερομηνία παράδοσης :3/4/2019

## **ΔΗΛΩΣΗ ΣΥΓΓΡΑΦΕΑ ΠΤΥΧΙΑΚΗΣ ΕΡΓΑΣΙΑΣ**

Ο κάτωθι υπογεγραμμένος Ιγνατίου Μιχαήλ Άγγελος του Πέτρου φοιτητής του Τμήματος Τμήμα Μηχανικών Βιομηχανικής Σχεδίασης & Παραγωγής του Πανεπιστημίου Δυτικής Αττικής, πριν αναλάβω την εκπόνηση της Πτυχιακής Εργασίας μου, δηλώνω ότι ενημερώθηκα για τα παρακάτω:

«Η Πτυχιακή Εργασία (Π.Ε) αποτελεί προϊόν πνευματικής ιδιοκτησίας τόσο του συγγραφέα, όσο και του Ιδρύματος και θα πρέπει να έχει μοναδικό χαρακτήρα και πρωτότυπο περιεχόμενο.

Απαγορεύεται αυστηρά οποιοδήποτε κομμάτι κειμένου της να εμφανίζεται αυτούσιο ή μεταφρασμένο από κάποια άλλη δημοσιευμένη πηγή. Κάθε τέτοια πράξη αποτελεί προϊόν λογοκλοπής και εγείρει θέμα Ηθικής Τάξης για τα πνευματικά δικαιώματα του άλλου συγγραφέα. Αποκλειστικός υπεύθυνος είναι ο συγγραφέας της Π.Ε, ο οποίος φέρει και την ευθύνη των συνεπειών, ποινικών και άλλων, αυτής της πράξης.

Πέραν των όποιων ποινικών ευθυνών του συγγραφέα, σε περίπτωση που το Ίδρυμα του έχει απονείμει Πτυχίο, αυτό ανακαλείται με απόφαση της Συνέλευσης του Τμήματος. Η Συνέλευση του Τμήματος με νέα απόφασή της, μετά από αίτηση του ενδιαφερόμενου, του αναθέτει εκ νέου την εκπόνηση Π.Ε με άλλο θέμα και διαφορετικό επιβλέποντα καθηγητή. Η εκπόνηση της εν λόγω Π.Ε πρέπει να ολοκληρωθεί εντός τουλάχιστον ενός ημερολογιακού βμήνου από την ημερομηνία ανάθεσής της.

**Ο Δηλών**

**Ημερομηνία**

**Ιγνατίου Μιχαήλ Άγγελος**

**3/4/2019**

# 1. Περίληψη

Το έξυπνο σπίτι (internet of things) είναι αδιαμφισβήτητα το πιο πολυσυζητημένο θέμα στον χώρο της τεχνολογίας τα τελευταία έτη φέρνοντας το μέλλον στο σήμερα. Η αγορά έχει κατακλειστεί από “Έξυπνες” συσκευές αλλά και από εργαλεία τα οποία καθιστούν εφικτό τον πειραματισμό πάνω στην αυτοματοποίηση λειτουργιών που διευκολύνουν τη ζωή μας καθώς και τον απομακρυσμένο τους έλεγχο. Ίσως το πιο διαδεδομένο και όχι άδικα, είναι το raspberry PI το οποίο παρέχει υπολογιστική ισχύ ισάξια προσωπικού υπολογιστή σε προσιτή τιμή και όγκο αντιστρόφως ανάλογο των δυνατοτήτων του. Στην παρούσα πτυχιακή μέσω του raspberry PI δίνεται η δυνατότητα ελέγχου σε μια σειρά απο ρελέ τα οποία ο χρήστης με την σειρά του μπορεί να τα συνδέσει σε ηλεκτρικές συσκευές. Η επικοινωνία γίνεται μέσω του πρωτοκόλου SSH ενώ ο προγραμματισμός για τον έλεγχο των ηλεκτρικών καταστάσεων τον ρελέ με Python. Η δημιουργία διεπαφής ιστού είναι γραμμένη σε HTML5, CSS, JAVASCRIPT και PHP μέσω της οποίας μπορούμε να επικοινωνούμε με το Raspberry PI και να του δίνουμε εντολές.

## 1.1. Abstract

Smart home(IOT) is without a doubt the top trend in technology realm nowadays bringing the future to everyday life .Market is already packed with commercial smart devices and tools that makes possible the experimentation in automation .Possibly the most ubiquitous of them all is Rasberry PI giving a vast spectrum of possibilities in an affordable cost and in volume reverse proportional of its capabilities .Via Rasberry PI is given the opportunity to interfere with a series of relays .Hence the end user can remote control the electrical devices that relays pins are connected at .Communication made possible via SSH protocol and programming of electrical states of relays with Python .Web interface is written in HTML5, CSS, JAVASCRIPT and PHP that communicates directly with rasberry PI .



## 2. Περιεχόμενα

1.	Περίληψη.....	3
1.1.	Abstract.....	3
2.	Contents .....	4
3.	Ιστορική αναδρομή αυτοματισμού .....	5
3.1.	Σύλληψη της ιδέας του αυτοματισμού .....	5
3.2.	Πρώτος ορισμός του συστήματος κατά τον Καλλικρατίδη .....	6
3.3.	Αυτόματες πύλες ναού .....	7
3.4.	Υδραυλικά αυτόματα.....	8
3.5.	Ιστορία απομακρυσμένου ελέγχου .....	9
4.	Internet Of things .....	10
4.1.	Τι είναι το Internet of Things ; .....	10
4.2.	Εξυπνα σπιτια .....	10
4.3.	Έξυπνη πόλη .....	11
4.4.	Internet of Medical Things (IOMT).....	12
5.	Raspberry PI.....	13
5.1.	Εφαρμογές Raspberry PI .....	14
5.2.	Προσωπικός υπολογιστής(PC) .....	14
5.3.	Μηχανή για torrent .....	15
5.4.	Αναπαραγωγή πολυμέσων .....	15
5.5.	Gaming.....	16
6.	Εγκατάσταση λειτουργικού συστήματος στο Raspberry.....	17
6.1.	Rasbian-Noobs .....	17
6.2.	Εγκατάσταση του λογισμικού .....	19
7.	Εκκίνηση του Raspberry PI , πρώτη επαφή .....	22
7.1.	SSH και raspberry PI.....	25
7.2.	Επικοινωνία του Ρελε 8 καναλιών με το Raspberry PI .....	27
7.3.	Έλεγχος παροχής ισχύος με python scripts .....	31
8.	Απομακρυσμένος έλεγχος μέσω ιστού .....	35
8.1.	Εγκατάσταση απαραίτητων εργαλείων .....	35
8.2.	Python Scripts .....	46
8.3.	Δοκιμάζοντας την WEB εφαρμογή .....	55
9.	Αποτελέσματα .....	60
10.	Βιβλιογραφία .....	61

### 3. Ιστορική αναδρομή αυτοματισμού

#### 3.1. Σύλληψη της ιδέας του αυτοματισμού

Ήδη από τα προ Χριστού χρόνια εμφανίστηκε ως φαντασία ο αυτοματισμός. Στην Ιλιάδα του Ομήρου όπου μπαίνει στο εργαστήρι του ξακουστού για την τέχνη του Ηφαίστου δεν άργησε να του αποδοθεί η θεική ιδιότητα. Περιγράφοντας τον θεό περιστοιχισμένο από εργαλεία, τις μηχανές και τα έργα του. Την μυθική για τότε τεχνολογία ο ποιητής περιγράφει με την φαντασία του έργα θαυμαστά, θαυματοργά τόσο με την έννοια του θαυμασμού που προκαλεί η τελειότητα της κατασκευής τους, όσο και με την έννοια του υπερφυσικού και του αδύνατου, της ακατόρθωτης τεχνικής επιθυμίας, που η ολοκλήρωσή τους αποδίδεται στους θεούς. Εκεί πρωτοσυναντάται η πρώτη σύλληψη, σαν φαντασία ακόμα, του απομακρυσμένου ελέγχου τριπόδων όπου όπως αναφέρει ο Όμηρος «αυτόματα» - αυτοκινούμενα γυρνούν στην σύναξη των θεών και προσφέρουν οίνο και νερό και μόνοι τους να γυρνούν στο οίκημα. Ένα θάυμα να τους βλέπει κανείς.

Λίγο πιο κάτω στην ίδια ραψωδία, ο ποιητής προβέπει-φαντάζεται την αναγκαιότητα της αυτοματοποίησης στην παραγωγή του εργαστηρίου του ηφαίστου με τα αυτορυθμιζόμενα φουσερά. «Πήγε ο Ήφαιστος στα φουσερά του, τα στρεψε προς τη φωτιά και τα πρόσταξε να αρχίσουν να δουλεύουν. Και τα φουσερά, είκοσι όλα μαζί, φυσούσανε μες τα καμίνια βγάζοντας κάθε λογής δυνατόν αέρα, άλλοτε γρήγορα σαν να βιαζότανε κι άλλοτε αργά, όπως ήθελε ο Ήφαιστος κι όπως το ζήταγε η δουλειά του».



«Το καμίνι του Ηφαίστου» Μελανοφόρο αγγείο, 6<sup>ος</sup> αιώνας Π.Χ.

Η φαντασία-προοικονομία του ποιητή δεν σταματά στην Ιλιάδα όπου καταπιάνεται με την τέχνη του πολέμου και τα αυτόματα συστήματα αποδίδονται στους θεούς. Η Οδύσσεια είναι ένας ύμνος στην ευστροφία και την εφευρετικότητα του Οδυσσέα και κατ' επέκταση του ανθρώπινου είδους. Πάνσοφος για την εποχή του ο Όμοιρος, είχε καταλάβει την αναγκαιότητα των θαλάσσιων μεταφορών και την βελτιστοποίησης της ναυσιπλοΐας. Τα πλοία των Φαιάκων είναι αυτόματα. Ξέρουν από μόνα τους να ταξιδεύουν, να προσανατολίζονται, να κατευθύνονται στον προορισμό τους χωρίς κυβερνήτες, χωρίς πηδάλιο, ανεξαρτήτως καιρικών συνθηκών. Περιγράφει με τα εξής λόγια τη λιμενοποιϊκή τέχνη των Φαιάκων και μια ιδιαίτερη ναυπηγοκατασκευαστική τεχνολογία :

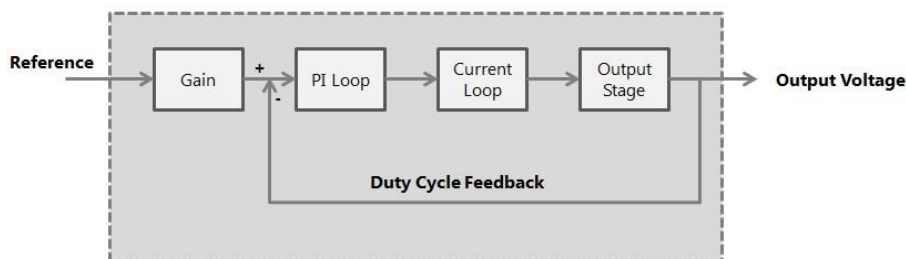
«Πες μου για τη χώρα σου και το λαό σου και την πόλη σου για να σε πάνε εκεί τα πλοία μας τα κατασκευασμένα με σκέψη. Γιατί δεν υπάρχουν κυβερνήτες στα πλία των Φαιάκων, ούτε πηδάλια σαν αυτά που έχουν τα άλλα καράβια. Παρα τα πλία των Φαιάκων ξέρουν τις διαθέσεις και τις σκέψεις των ανθρώπων και γνωρίζουν τις πατρίδες όλων, και με εξαιρετική ταχύτητα διανύουν τις θαλάσσινες αποστάσεις, ακόμη και όταν έχει σκοτάδι και συννεφιά. Και ποτέ δεν υπάρχει φόβος να πάθουν καμιά βλάβη.»

Σε αυτό το απόσπασμα ο Όμοιρος πάει ένα βήμα παρακάτω και φαντάζεται πέρα του αυτομάτου και υπονοεί ένα τεχνικό υπόβαθρο πάνω στο οποίο θα στηριζόταν αυτή η τεχνητή νοημοσύνη

### 3.2. Πρώτος ορισμός του συστήματος κατά τον Καλλικρατίδη

Τον όρο «σύστημα» τον εισάγει για πρώτη φορά ο πυθαγόρειος Καλλικρατίδης όπου και δίνει τον ορισμό του, και ειδικότερα του συστημάτων ελέγχου. Ο ορισμός αυτός είναι :

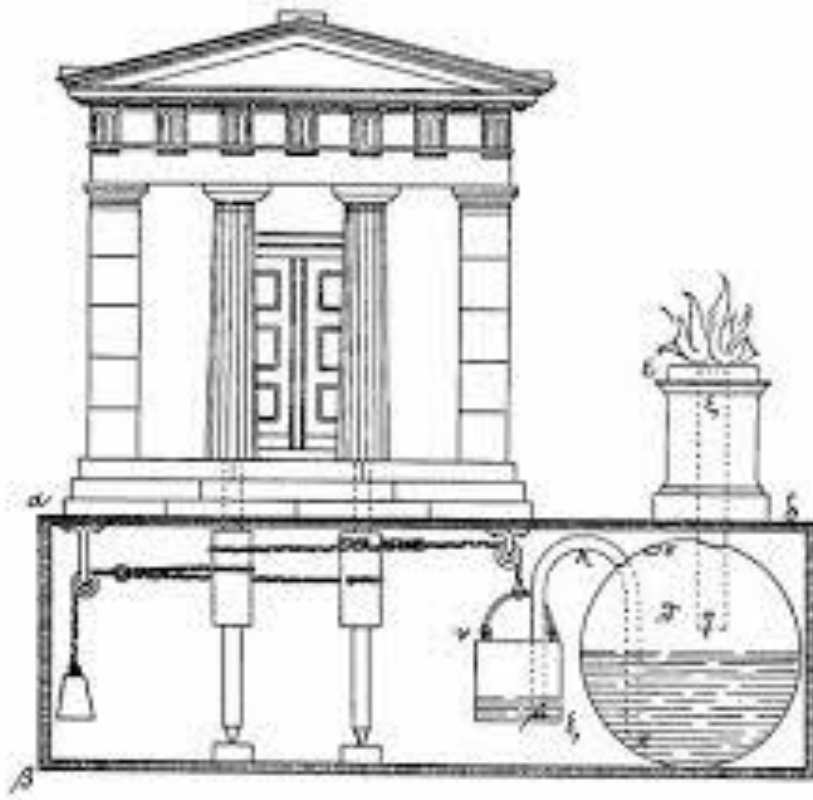
- «Κάθε σύστημα αποτελείται, συνίσταται από στοιχεία αντίθετα και ανόμοια.
- Τα στοιχεία αυτά συντίθενται, συνενώνονται σε ένα, το άριστον
- Επανέρχονται, καταλήγουν, επαναφέρονται στον κοινό στόχο



Ορισμός συστήματος

### 3.3. Αυτόματες πύλες ναού

Από τους αυτοματισμούς και την τεχνογνωσία που άλλαξαν το ρού της ιστορίας ήταν η αξιοποίηση της διαστολής του θερμαινόμενου αέρα .Ο Ήρων εκμεταλευόμενος την ιδιότητα αυτή αυτοματοποίησε το άνοιγμα των πυλών ενός ναού δίνοντας του και στοιχεία θεάματος και μεταφυσικού .Ο μηχανισμός λειτουργούσε ως εξής.Κατασκευάζεται ναός , έτσι ώστε μόλις γίνει η θυσία στο βωμό που βρίσκεται στην είσοδο του ναού,οι πόρτες του ναού να ανοίγουν αυτόματα,και μόλις σβήσει η φωτιά στο βωμόοι πόρτες να κλείνουν πάλι από μόνες τους.

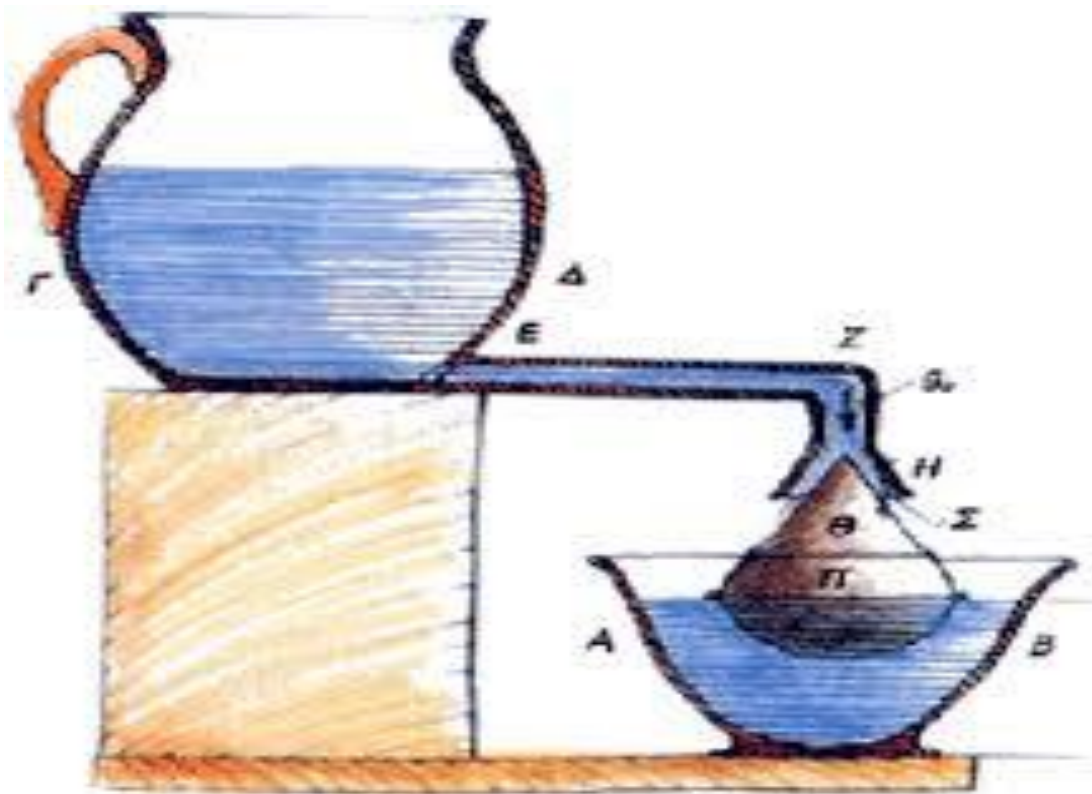


*Αυτόματες πύλες ναού(Ήρων,πνευματικά)*

Η κίνηση επιτυγχάνεται με τη διαστολή του θερμαινόμενου αέρα ,διοχετεύοντας νερό από ένα σταθερό σε ένα κινητό δοχείο.

### 3.4. Υδραυλικά αυτόματα

Ο Κτησίβιος κατασκεύασε ένα δοχείο το οποίο αυτόματα έλεγχε τη στάθμη υγρού με τη βοήθεια ενός κωνικού πλωτήρα ο οποίος με τη σειρά του προσαρμοζόταν ακριβώς σε αντίστοιχο κοίλο κωνικό ακροφύσιο. το ακροφύσιο αυτό είναι συνδεδεμένο μέσω ενός σωλήνα με το μεγάλο δοχείο. Αφαιρώντας κρασί με ένα κύπελο από το δοχείο, ανοίγει η υδραυλική βαλβίδα, κρασί ρέει από το ακροφύσιο στο δοχείο μέχρι να συμπληρωθεί το κρασί που αφαιρέθηκε και να αποκατασταθεί η στάθμη του υγρού στο δοχείο



*έλεγχος στάθμης υγρού με κωνικό πλωτήρα κατά τον Κτησίβιο*



### 3.5. Ιστορία απομακρυσμένου ελέγχου

Αν και ο απομακρυσμένος έλεγχος είναι πιο στενά συνδεδεμένος με τον έλεγχο συσκευών όπως η τηλεόραση και το air-condition ,προυπήρχε των συσκευών αυτών.

Ένας εκ των πρωτοπόρων ήταν ο καταξιωμένος Σερβο-Αμερικάνος εφευρέτης Nikola Tesla ο οποίος κατασκεύασε ένα από τα πρώτα ασύρματα συστήματα απομακρυσμένου ελέγχου το οποίο αποκάλυψε στο Madison square garden της Νέας Υόρκης το 1898.Ονόμασε το πρωτοπόρο του αυτό σύστημα ,το οποίο μπορούσε να χρησιμοποιηθεί για να ελέγξει ένα μεγάλο έυρος μηχανημάτων,τηλεαυτοματισμό.Για την παρουσίασή του ο Tesla χρησιμοποίησε ένα μικρό σκάφος το οποίο ελεγχόταν με ραδιοκύματα.Για την επίτευξη αυτής της απομακρυσμένης επικοινωνίας το μικρό αυτό σκάφος είχε μια μικρή μεταλική κεραία η οποία μπορούσε να λάβει μόνο μία ραδιοσυχνότητα .

Ο Tesla έστειλε σήματα στο καράβι χρησιμοποιώντας ένα κουτί –η εκδοχή του για τον απομακρυσμένο έλεγχο-εξοπλισμένο με ένα μοχλό και ένα τηλεγραφικό εργαλείο το οποίο ήταν αρχικά σχεδιασμένο να στέλνει σήματα morse.Τα σήματα που παράγονταν από αυτό το κουτί ανοιγόκλειναν ηλεκτρικές επαφές πάνω στο καράβι , τα οποία με τη σειρά τους,ρύθμιζαν την προπέλα και το πηδάλιο, επιτρέποντας στον χειριστή να ελέγχει την κίνηση του πλοιαρίου.

Σύντομα μετά από αυτήν την πρωτοπορία του Tesla ,ένας Ισπανός μηχανικός ονόματι Leonardo Torres-Quevedo χρησιμοποίησε ασύρματους τηλεγραφικούς πομπούς αρχικά για τον έλεγχο τρικύκλων και έπειτα μηχανοκίνητα καράβια ακόμα και υποβρύχιες τορπίλες

Οι δουλεία αυτών των 2 εφευρετών ήταν ο προάγγελος των πραγμάτων που ακολούθησαν.Στον πρώτο παγκόσμιο πόλεμο,το γερμανικό ναυτικό χρησιμοποίησε τηλεκατευθυνόμενα σκάφη με εκρηκτικά για να επιτεθούν στους αντιπάλους τους.Στον δεύτερο παγκόσμιο πόλεμο Γερμανοί και Αμερικάνοι χρησιμοποίησαν τηλεκατευθυνόμενες οβίδες και τορπίλες.

## 4. Internet Of things

### 4.1. Τι είναι το Internet of Things ;

Το ΙΟΤ στην πραγματικότητα είναι η επικοινωνία μιας συσκευής με μια άλλη, γεγονός που μας επιτρέπει να έχουμε ένα τεράστιο εύρος ελέγχου με πολυποίκιλα πλεονεκτήματα. Υπολογίζεται ότι αναλογούν 0.8 έξυπνες συσκευές ανα άνθρωπο στον πλανήτη ως τώρα ενώ αναμένεται να εκτιναχθεί στις 3 μέχρι το τέλος του 2020. Μπορεί να βρει εφαρμογή σε όλο το φάσμα της καθημερινότητας του μέσου καταναλωτή όπως και στην βιομηχανία.

### 4.2. Έξυπνα σπιτια

Το έξυπνο σπίτι του μέλλοντος κινείται κυρίως σε 3 άξονες. Την χρήση των πόρων, την ασφάλεια και την βολή που θα προσφέρει. Αποσκοπεί στην μείωση δαπανών καθώς και την επίτευξη καλύτερων βιοτικών επιπέδων. Όσο αφορά την ασφάλεια θα μπορεί να ανιχνεύει και να προλαμβάνει πυρκαγιές και εισβολές από επίδοξους ληστές. Καινούρια πεδία και θέσεις εργασίας ανοίγουν σε τομείς όπως

- Πάροχοι διαδικτύου
- Κατασκευαστές συσκευών
- Τηλεπικοινωνιακοί φορείς
- Πάροχοι οπτικοακουστικών μέσων
- Εταιρείες παροχής υπηρεσιών ασφαλείας



### 4.3. Έξυπνη πόλη

Το ΙΟΤ ανοίγει ένα φωτεινό παράθυρο στην καθημερινότητα δισεκατομμυρίων ανθρώπων που συμβιώνουν στο άστυ. Μιά έξυπνη πόλη ορίζεται ως μια πόλη που παρακολουθεί και ενσωματώνει τις συνθήκες όλων των κρίσιμων υποδομών της, συμπεριλαμβανόμενων των δρόμων, γέφυρες, σήραγγες, σιδηροδρομικές / υπόγειες διαβάσεις, αεροδρόμια, λιμάνια, επικοινωνίες, φυσικοί πόροι. Προηγμένα συστήματα παρακολούθησης και έξυπνοι αισθητήρες συλλέγουν δεδομένα και τα αξιοποιούν σε πραγματικό χρόνο με στόχο την λήψη αποφάσεων για την διαχείριση της πόλης. Ένας από τους τομείς που χρίζει διευθέτησης και το ΙΟΤ έρχεται να δώσει λύση είναι η μεταφορές. Έξυπνοι φωτεινοί σηματοδότες λαμβάνοντας δεδομένα από αισθητήρες στις οδικές αρτηρίες αντιλαμβανόμενοι την φόρτιση τους θα βελτιστοποιούν την κίνηση στην πόλη και θα ελαχιστοποιούν τον χρόνο αναμονής σε αυτούς. Οι δυνατότες δεν σταματούν μόνο στις μεταφορές αλλά και στην διαχείριση κρίσεων. Για παράδειγμα, οι πόροι μπορούν να δεσμευτούν πριν από μια μεγάλη διακοπή νερού για συντήρηση του υδροδοτικού δικτύου, τα συνεργεία που ρίχνουν αλάτι αποστέλλονται μόνο όταν μια συγκεκριμένη γέφυρα έχει αρχίσει να παγώνει και η χρήση των επιθεωρητών μειώνεται, γνωρίζοντας την κατάσταση της ζωής όλων των υποδομών.

Τεχνολογίες, όπως έξυπνη μέτρηση, τα ασύρματα δίκτυα αισθητήρων, οι ανοικτές πλατφόρμες, οι ευρυζωνικές υπηρεσίες υψηλής ταχύτητας και το Cloud computing είναι βασικά δομικά στοιχεία της υποδομής μιας έξυπνης πόλης



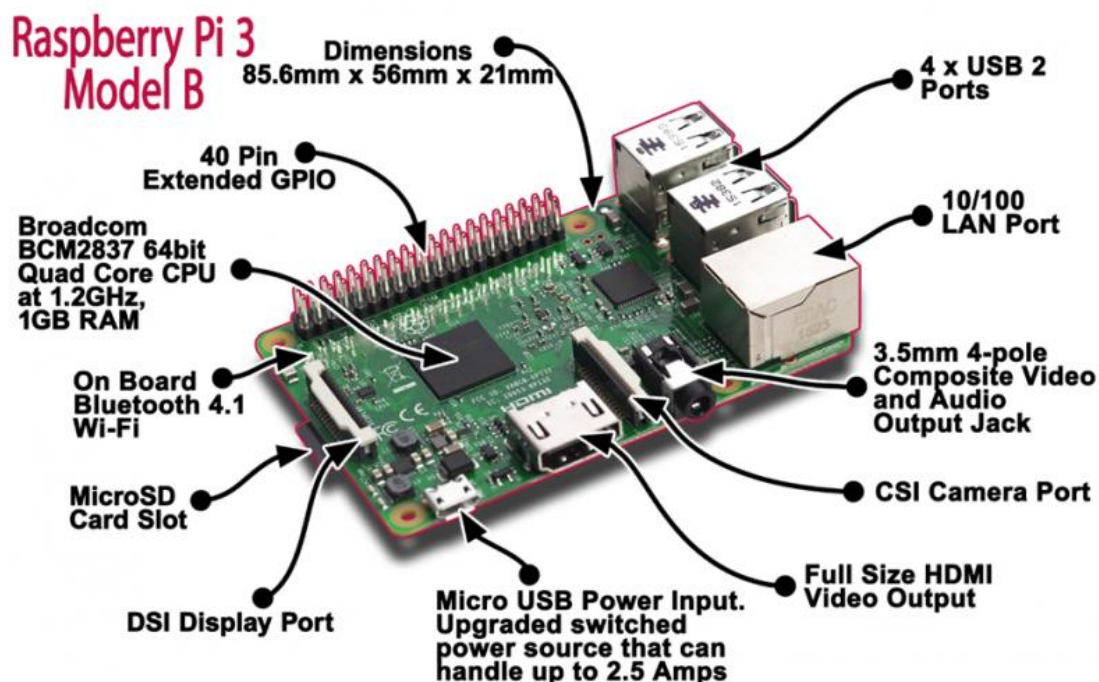
#### 4.4. Internet of Medical Things (IOMT)

Το ΙοMT επιτρέπει αλληλεπίδραση μεταξύ μηχανών και επεμβατικές λύσεις σε πραγματικό χρόνο, που ριζικά μεταμορφώνει την παροχή υγειονομικής περίθαλψης όπως και οικονομικής προσιτότητας και αξιοπιστίας. Κύρια πλεονεκτήματα είναι:

- Ο απομακρυσμένος έλεγχος σε πραγματικό χρόνο της υγείας των ασθενών, αποφεύγοντας άσκοπες επισκέψεις σε γιατρούς, ενώ και η φροντίδα στο σπίτι θα αποτελεί στο μέλλον επιλογή.
- Βασιζόμενοι σε μια περιεκτική βάση γνώσεων οι οποίες έχουν συγκεντρωθεί από προηγούμενα κρούσματα της νόσου και από αποδεδειγμένη έρευνα, οι γιατροί μπορούν να χρησιμοποιήσουν ιατρική βασισμένη σε συγκεκριμένα στοιχεία για τη βελτίωση της υγείας των ασθενών τόσο σε επίπεδο επεμβατικό όσο και προληπτικό.
- Η ευεξία και η πρόληψη θα είναι εφικτή με συστήματα παρακολούθησης για τη διατροφή, τη σωματική δραστηριότητα και την ποιότητα ζωής. Επανάστατικές συσκευές όπως, wearable devices, εμφυτεύσιμα τσιπ και ενσωματωμένα συστήματα σε βιοϊατρικές συσκευές, παρακολουθούν συνεχή δεδομένα σχετικά με τη δραστηριότητα των ασθενών.
- Η συντήρηση των εμβολίων με αξιόπιστα συστήματα ελέγχου. Ενώ τα εμβόλια σώζουν εκατομμύρια ζωές κάθε χρόνο και είναι από τις πιο οικονομικές και από τις αποτελεσματικές επεμβάσεις υγείας που έχουν αναπτυχθεί, περίπου 1,5 εκατομμύριο παιδιά πεθαίνουν κάθε χρόνο από ασθένειες που προλαμβάνονται με εμβολιασμό σύμφωνα με τον Παγκόσμιο Οργανισμό Υγείας. Μερικοί παράγοντες που συμβάλλουν στη διαθεσιμότητα των εμβολίων παγκοσμίως, περιλαμβάνουν αναξιόπιστα συστήματα μεταφοράς και διαλείπουσες εγκαταστάσεις αποθήκευσης, τα οποία καθιστούν δύσκολη τη διατήρηση εμβολίων υψηλής ποιότητας που απαιτούν ψύξη.



## 5. Raspberry PI



Το Raspberry PI ξεκίνησε ως μια λύση για την εκμάθηση των ηλεκτρονικών υπολογιστών στις αναπτυσσόμενες χώρες . Η επαναστατικότητα του και επομένως ή επιτυχία του έγκειται στην κατασκευή ενός ικανοποιητικής ισχύος ολοκληρωμένου υπολογιστή (μικροεπεξεργαστής,μνήμη,είσοδος-έξοδος) σε μια μόνο πλακέτα.Η υπολογιστική ταχύτητα των επεξεργαστών ποικίλει από 700MHz έως 1.4GHz για το Raspberry pi B+ . Η μνήμη της πλακέτας φτάνει ως το 1GB ram ενώ έχει υποδοχή για SD(Secure Digital) card που χρησιμοποιούνται για να αποθηκεύσουν το λειτουργικό σύστημα και την μνήμη του προγράμματος είτε σε SDHC είτε σε MicroSDHC. Η έκδοση που χρησιμοποιήσαμε για να διεξαγάγουμε όλα τα πειράματά μας ήταν η Raspberry Pi 3 Model B. Παρά το μικρό όγκο του, το Raspberry Pi διαθέτει επεξεργαστή ARMv8 τετραπλού πυρήνα (64 bit) που χρονομετρείται στα 1200MHz, έναν πυρήνα γραφικών 3D VideoCore IV με θύρα Ethernet, διασύνδεση δικτύου 802.11n ασύρματου δικτύου LAN, συνδυασμένη υποδοχή ήχου 3.5 mm και σύνθετο βίντεο, διασύνδεση κάμερας (CSI), θύρα USB, , Bluetooth 4.1, υποδοχή κάρτας Micro SD (push-pull αντί push-push στην προηγούμενη έκδοση) και 40 ακροδέκτες γενικής χρήσης για σύνδεση με άλλα ηλεκτρονικά και περιφερειακά (pin GPIO). Το λειτουργικό σύστημα αποθηκεύεται στην κάρτα micro SD που πρέπει να έχει μέγεθος τουλάχιστον 4 GB και τουλάχιστον επίπεδο 4. Στην αγορά διατίθενται ήδη κάρτες micro SD προς πώληση με φορτωμένο το λειτουργικό σύστημα αλλά στην παρούσα εργασία και για μείωση κόστους όσο και για εκπαιδευτικούς σκοπούς επιλέχθηκε να εγκατασταθεί από τον συντάκτη της. Για το σκοπο αυτό, χρειάστηκε ένας προσαρμογέα για να συνδεθεί η κάρτα micro SD στον υπολογιστή .



Για την σύνδεση των περιφερειακών επίσης χρειάστηκαν ένα καλώδιο HDMI, ένα καλώδιο Ethernet, ένα ποντίκι, ένα πληκτρολόγιο και μια οθόνη. Μετά τη συγκέντρωση όλων αυτών των απαραίτητων στοιχείων, η εγκατάσταση του λογισμικού μπορεί να προχωρήσει.

### 5.1. Εφαρμογές Raspberry PI

Με το raspberry PI μπορούν να εκτελεστούν όλες οι βασικές λειτουργίες ενός σύγχρονου υπολογιστή, επιτρέποντας σε έναν άπειρο χρήστη να περιηγηθεί στο διαδίκτυο, να μεταδίδει το περιεχόμενό του και να πειραματιστεί ακόμα και στον προγραμματισμό ( που ήταν και αρχική σύλληψη της ιδέας για την κατασκευή της πλακέτας) .Κάποιες από τις πιο ευρέως χρησιμοποιούμενες εφαρμογές της συσκευής είναι η παρακάτω .

### 5.2. Προσωπικός υπολογιστής(PC)

Όπως αναφέρθηκε το Raspberry PI έχει επαρκή δύναμη για να καλύψει τις ανάγκες του μέσου ,καθημερινού χρήστη με κύριο πλεονέκτημα την φορητότητα του για την μετατροπή του σε προσωπικό υπολογιστή. Συνδέοντας περιφερειακά όπως οθόνη, πληκτρολόγιο και ποντίκι το raspberry PI μεταμορφώνεται απλά και γρήγορα σε έναν προσωπικό υπολογιστή δίνοντας ακόμα την δυνατότητα λήψης και εγκατάστασης των Windows 10,νομίμως και χωρίς χρέωση .



### 5.3. Μηχανή για torrent



Εκείνοι που συμμετέχουν ενεργά στην torrent κοινότητα μπορούν να χρησιμοποιήσουν το Raspberry Pi ως αποκλειστικό μηχάνημα για τον διαμοιρασμό των torrent. Καταναλώνει πολύ λίγη ενέργεια, ώστε να μπορείτε να λειτουργείτε 24 ώρες το 24ωρο και χωρίς να περιορίζει τους πόρους του υπολογιστή σας.

### 5.4. Αναπαραγωγή πολυμέσων

Μια πολύ δημοφιλής χρήση του Raspberry Pi είναι η αναπαραγωγή πολυμέσων καθώς αυτό μπορεί απλούστατα να επιτευχθεί με την σύνδεση του απλά με μια τηλεόραση. Με αυτόν τον τρόπο ο τελικός χρήστης είναι έτοιμος να απολάυσει ταινίες και να διαχειριστεί το περιεχόμενο των πολυμέσων με τρόπο φιλικό προς τον χρήστη. Για τον σκοπό είναι προσαρμοσμένη η διανομή OSMC Open Source Linux με ποιοτικό περιεχόμενο.



## 5.5. Gaming



Με μια γρήγορη περιήγηση στο γραφικό περιβάλλον ο χρήστης μπορεί να βρεί εγκατεστημένα παιχνίδια όπως το funny boat και το Minecraft . Ωστόσο δεν είναι μόνο αυτά που κάποιος μπορεί να απολάυσει στην πλακέτα.Με την πληκτρολόγηση της εντολής `$ apt-cache search games` θα εμφανιστεί μια ατελείωτη λίστα παιχνιδιών ως έξοδος.Ο τίτλος του παιχνιδιού είναι η πρώτη πρόταση πριν το σύμβολο "-" .Αφού βρεί ο χρήστης όποιο τον ενδιαφέρει δεν μένει παρά μόνο να το εγκαταστήσει με την εντολή `$ sudo apt -get install «ονομα παιχνιδιού απτη λίστα»` .

Μερικά απο τα πιο δημοφιλή είναι τα παρακάτω

- Minecraft PI edition
- Quake 3
- Open Arena (command `sudo apt-get install Openarena`)
- Flare (command `sudo apt-get install flare-game`)
- Battleball-a 3D tank arcade (command `sudo apt-get install battleball`)
- Tint (command `sudo apt-get install tint`)
- Freesweep (command `sudo apt-get install freesweep`)
- Gnobe Nibbles (command `sudo apt-get install gnome-nimble`)



## 6. Εγκατάσταση λειτουργικού συστήματος στο Raspberry

### 6.1. Rasbian-Noobs

Το λειτουργικό σύστημα Rasbian-Noobs είναι ένα λειτουργικό σύστημα παρεμφερές του Debian(Διανομής Linux). Η επιλογή δεν έγινε τυχαία, καθώς αυτό το λειτουργικό ενδείκνυται σε εφαρμογές του Raspberry που θέλουν να κάνουν χρήση του προεγκατεστημένου του λογισμικού για προγραμματισμό και εκπαίδευση. Διατίθεται όπως προαναφέρθηκε δωρεάν .

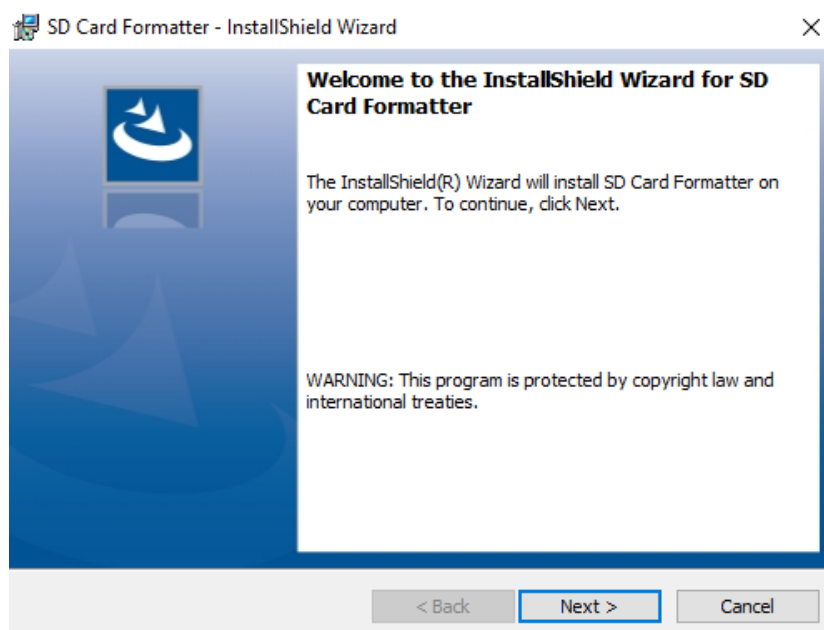
Κατεβάζοντας το λειτουργικό σύστημα και συνδέοντας την κάρτα Micro SD με το προσαρμογέα στο υπολογιστή μας ο χρήστης είναι σε θέση να διαμορφώσει την κάρτα με τρόπο που προτείνει το ίδιο λογισμικό.

YOU ACKNOWLEDGE AND AGREE THAT YOU HAVE READ THIS AGREEMENT AND INTEND TO BE BOUND AS IF YOU HAD SIGNED THIS AGREEMENT IN WRITING. IF YOU ARE ACTING ON BEHALF OF AN ENTITY, YOU WARRANT THAT YOU HAVE THE AUTHORITY TO ENTER INTO THIS AGREEMENT ON BEHALF OF SUCH ENTITY AND BIND SUCH ENTITY TO THE TERMS OF THIS AGREEMENT.

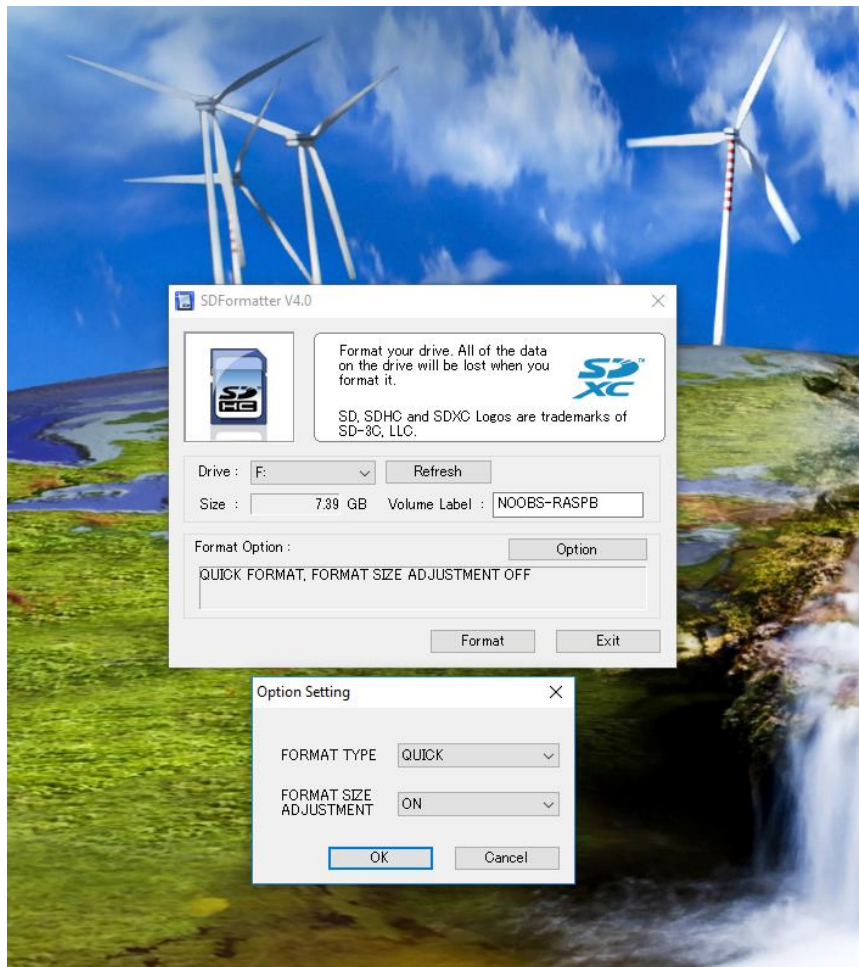


Εγκαθιστούμε την εφαρμογή κάνοντας κλικ στο κουμπί "Accept" στο κάτω μέρος της σελίδας, όπως φαίνεται στην παραπάνω εικόνα.

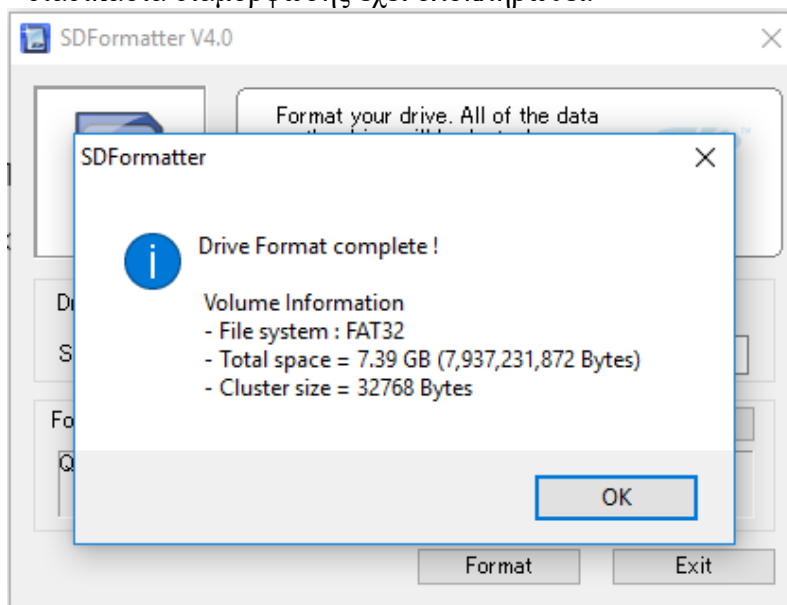
Στη συνέχεια, ο χρήστης εκτελεί το αρχείο "**SD\_CardFormatter.exe**", εγκαθιστά και τέλος τρέχει την εφαρμογή .



Ξεκινώντας το πρόγραμμα, ο χρήστης κάνει κλικ στο κουμπί "Option" Στο παράθυρο της εφαρμογής και αλλάζει τη μορφή προσαρμογής μεγέθους σε "ON".




Στη συνέχεια, κάνει κλικ στο κουμπί **Format** και σε λίγα δευτερόλεπτα η διαδικασία διαμόρφωσης έχει ολοκληρωθεί.



Μετά την περάτωση της διαμόρφωσης της κάρτας SD card μας, κατεβάζουμε την τελευταία έκδοση του Noobs, είτε απευθείας από τον ιστότοπο με τη μορφή συμπιεσμένου αρχείου .zip είτε μέσω Torrent.

**NOOBS Lite** contains the same operating system installer without Raspbian installed. It provides the same operating system selection menu allowing you to select Raspbian and other images to be downloaded and installed.

The NOOBS files contained in the ZIP archive are over 4GB in size, which means that these archives use features which are not supported by older unzip programs on some platforms. If you find that the download appears to be corrupt or that you cannot unzip the files correctly, please try using [7Zip](#) (Windows) or [The Unarchiver](#) (Macintosh). Both are free of charge and have been tested to unzip the files correctly.



**NOOBS**  
Offline and network install

Version: 2.3.0  
Release date: 2017-03-03

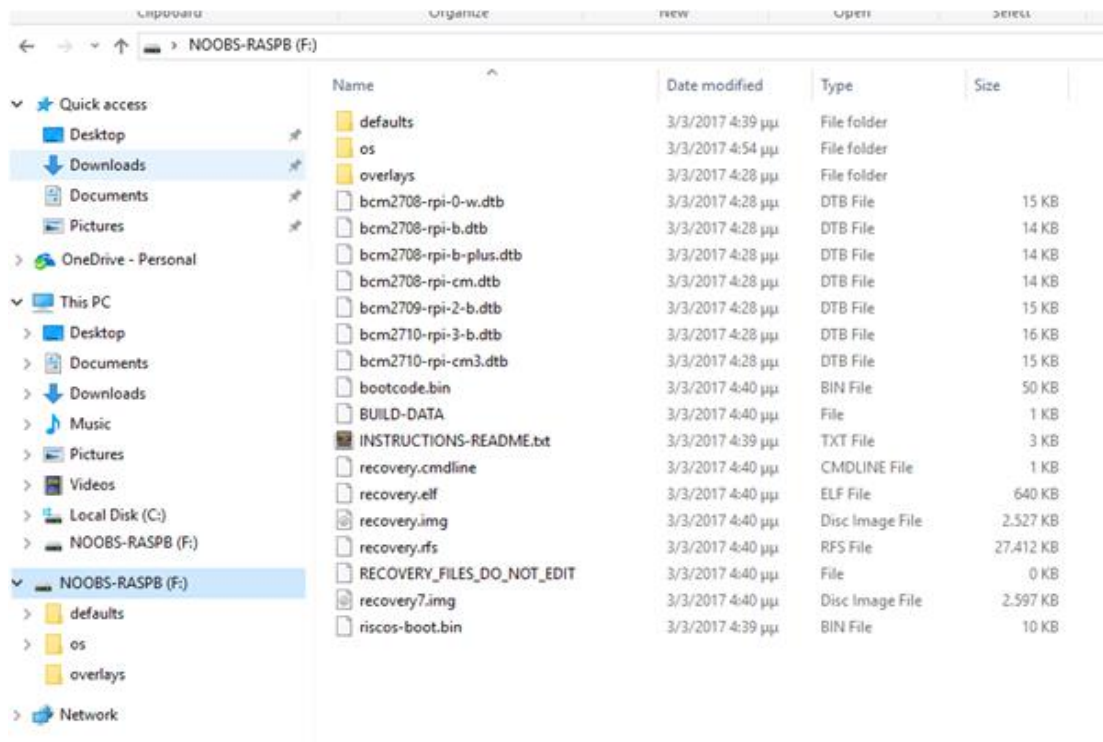
[Download Torrent](#) [Download ZIP](#)

SHA-1: 9bb4b421a9a39ecbe75701de35ba9a3524b801c9

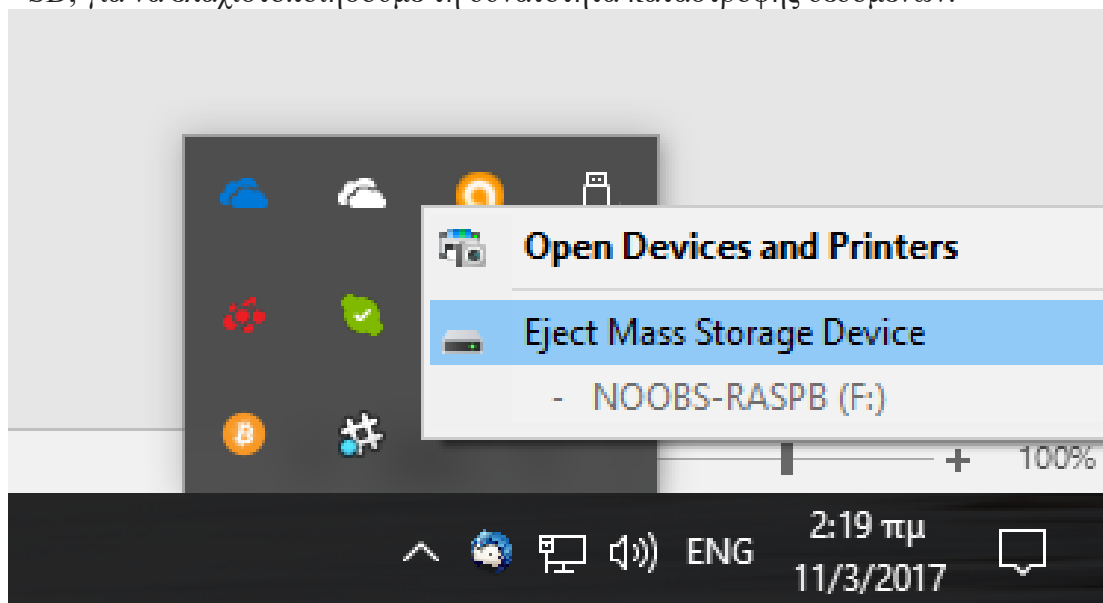
Note: Raspbian and NOOBS contain Java SE Platform Products, licensed to you under the Oracle License Agreement available [here](#).

## 6.2. Εγκατάσταση του λογισμικού

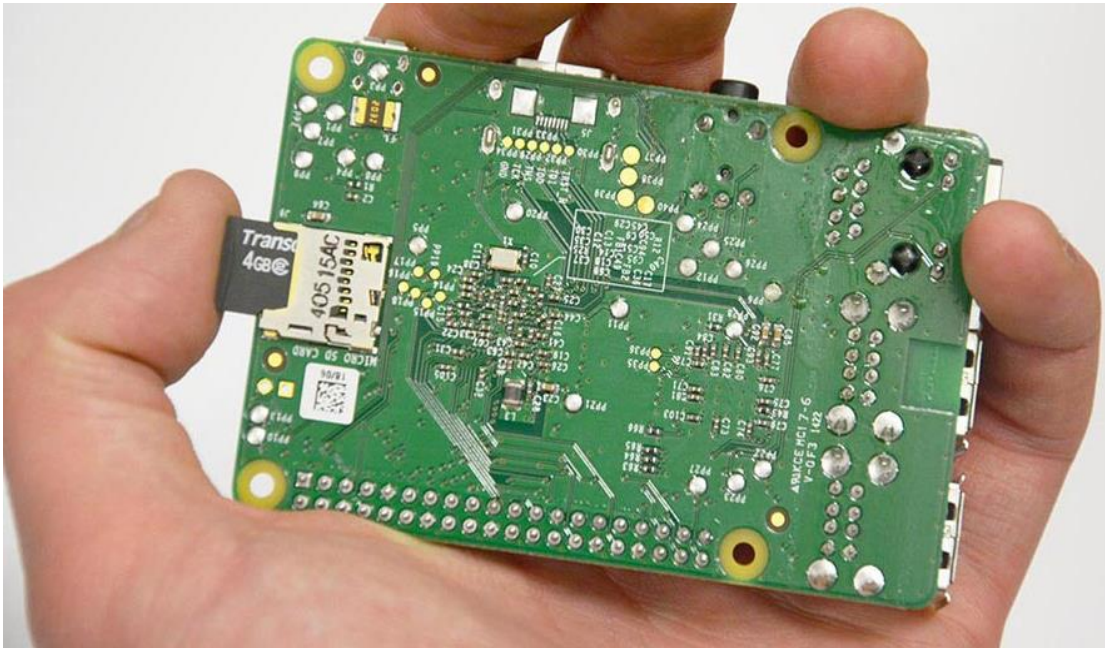
Μετά τη λήψη του συμπιεσμένου αρχείου .zip του Noobs, εξάγουμε όλο το περιεχόμενό του σε ένα νέο φάκελο. Στη συνέχεια περιηγούμε σε αυτόν τον φάκελο και αντιγράφουμε όλα τα αρχεία στην κάρτα SD μας



Αφού ολοκληρωθεί η διαδικασία αντιγραφής, αφαιρούμε με ασφάλεια την κάρτα SD, για να ελαχιστοποιήσουμε τη δυνατότητα καταστροφής δεδομένων.

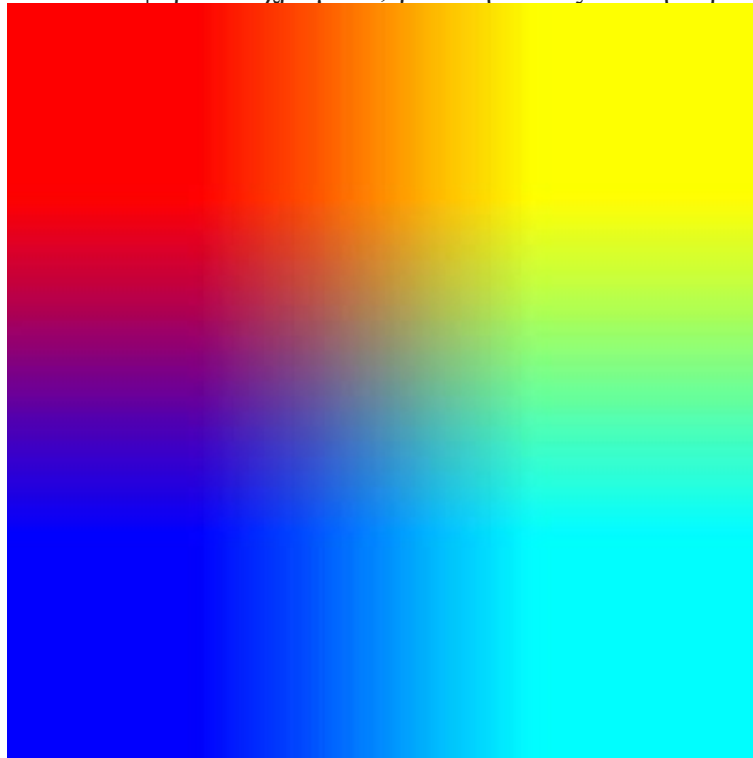


Μετά την εγκατάσταση του λογισμικού στην κάρτα MicroSD, ο χρήστης είναι σε θέση να την τοποθετήσει στην πλακέτα του Raspberry Pi. Μετά τη σύνδεση ενός ποντικιού και ενός πληκτρολογίου στο Raspberry Pi χρησιμοποιούμε ένα καλώδιο HDMI για τη σύνδεση με την οθόνη. Τώρα είμαστε έτοιμοι να βάλουμε το τροφοδοτικό. Όπως θα παρατηρήσετε, δεν υπάρχει διακόπτης On / Off στο Raspberry Pi. Πρέπει να το βάλουμε στην πρίζα και θα ξεκινήσει άμεσα.

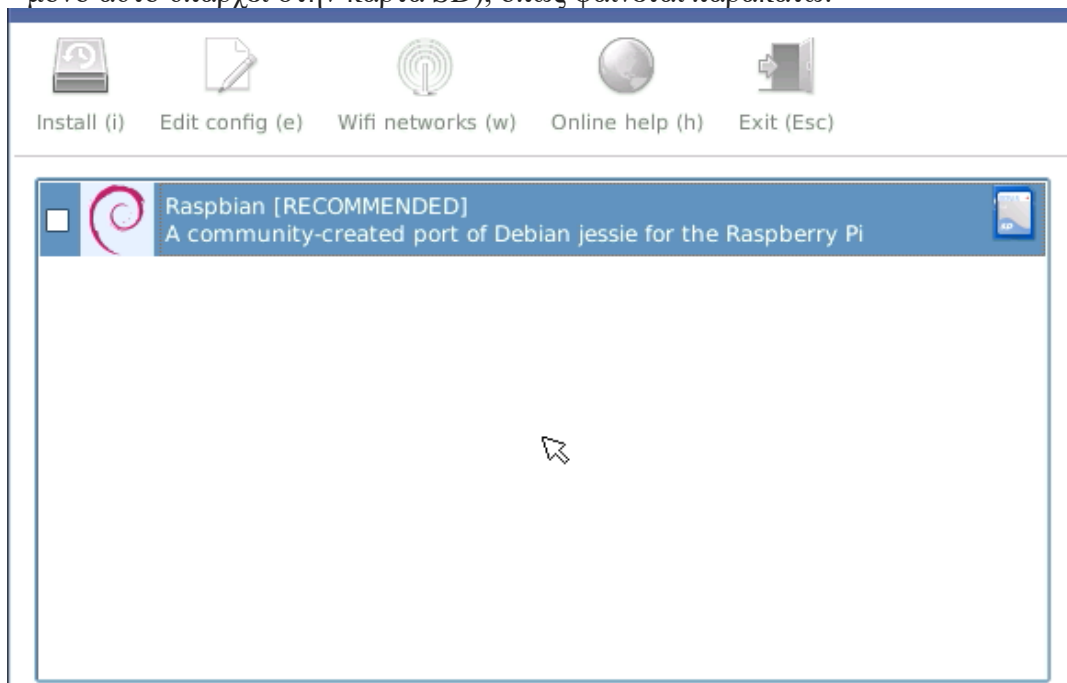


## 7. Εκκίνηση του Raspberry PI , πρώτη επαφή

Κατά την εκκίνηση του Raspberry Pi, το πρώτο πράγμα που βλέπει ο χρήστης είναι μια οθόνη με πολλά διαφορετικά χρώματα, γνωστή και ως οθόνη ουράνιο τόξο.



Το Raspbian θα είναι το μόνο λειτουργικό σύστημα που υπάρχει στη λίστα (αφού μόνο αυτό υπάρχει στην κάρτα SD), όπως φαίνεται παρακάτω.



Τσεκάροντας την επιλογή Raspbian και κάνοντας κλικ στο Install, το σύστημα μας προειδοποιεί ότι θα διαγραφούν όλα τα περιεχόμενα της κάρτας SD.

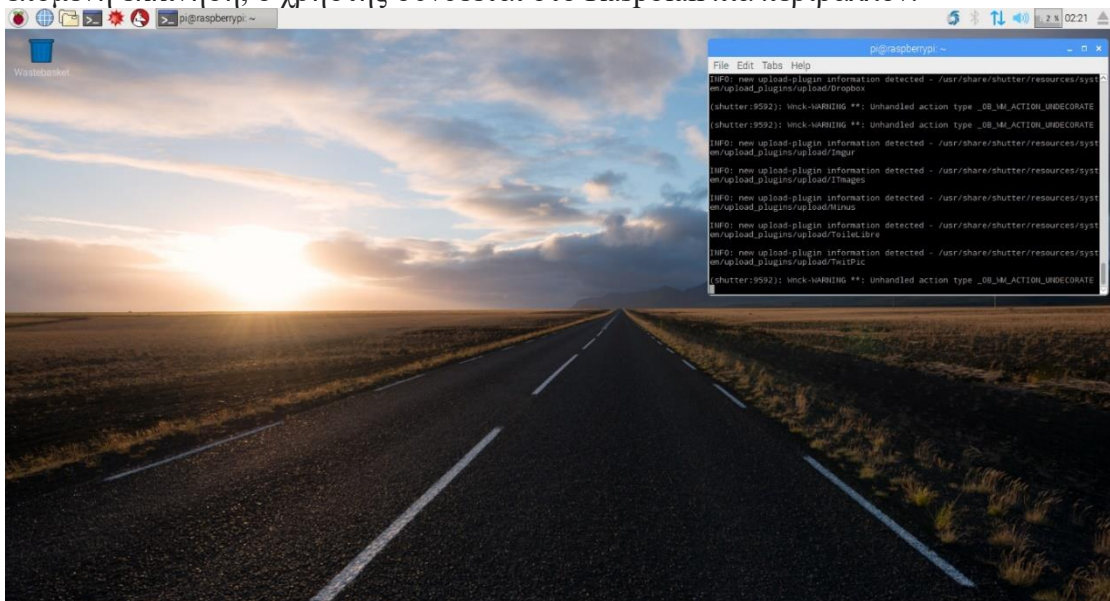


Μετά την σχετική επιβεβαίωση η εγκατάσταση ξεκινάει. Θα χρειαστεί λίγος χρόνος για να ολοκληρωθεί ανάλογα με την ταχύτητα της κάρτας SD. Μια αργή κάρτα Class 4 έχει ρυθμό μεταφοράς δεδομένων περίπου 1-1,5MB / s, ενώ μια γρήγορη κάρτα μπορεί να φτάσει έως και 5,8MB / s.

Το σύστημα δείχνει το μήνυμα ότι το λειτουργικό σύστημα έχει εγκατασταθεί με επιτυχία.

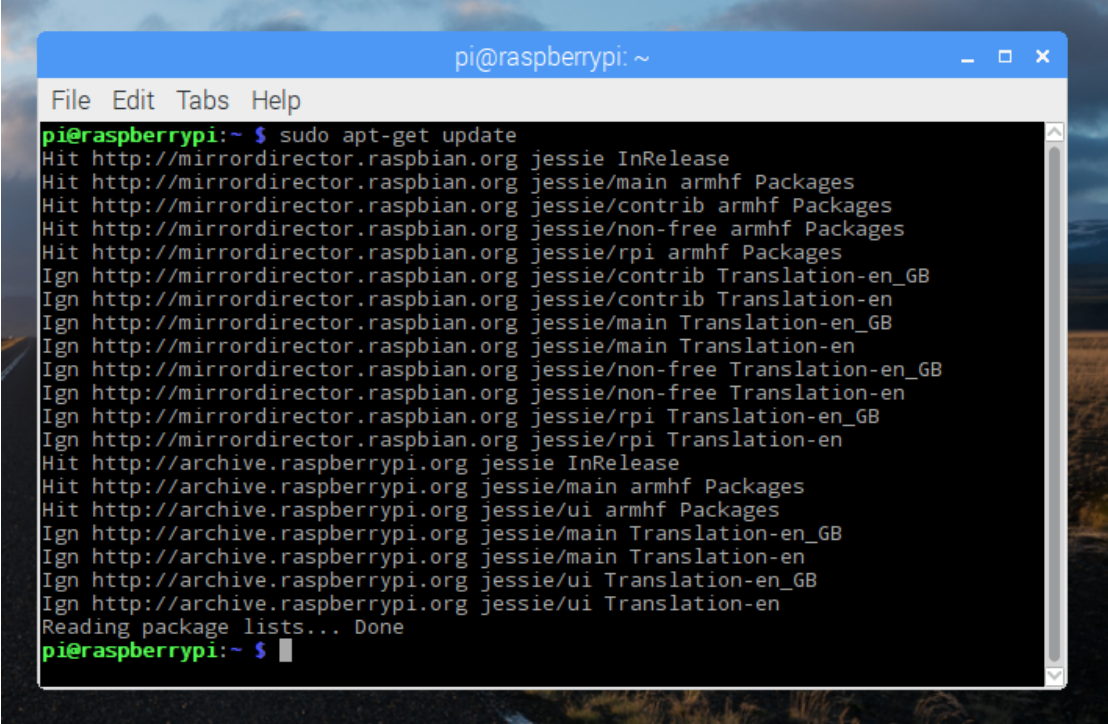


Κάνοντας κλικ στο κουμπί OK, η Raspberry Pi θα ξεκινήσει ξανά. Κατά την επόμενη εκκίνηση, ο χρήστης συνδέεται στο Raspbian πια περιβάλλον.



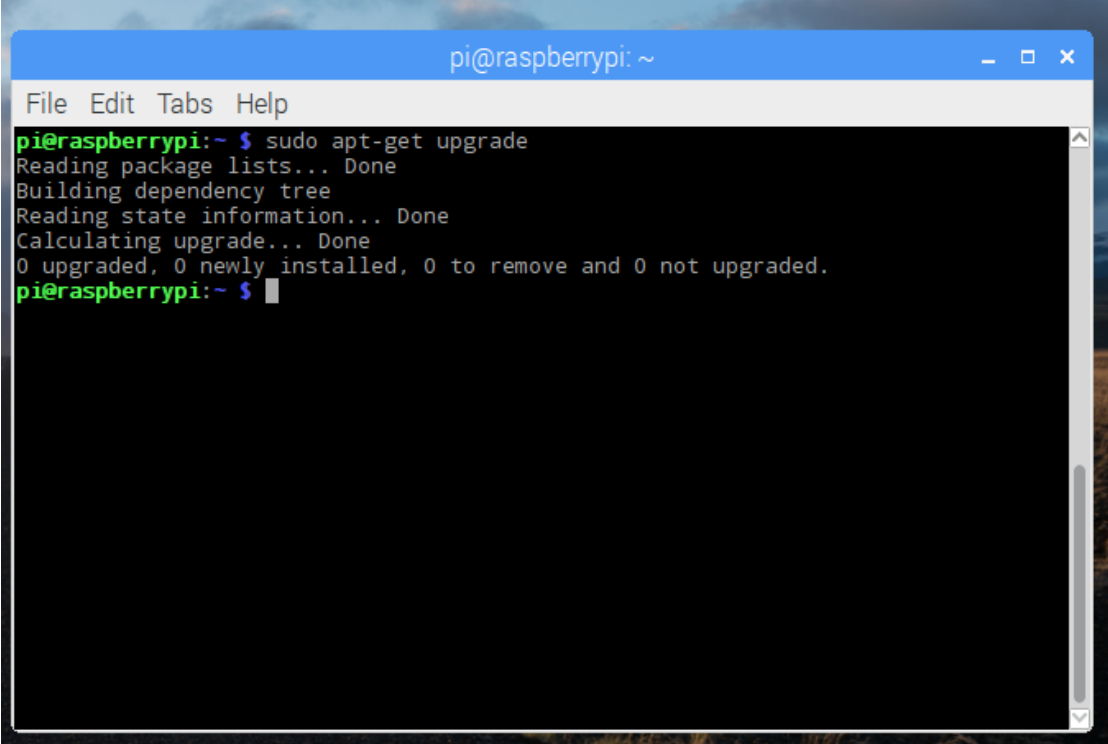
Για την ενημέρωση του περιβάλλοντος linux και την λήψη της τελευταίας έκδοσης των πακέτων ο χρήστης πληκτρολογεί και εκτελεί δύο βασικές εντολές των linux.

- Sudo apt-get update



```
pi@raspberrypi: ~  
File Edit Tabs Help  
pi@raspberrypi:~ $ sudo apt-get update  
Hit http://mirrordirector.raspbian.org jessie InRelease  
Hit http://mirrordirector.raspbian.org jessie/main armhf Packages  
Hit http://mirrordirector.raspbian.org jessie/contrib armhf Packages  
Hit http://mirrordirector.raspbian.org jessie/non-free armhf Packages  
Hit http://mirrordirector.raspbian.org jessie/rpi armhf Packages  
Ign http://mirrordirector.raspbian.org jessie/contrib Translation-en_GB  
Ign http://mirrordirector.raspbian.org jessie/contrib Translation-en  
Ign http://mirrordirector.raspbian.org jessie/main Translation-en_GB  
Ign http://mirrordirector.raspbian.org jessie/main Translation-en  
Ign http://mirrordirector.raspbian.org jessie/non-free Translation-en_GB  
Ign http://mirrordirector.raspbian.org jessie/non-free Translation-en  
Ign http://mirrordirector.raspbian.org jessie/rpi Translation-en_GB  
Ign http://mirrordirector.raspbian.org jessie/rpi Translation-en  
Hit http://archive.raspberrypi.org jessie InRelease  
Hit http://archive.raspberrypi.org jessie/main armhf Packages  
Hit http://archive.raspberrypi.org jessie/ui armhf Packages  
Ign http://archive.raspberrypi.org jessie/main Translation-en_GB  
Ign http://archive.raspberrypi.org jessie/main Translation-en  
Ign http://archive.raspberrypi.org jessie/ui Translation-en_GB  
Ign http://archive.raspberrypi.org jessie/ui Translation-en  
Reading package lists... Done  
pi@raspberrypi:~ $
```

- Sudo apt-get upgrade

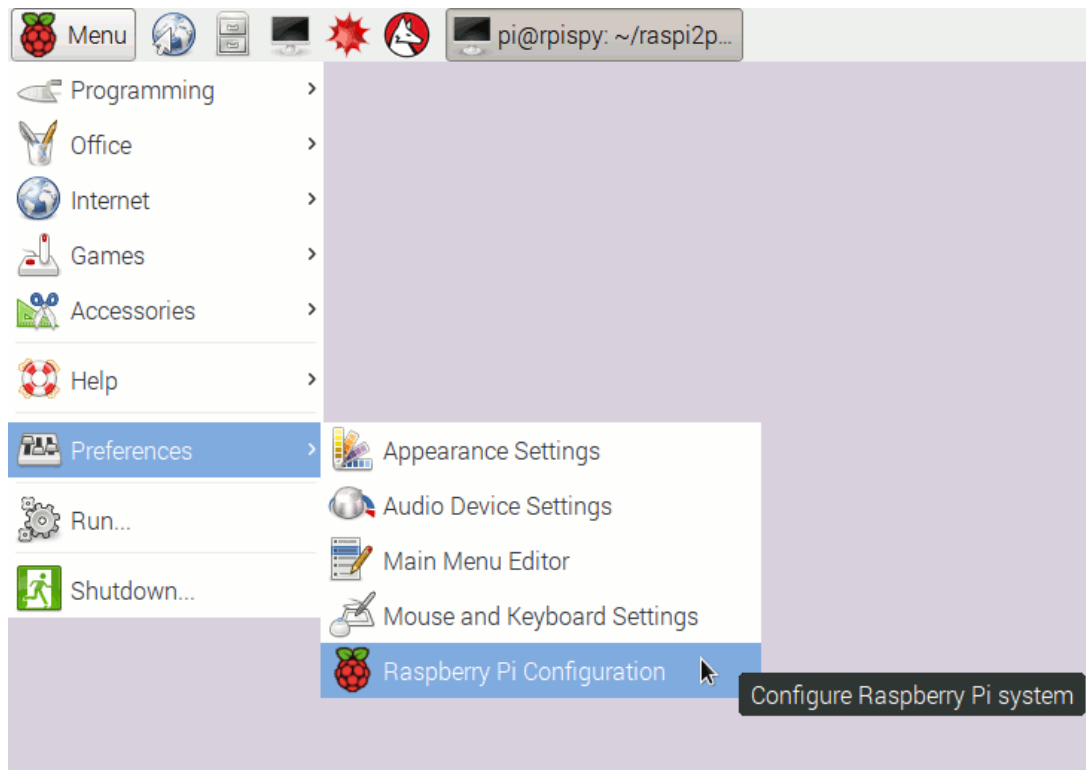


```
pi@raspberrypi: ~  
File Edit Tabs Help  
pi@raspberrypi:~ $ sudo apt-get upgrade  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
Calculating upgrade... Done  
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.  
pi@raspberrypi:~ $
```

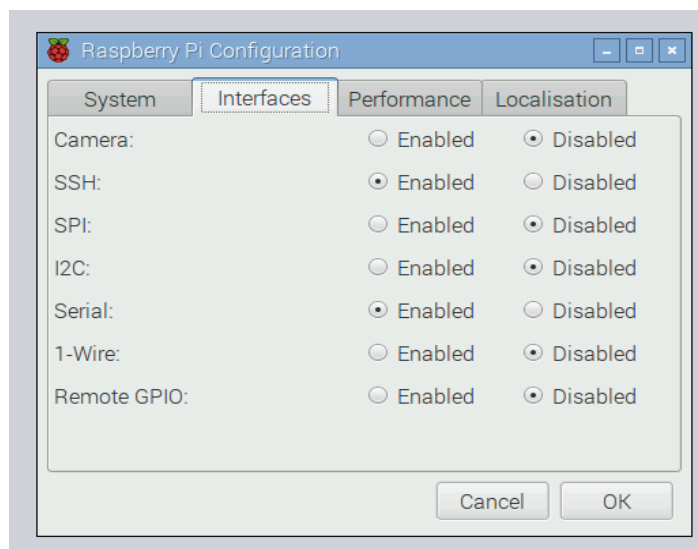


## 7.1. SSH και raspberry PI

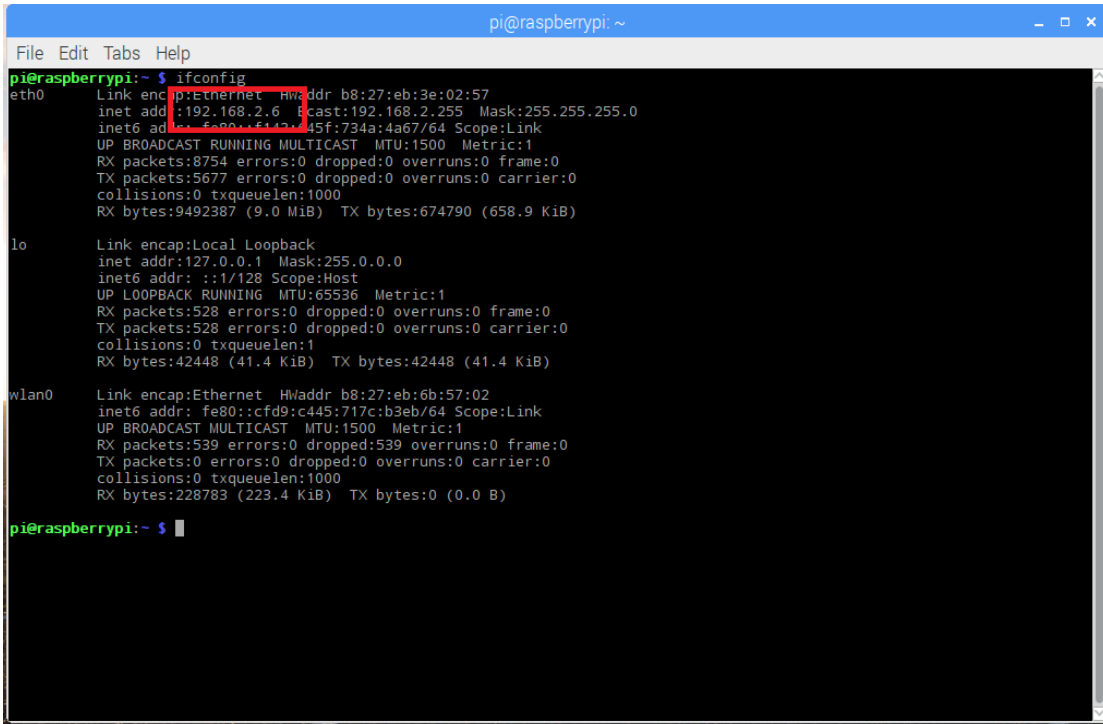
Το SSH (Secure Shell) είναι ένα ασφαλές δικτυακό πρωτόκολλο το οποίο επιτρέπει τη μεταφορά δεδομένων μεταξύ δύο υπολογιστών. Το SSH όχι μόνο κρυπτογραφεί τα δεδομένα που ανταλλάσσονται κατά τη συνεδρία, αλλά προσφέρει ένα ασφαλές σύστημα αναγνώρισης καθώς και άλλα χαρακτηριστικά όπως ασφαλή μεταφορά αρχείων (SSH File Transfer Protocol, SFTP), κλπ. Μετά τη νέα εγκατάσταση του Raspbian παρατηρούμε ότι ο SSH Server στο Raspberry PI είναι απενεργοποιημένος από προεπιλογή. Για την ενεργοποίησή του, ο χρήστης ακολουθεί τα παρακάτω βήματα **Main Menu → Preferences → Raspberry Pi Configuration**



Στη συνέχεια, κάνοντας κλικ στην καρτέλα "Διεπαφές" επιλέγει "Enabled" στο πεδίο "SSH".

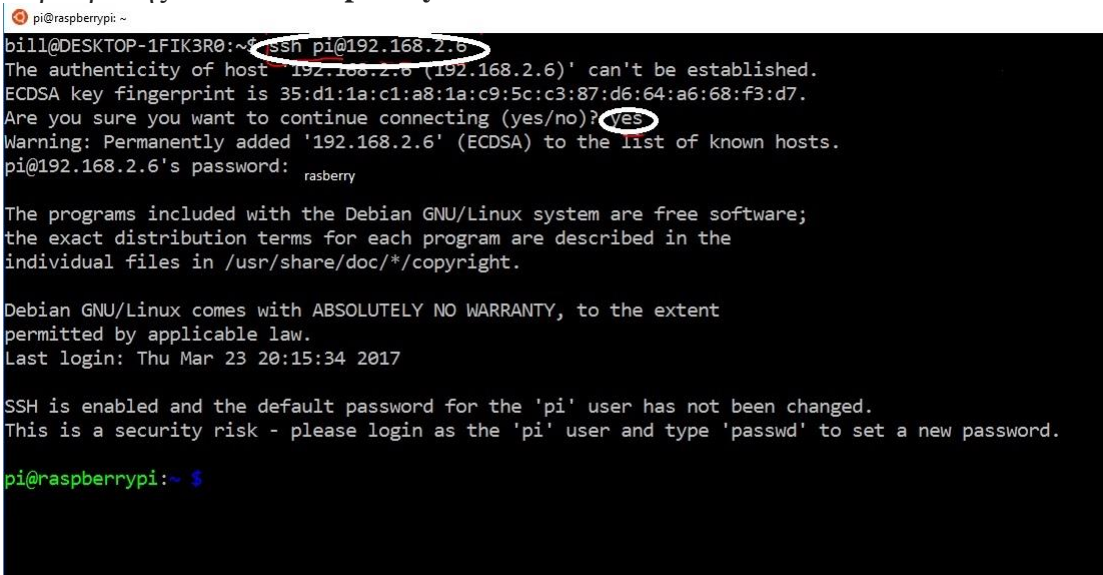


Καταυτόν τον τρόπο επιτυγχάνεται η σύνδεση με το Raspberry Pi από άλλο μηχάνημα μέσω SSH και κατεπέκταση ο **απομακρυσμένος του έλεγχος**. Για να καταφέρει ο χρήστης αυτήν την επικοινωνία πρέπει να γνωρίζει την διεύθυνση IP της συσκευής. Ο χρήστης είναι σε θέση να την γνωρίζει πληκτρολογώντας την εντολή **ifconfig** όπως επισημαίνεται παρακάτω.



```
pi@raspberrypi: ~  
File Edit Tabs Help  
pi@raspberrypi:~$ ifconfig  
eth0      Link encap:Ethernet HWaddr b8:27:eb:3e:02:57  
          inet addr:192.168.2.6 Bcast:192.168.2.255 Mask:255.255.255.0  
          inet6 addr: fe80::cfd9:c445:717c:b3eb/64 Scope:Link  
          UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1  
          RX packets:8754 errors:0 dropped:0 overruns:0 frame:0  
          TX packets:5677 errors:0 dropped:0 overruns:0 carrier:0  
          collisions:0 txqueuelen:1000  
          RX bytes:9492387 (9.0 MiB) TX bytes:674790 (658.9 KiB)  
  
lo        Link encap:Local Loopback  
          inet addr:127.0.0.1 Mask:255.0.0.0  
          inet6 addr: ::1/128 Scope:Host  
          UP LOOPBACK RUNNING MTU:65536 Metric:1  
          RX packets:528 errors:0 dropped:0 overruns:0 frame:0  
          TX packets:528 errors:0 dropped:0 overruns:0 carrier:0  
          collisions:0 txqueuelen:1  
          RX bytes:42448 (41.4 KiB) TX bytes:42448 (41.4 KiB)  
  
wlan0     Link encap:Ethernet HWaddr b8:27:eb:6b:57:02  
          inet6 addr: fe80::cfd9:c445:717c:b3eb/64 Scope:Link  
          UP BROADCAST MULTICAST MTU:1500 Metric:1  
          RX packets:539 errors:0 dropped:539 overruns:0 frame:0  
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0  
          collisions:0 txqueuelen:1000  
          RX bytes:228783 (223.4 KiB) TX bytes:0 (0.0 B)  
  
pi@raspberrypi:~$
```

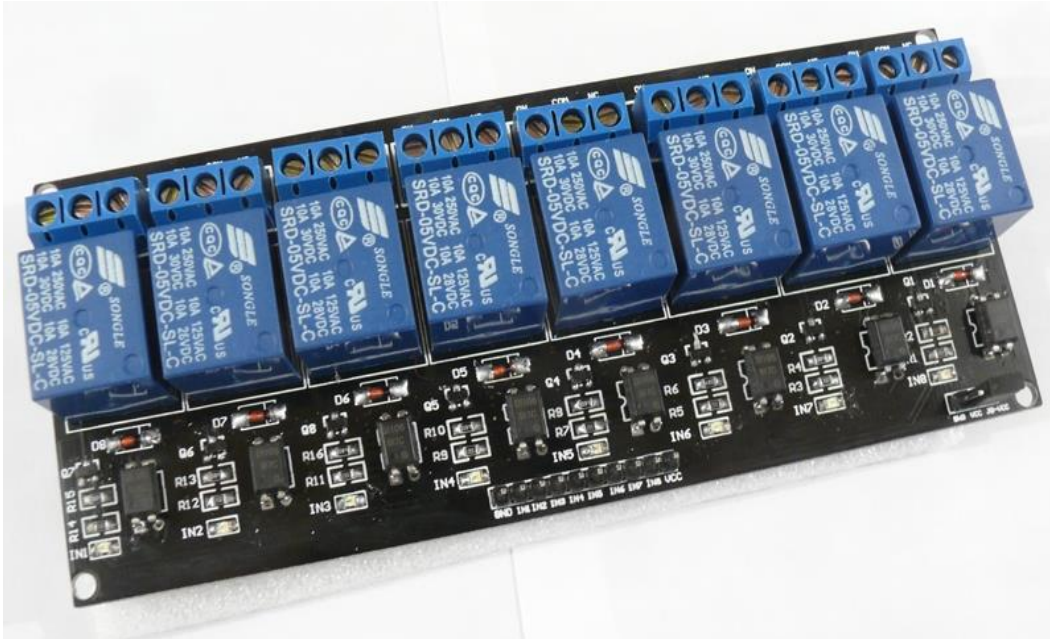
Για την σύνδεση ως χρήστης **pi** χρησιμοποιείται ο προεπιλεγμένος κωδικός πρόσβασης που είναι **raspberrypi**.



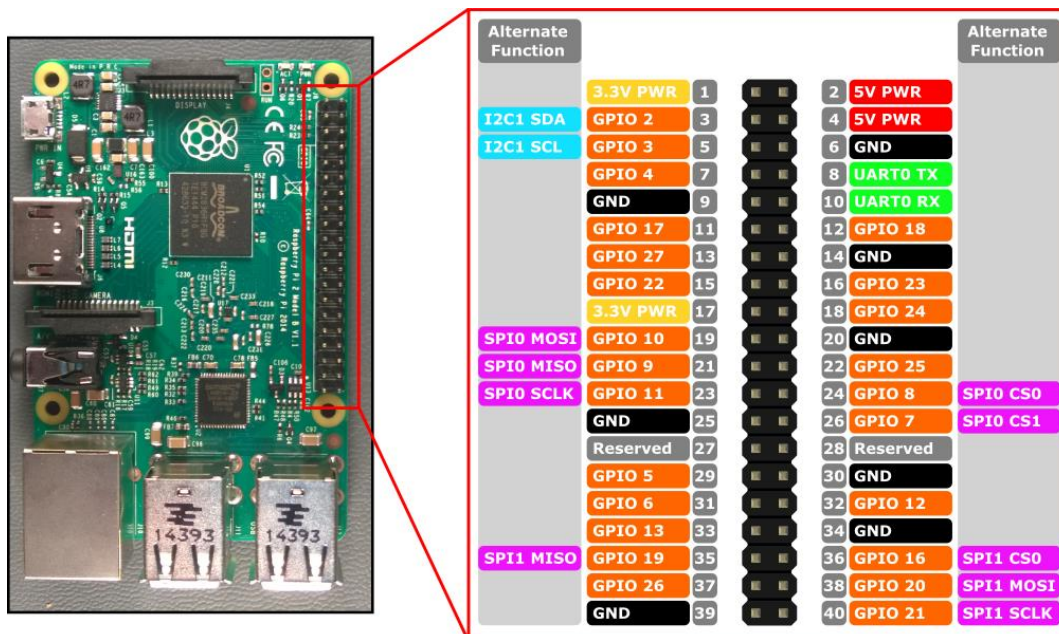
```
pi@raspberrypi: ~  
bill@DESKTOP-1FIK3R0:~$ ssh pi@192.168.2.6  
The authenticity of host '192.168.2.6 (192.168.2.6)' can't be established.  
ECDSA key fingerprint is 35:d1:1a:c1:a8:1a:c9:5c:c3:87:d6:64:a6:68:f3:d7.  
Are you sure you want to continue connecting (yes/no)? yes  
Warning: Permanently added '192.168.2.6' (ECDSA) to the list of known hosts.  
pi@192.168.2.6's password: raspberrypi  
  
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: Thu Mar 23 20:15:34 2017  
  
SSH is enabled and the default password for the 'pi' user has not been changed.  
This is a security risk - please login as the 'pi' user and type 'passwd' to set a new password.  
pi@raspberrypi:~$
```

## 7.2. Επικοινωνία του Ρελέ 8 καναλιών με το Raspberry PI

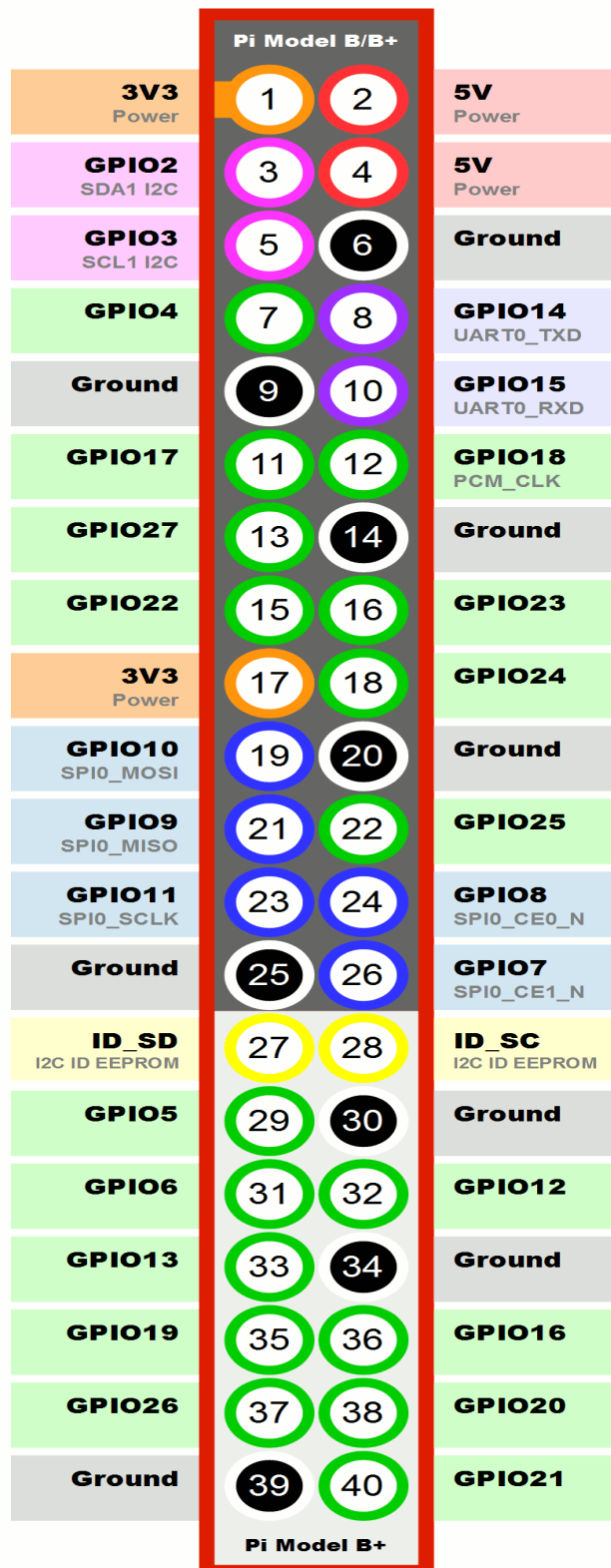
Ο λόγος που αρχικά επιλέχθηκε το ρελέ 8 καναλιών είναι γιατί παρά τις οκταπλές δυνατότητες ελέγχου που προσφέρει κοστίζει μόνο όσο 4 φορές το μονό ρελέ γεγονός που το καθιστά εξαιρετικά φτηνή επιλογή ώστε να ελέγχει ο χρήστης απομακρυσμένα περισσότερες ηλεκτρικές συσκευές. Παρακάτω παρουσιάζονται οι ακροδέκτες που καθιστούν εφικτή τη σύνδεση των ρελέ με το raspberry PI και αναλυτικότερα :



GPIO Pins του Rasberry



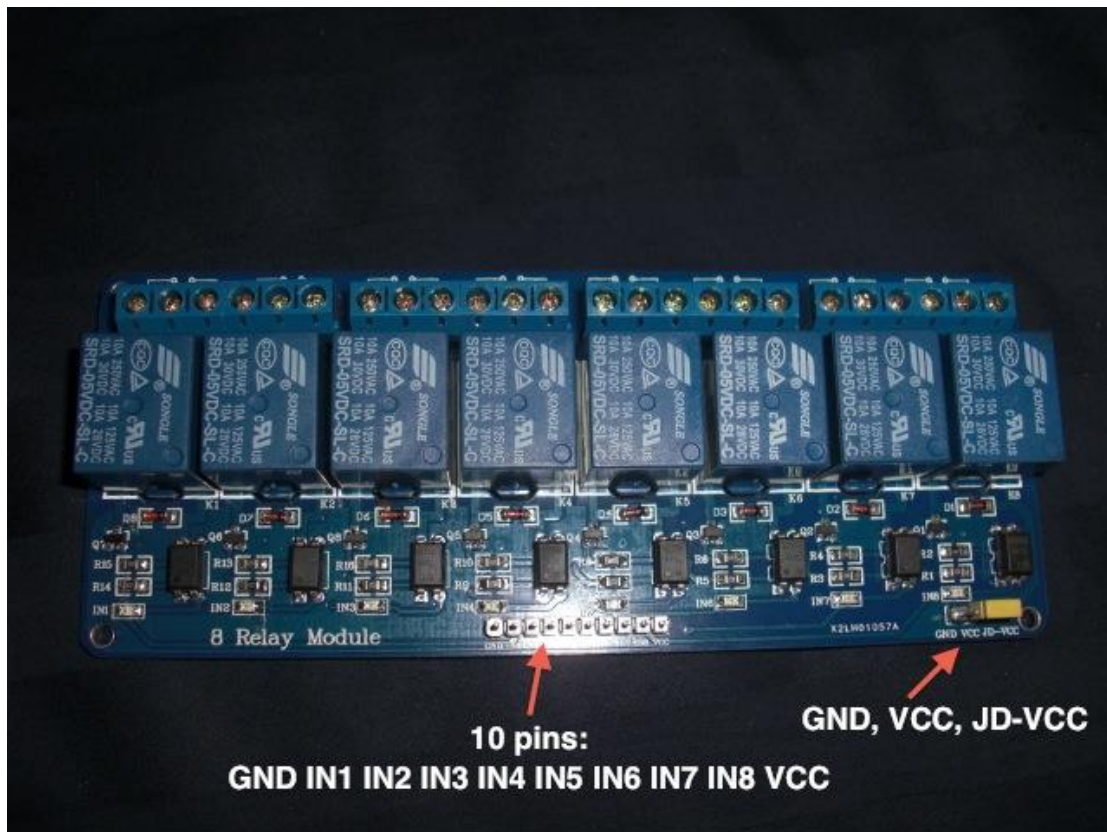
Τα pins του Rasberry φαίνονται αναλυτικότερα στην παρακάτω εικόνα.



[www.raspberrypi-spy.co.uk](http://www.raspberrypi-spy.co.uk)



Οι ακροδέκτες των ρελέ



1<sup>ο</sup> ακροδέκτης (pin) → Γείωση (GND)

2<sup>ος</sup> – 9<sup>ος</sup> ακροδέκτης (pin) → IN1 - IN8 → Όλα τα διαθέσιμα κανάλια / εισόδους για τη σύνδεση οποιασδήποτε συσκευής

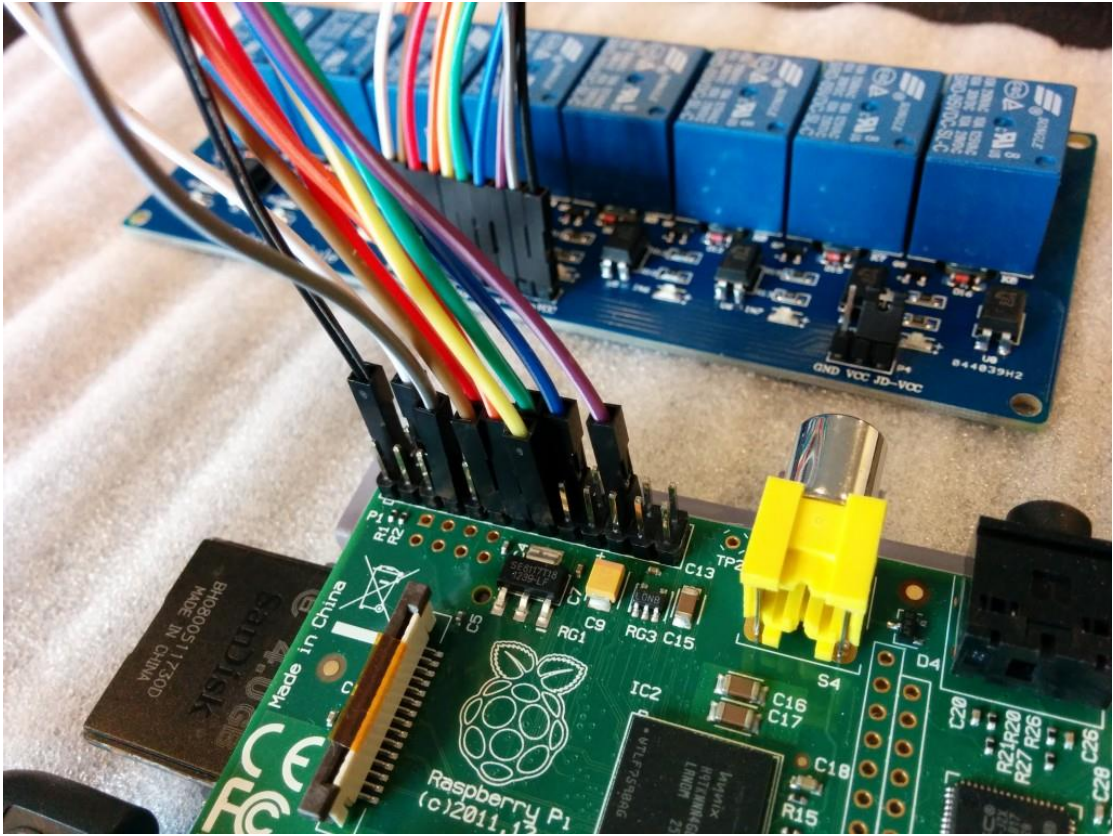
10<sup>ος</sup> ακροδέκτης → Τροφοδοσία ρεύματος (VCC)

Για τη σύνδεση Relay με Raspberry Pi, προαπαιτούμενο είναι καλώδια τύπου Female to Female .

1) Το VCC Pin στο ρελέ με το δεύτερο PIN στο Raspberry Pi (Ισχύς 5 Volt)

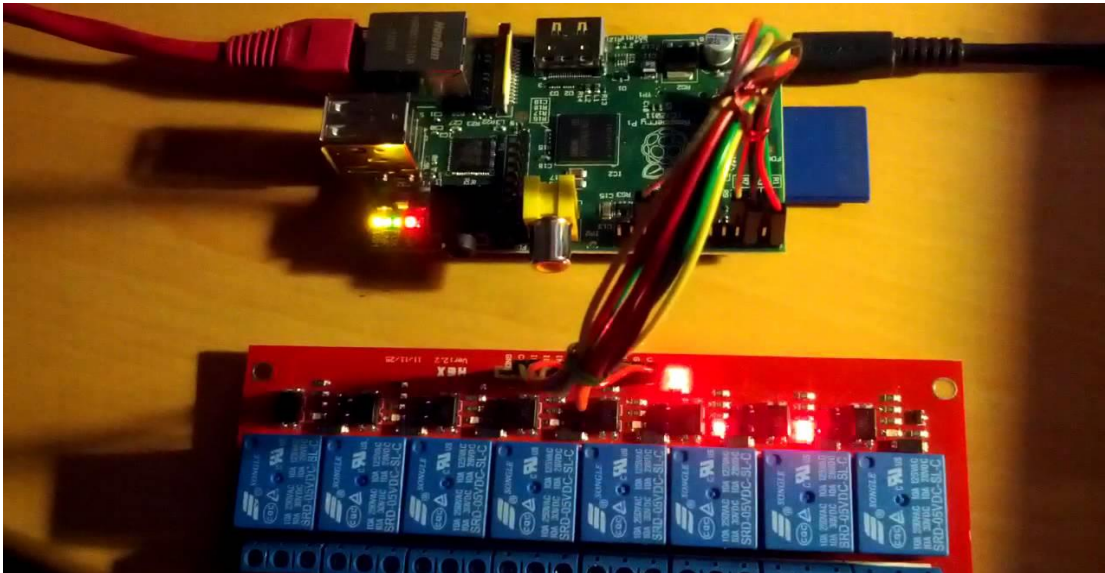
2) Το GND του ρελέ με τον ακροδέκτη γείωσης του Raspberry Pi

3) Όλες οι 8 εισόδους του ρελέ (IN1-IN8) με 8 GPIO Raspberry Pi (το raspberry PI έχει πληθώρα GPIO οπότε ο χρήστης μπορεί να επιλέξει όποια θέλει)



### 7.3. Έλεγχος παροχής ισχύος με python scripts

Δημιουργώντας ένα script με την python, ο χρήστης δύναται να καθορίσει την κατάσταση του,αν περνάει το ρεύμα στην ηλεκτρική συσκευή που είναι με τη σειρά του συνδεδεμένο το ρελέ ή όχι.Στο συγκεκριμένο module με τα ρελέ όταν το ρεύμα ρέει μέσω ενός καναλιού, ανάβει το αντίστοιχο LED φως του συγκεκριμένου καναλιού.



Ως ένα πρώτο πειραματισμό για την εξοικίωση με την python,των βιβλιοθηκών που θα χρησιμοποιηθούν ,δημιουργήθηκε ένα Test script το οποίο επιτρέπει να περνάει το ρεύμα από τα ρελέ , ένα τη φορά , με καθυστέρηση 0.2 δευτερολέπτων. Στην λίστα “Pinlist” συμπεριλήφθηκαν όσα Pins χρησιμοποιήθηκαν για το συγκεκριμένο project.Για την περάτωση του κρίθηκε αναγκαίο η εισαγωγή βιβλιοθηκών για τη διαχείριση των GPIO αλλά και για την δημιουργία των παύσεων των 2 δευτερολέπτων.

**GPIO.LOW** → Το κύκλωμα επιτρέπει τη διέλευση ηλεκτρικής ενέργειας

**GPIO.HIGH** → Το κύκλωμα δεν επιτρέπει τη διέλευση ηλεκτρικής ενέργειας

**Κώδικας Python:**

```
1 #!/usr/bin/python
2 import RPi.GPIO as GPIO
3 import time
4
5 GPIO.setmode(GPIO.BCM)
6
7 # init list with pin numbers
8 pinList = [21, 26, 20, 19, 16, 13, 6, 12]
9
```

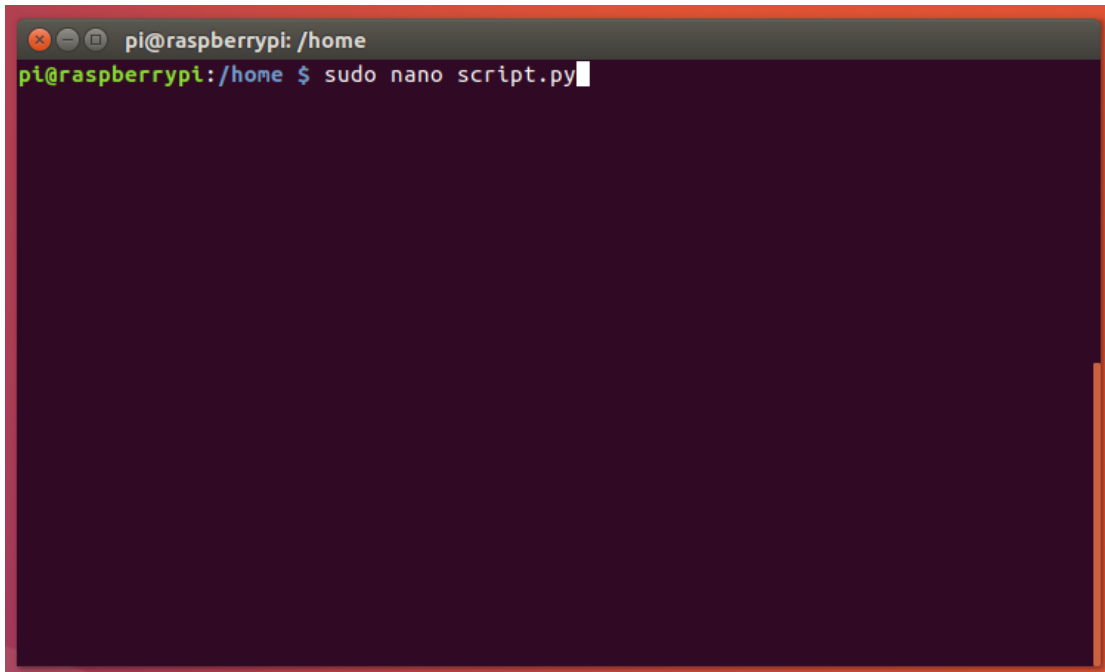
```

10 # through pins and set mode and state to 'low'
11 for i in pinList:
12     GPIO.setup(i, GPIO.OUT)
13     GPIO.output(i, GPIO.HIGH)
14
15 # time to sleep between operations in the main loop
16 SleepTimeL = 0.2
17
18 # main loop
19 try:
20     GPIO.output(21, GPIO.LOW)
21     print "INPUT 1 - GPIO: 21 - LIGHT IS ON"
22     time.sleep(SleepTimeL);
23     GPIO.output(26, GPIO.LOW)
24     print "INPUT 2 - GPIO: 26 - LIGHT IS ON"
25     time.sleep(SleepTimeL);
26     GPIO.output(20, GPIO.LOW)
27     print "INPUT 3 - GPIO: 20 - LIGHT IS ON"
28     time.sleep(SleepTimeL);
29     GPIO.output(19, GPIO.LOW)
30     print "INPUT 4 - GPIO: 19 - LIGHT IS ON"
31     time.sleep(SleepTimeL);
32     GPIO.output(16, GPIO.LOW)
33     print "INPUT 5 - GPIO: 16 - LIGHT IS ON"
34     time.sleep(SleepTimeL);
35     GPIO.output(13, GPIO.LOW)
36     print "INPUT 6 - GPIO: 13 - LIGHT IS ON"
37     time.sleep(SleepTimeL);
38     GPIO.output(6, GPIO.LOW)
39     print "INPUT 7 - GPIO: 6 - LIGHT IS ON"
40     time.sleep(SleepTimeL);
41     GPIO.output(12, GPIO.LOW)
42     print "INPUT 8 - GPIO: 12 - LIGHT IS ON"
43     time.sleep(SleepTimeL);
44     print "Good bye - All LIGHTS ARE OFF"
45
46
47 # End program cleanly with keyboard
48 except KeyboardInterrupt:
49     print "  Quit"
50
51 # Reset GPIO settings
52 GPIO.cleanup()

```

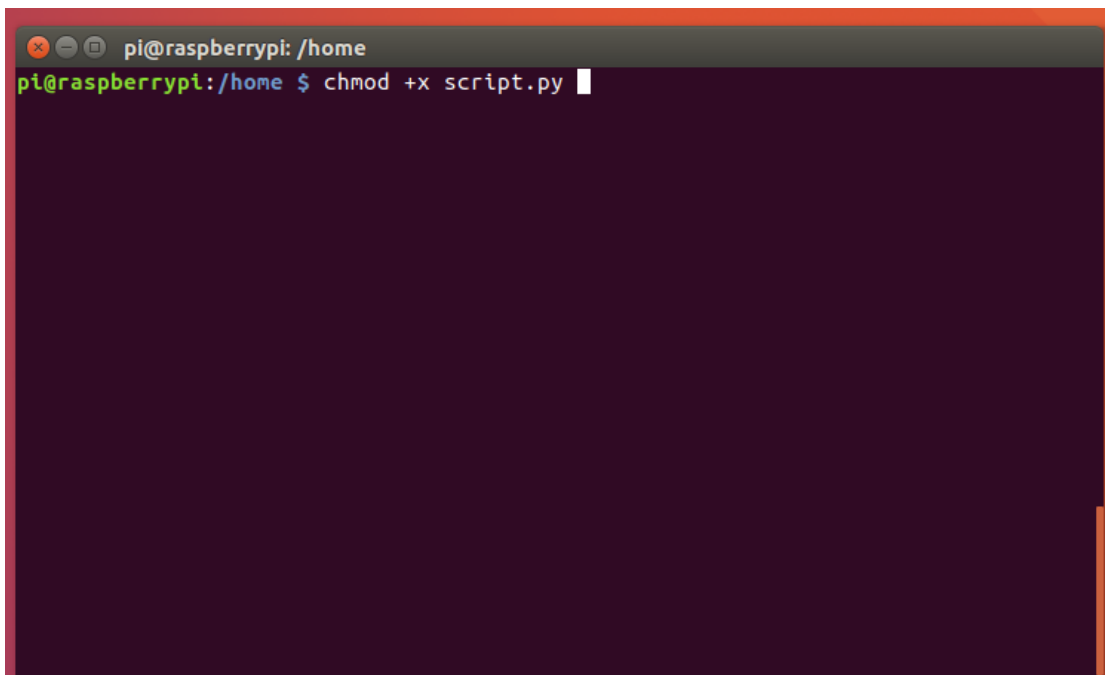


Για την εκτέλεση του κώδικα από το Raspberry, ο συντάκτης εισέρχεται ως χρήστης pi στο raspberry PI και δημιουργεί αρχείο στον αρχικό κατάλογο με επέκταση .py (Π.χ script.py)



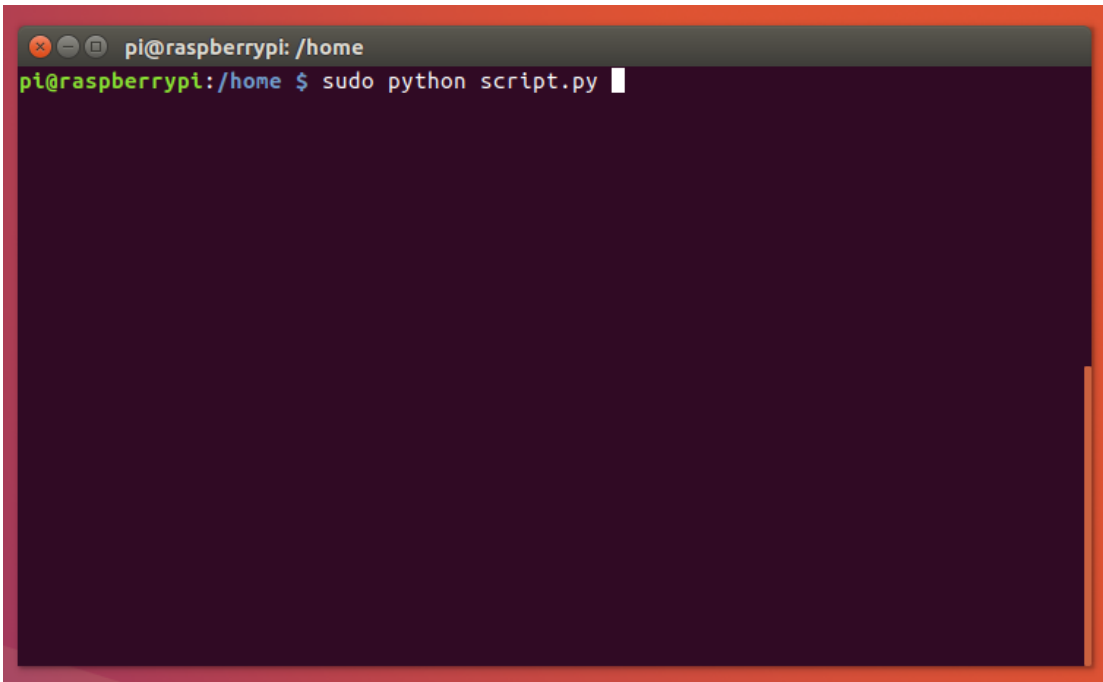
```
pi@raspberrypi: /home
pi@raspberrypi: /home $ sudo nano script.py
```

Το αρχείο μπορεί να μετατραπεί σε εκτελέσιμο με την εντολή :



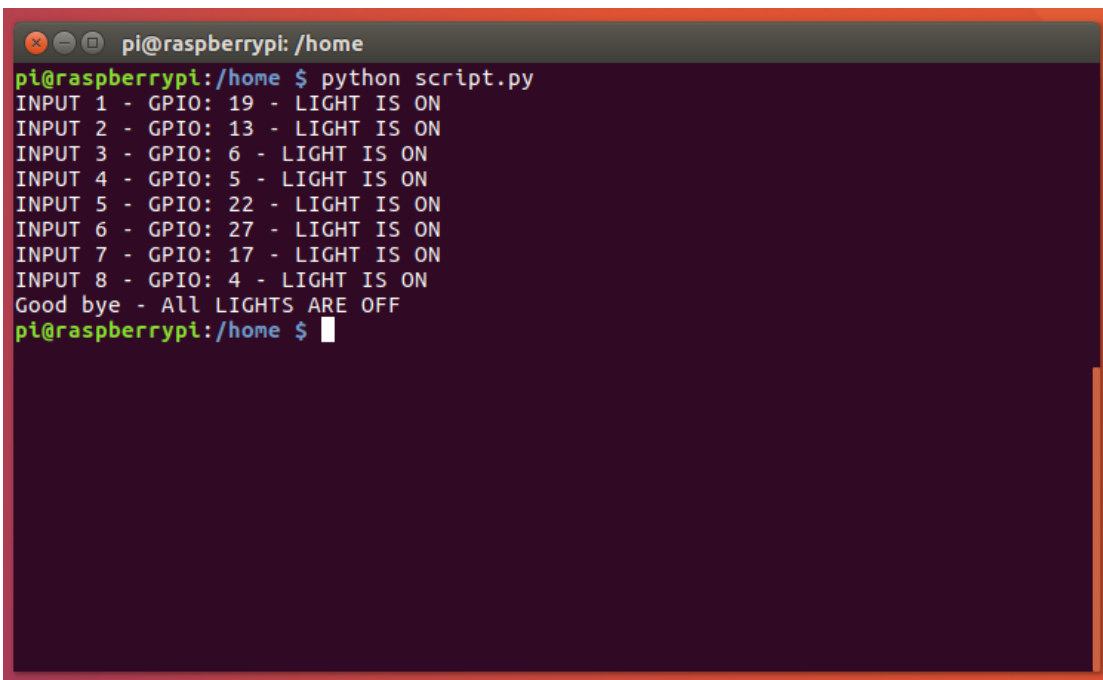
```
pi@raspberrypi: /home
pi@raspberrypi: /home $ chmod +x script.py
```

Για την εκτέλεση του:



```
pi@raspberrypi: /home
pi@raspberrypi: /home $ sudo python script.py
```

Όπως αναμενόταν οι λυχνίες , μια τη φορά, άναβαν με καθυστέρηση 0.2 δευτερολέπτων ενώ μετά το πέρας και της τελευταίας λυχνίας έσβησαν όλες.



```
pi@raspberrypi: /home
pi@raspberrypi: /home $ python script.py
INPUT 1 - GPIO: 19 - LIGHT IS ON
INPUT 2 - GPIO: 13 - LIGHT IS ON
INPUT 3 - GPIO: 6 - LIGHT IS ON
INPUT 4 - GPIO: 5 - LIGHT IS ON
INPUT 5 - GPIO: 22 - LIGHT IS ON
INPUT 6 - GPIO: 27 - LIGHT IS ON
INPUT 7 - GPIO: 17 - LIGHT IS ON
INPUT 8 - GPIO: 4 - LIGHT IS ON
Good bye - All LIGHTS ARE OFF
pi@raspberrypi: /home $
```

## 8. Απομακρυσμένος έλεγχος μέσω ιστού

Η επικοινωνία και η εντολοδότηση του Rasberry με τον ιστό θα επιτευχθεί με php και η ανάπτυξη διεπαφής ιστου με HTML5,CSS,JAVASCRIPT.

### 8.1. Εγκατάσταση απαραίτητων εργαλείων

Για την εκτέλεση της εφαρμογής προαπαιτούμενο εργαλείο είναι ο Apache Web Server και η PHP.Η εγκατάσταση τους γίνεται με την πλήκτρολόγηση των παρακάτω εντολών στο τερματικό.

Για το apache web server:

```
pi@raspberrypi: ~  
pi@raspberrypi:~ $ sudo apt-get install apache2
```

Για το PHP:

```
pi@raspberrypi: ~  
pi@raspberrypi:~ $ sudo apt-get install php5
```

Για την δημιουργία νέου φακέλου(php-controller ο οποίος θα περιέχει ολόκληρη την PHP εφαρμογή) ο συντάκτης κατευθύνεται / var / www / html /

```
pi@raspberrypi: /var/www/html  
pi@raspberrypi:/home $ cd /var/www/html/  
pi@raspberrypi:/var/www/html $ sudo mkdir php-controller  
pi@raspberrypi:/var/www/html $ ls -al  
total 12  
drwxr-xr-x 3 root root 4096 Mar 23 20:57 .  
drwxr-xr-x 3 root root 4096 Mar 23 17:49 ..  
drwxr-xr-x 2 root root 4096 Mar 23 20:57 php-controller  
pi@raspberrypi:/var/www/html $
```

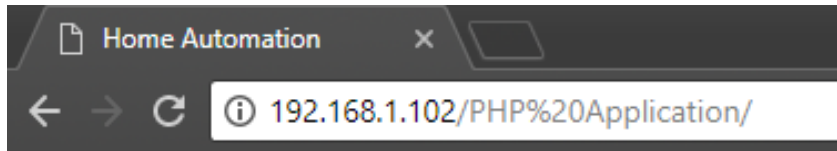
Για ταχύτερη εύρεση των αρχείων κρίθηκε αναγκαία από τον συντάκτη, η δημιουργία μερικών υποφακέλων για την καλύτερη οργάνωση των αρχείων μας. Από το παρακάτω screenshot με την λίστα των φακέλων εμφανίζονται οι εξής:

- Ένας φάκελος με όνομα **css** στο οποίο τοποθετήθηκαν όλα τα **css αρχεία** που θα χρησιμοποιηθούν για το **styling** στην ιστοσελίδα.
- Ένας φάκελος με όνομα **js** που περιέχει όλα τα **αρχεία javascript**, συμπεριλαμβανομένης και της βιβλιοθήκης JQuery .
- Ένας φάκελος που ονομάζεται **php** όπου συγκεντρώθηκαν όλα τα **αρχεία PHP**.
- Ένας φάκελος που ονομάζεται **scripts** όπου αποθηκεύτηκαν όλα τα Python scripts που εκτελούνται όταν πατηθούν συγκεκριμένα κουμπιά της εφαρμογής .
- Το αρχείο **index.php** που είναι η πρώτη σελίδα που εμφανίζεται κατά την επίσκεψή στην εφαρμογή Ιστού .

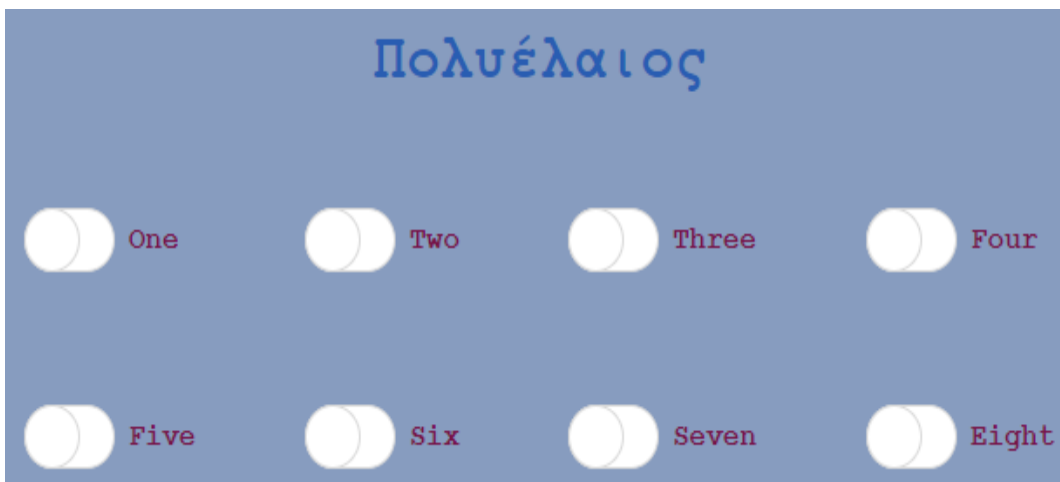
```
pi@raspberrypi: /var/www/html/php-controller
pi@raspberrypi:/var/www/html/php-controller $ ls -al
total 28
drwxr-xr-x 6 pi www-data 4096 Mar 23 22:45 .
drwxr-xr-x 3 pi www-data 4096 Mar 23 21:04 ..
drwxr-xr-x 2 pi www-data 4096 Jun 21 2016 css
-rw-r--r-- 1 pi www-data 1016 Mar 23 22:29 index.php
drwxr-xr-x 2 pi www-data 4096 Apr 27 2016 js
drwxr-xr-x 2 pi www-data 4096 Mar 23 22:01 php
drwxr-xr-x 2 pi www-data 4096 Mar 23 22:13 scripts
pi@raspberrypi:/var/www/html/php-controller $
```

Για την επίσκεψη της ιστοσελίδας, Ο χρήστης ανοίγει ένα πρόγραμμα περιήγησης και πληκτρολογεί τη **διεύθυνση IP** του Raspberry Pi (στη δική μας περίπτωση: 192.168.1.102), στη συνέχεια το σύμβολο της καθέτου και τέλος το όνομα του φακέλου στον οποίο τοποθετήθηκαν όλα τα αρχεία , ο φάκελος πρέπει να είναι μέσα στο / var / www / html και στην περίπτωσή μας, ονομάζεται: **PHP Application**)

Επομένως, η τελική μορφή του συνδέσμου πρέπει να είναι η εξής:



Η σελίδα προορισμού θα πρέπει να είναι παρόμοια με την ακόλουθη:



Όπως υποδεικνύει η παραπάνω εικόνα, υπάρχουν οχτώ ξεχωριστοί διακόπτες.

#### **index.php:**

```
1 <html>
2   <head>
3     <title> Home Automation</title>
4
5     <!-- we include all the appropriate javascript and css files -
->
6     <link rel="stylesheet" type="text/css" href= "css/Switch.css">
7     <script type="text/javascript" src="js/jquery-1.12.3.min.js">
</script>
8     <script type="text/javascript" src="js/functions.js">
</script>
9   </head>
10
11  <body style="background-color:#879cbf">
12    <h1 align="center"><font face="Courier" size="6"
color="295db2"> Πολυέλαιος </h1>
13
```

```

14     <table align="center" cellpadding="30">
15         <tr>
16             <td>
17                 <input type='checkbox' class='ios8-switch ios8-switch-lg'
18                 id='checkbox-1'>
19                 <label for='checkbox-1'><font
20                 color="761549"><b>One</b></font></label>
21             </td>
22             <td>
23                 <input type='checkbox' class='ios8-switch ios8-switch-lg'
24                 id='checkbox-2'>
25                 <label for='checkbox-2'><font
26                 color="761549"><b>Two</b></font></label>
27             </td>
28             <td>
29                 <input type='checkbox' class='ios8-switch ios8-switch-lg'
30                 id='checkbox-3'>
31                 <label for='checkbox-3'><font
32                 color="761549"><b>Three</font></label>
33             </td>
34             <td>
35                 <input type='checkbox' class='ios8-switch ios8-switch-lg'
36                 id='checkbox-4'>
37                 <label for='checkbox-4'><font
38                 color="761549"><b>Four</font></label>
39             </td>
40             </tr>
41             <tr>
42                 <td>
43                     <input type='checkbox' class='ios8-switch ios8-switch-lg'
44                     id='checkbox-5'>
45                     <label for='checkbox-5'><font
46                     color="761549"><b>Five</font></label>
47                 </td>
48                 <td>
49                     <input type='checkbox' class='ios8-switch ios8-switch-lg'
50                     id='checkbox-6'>
51                     <label for='checkbox-6'><font
52                     color="761549"><b>Six</font></label>
53                 </td>
54                 <td>
55                     <input type='checkbox' class='ios8-switch ios8-switch-lg'
56                     id='checkbox-7'>
57                     <label for='checkbox-7'><font
58                     color="761549"><b>Seven</font></label>
59                 </td>
60             </tr>
61         </table>

```

```

47         <input type='checkbox' class='ios8-switch ios8-switch-lg'
id='checkbox-8'>
48         <label for='checkbox-8'><font
color="761549"><b>Eight</font></label>
49         </td>
50     </tr>
51 </table>
52 </body>
53 </html>

```

Η JQuery είναι απαραίτητη ώστε ο χρήστης να καλέσει τις αντίστοιχες σελίδες php μέσω του AJAX. Αυτή η λειτουργία υλοποιείται μέσα στο αρχείο **functions.js**.

### Functions.js:

```

1 $(document).ready(function()
2 {
3     $("#checkbox-1").click(function()
4     {
5         if($('#checkbox-1').is(':checked') )
6         {
7             $.ajax({url: "php/one-checked.php "});
8         }
9         else
10        {
11            $.ajax({url: "php/one-unchecked.php"});
12        }
13    });
14
15    $("#checkbox-2").click(function()
16    {
17        if($('#checkbox-2').is(':checked') )
18        {
19            $.ajax({url: "php/two-checked.php"});
20        }
21        else
22        {
23            $.ajax({url: "php/two-unchecked.php"});
24        }
25    });
26
27    $("#checkbox-3").click(function()
28    {
29        if($('#checkbox-3').is(':checked') )
30        {
31            $.ajax({url: "php/three-checked.php"});

```

```

32     }
33     else
34     {
35         $.ajax({url: "php/three-unchecked.php"});
36     }
37 });
38
39 $( "#checkbox-4" ).click(function()
40 {
41     if( $('#checkbox-4').is(':checked') )
42     {
43         $.ajax({url: "php/four-checked.php"});
44     }
45     else
46     {
47         $.ajax({url: "php/four-unchecked.php"});
48     }
49 });
50
51 $( "#checkbox-5" ).click(function()
52 {
53     if( $('#checkbox-5').is(':checked') )
54     {
55         $.ajax({url: "php/five-checked.php"});
56     }
57     else
58     {
59         $.ajax({url: "php/five-unchecked.php"});
60     }
61 });
62
63 $( "#checkbox-6" ).click(function()
64 {
65     if( $('#checkbox-6').is(':checked') )
66     {
67         $.ajax({url: "php/six-checked.php"});
68     }
69     else
70     {
71         $.ajax({url: "php/six-unchecked.php"});
72     }
73 });
74
75 $( "#checkbox-7" ).click(function()
76 {
77     if( $('#checkbox-7').is(':checked') )
78     {

```



```

79     $.ajax({url: "php/seven-checked.php"});
80   }
81   else
82   {
83     $.ajax({url: "php/seven-unchecked.php"});
84   }
85 });
86
87 $( "#checkbox-8" ).click(function()
88 {
89   if( $('#checkbox-8').is(':checked') )
90   {
91     $.ajax({url: "php/eight-checked.php"});
92   }
93   else
94   {
95     $.ajax({url: "php/eight-unchecked.php"});
96   }
97 });
98 });

```

Προχωρώντας, το γραφικό μέρος της εφαρμογής καλύπτεται από τα αρχεία με επέκταση `css` που περιλαμβάνονται στο φάκελο `"css"`. Το αρχείο **Switch.css** του CSS περιέχει τον κώδικα για τη διακόσμηση των διακοπών που διαχειρίζονται τα κανάλια του ρελέ.

#### Switch.cs:

```

1 input[type="checkbox"].ios8-switch {
2   position: absolute;
3   margin: 8px 0 0 16px;
4 }
5 input[type="checkbox"].ios8-switch + label {
6   position: relative;
7   padding: 5px 0 0 50px;
8   line-height: 2.0em;
9 }
10 input[type="checkbox"].ios8-switch + label:before {
11   content: "";
12   position: absolute;
13   display: block;
14   left: 0;
15   top: 0;
16   width: 40px; /* x*5 */
17   height: 24px; /* x*3 */
18   border-radius: 16px; /* x*2 */
19   background: #fff;

```

```

20     border: 1px solid #d9d9d9;
21     -webkit-transition: all 0.3s;
22     transition: all 0.3s;
23 }
24 input[type="checkbox"].ios8-switch + label:after {
25     content: "";
26     position: absolute;
27     display: block;
28     left: 0px;
29     top: 0px;
30     width: 24px; /* x*3 */
31     height: 24px; /* x*3 */
32     border-radius: 16px; /* x*2 */
33     background: #fff;
34     border: 1px solid #d9d9d9;
35     -webkit-transition: all 0.3s;
36     transition: all 0.3s;
37 }
38 input[type="checkbox"].ios8-switch + label:hover:after {
39     box-shadow: 0 0 5px rgba(0,0,0,0.3);
40 }
41 input[type="checkbox"].ios8-switch:checked + label:after {
42     margin-left: 16px;
43 }
44 input[type="checkbox"].ios8-switch:checked + label:before {
45     background: #55D069;
46 }
47
48 /* LARGE */
49
50 input[type="checkbox"].ios8-switch-lg {
51     margin: 10px 0 0 20px;
52 }
53 input[type="checkbox"].ios8-switch-lg + label {
54     position: relative;
55     padding: 7px 0 0 60px;
56     line-height: 2.3em;
57 }
58 input[type="checkbox"].ios8-switch-lg + label:before {
59     width: 50px; /* x*5 */
60     height: 30px; /* x*3 */
61     border-radius: 20px; /* x*2 */
62 }
63 input[type="checkbox"].ios8-switch-lg + label:after {
64     width: 30px; /* x*3 */
65     height: 30px; /* x*3 */
66     border-radius: 20px; /* x*2 */

```

```

67 }
68 input[type="checkbox"].ios8-switch-lg + label:hover:after {
69     box-shadow: 0 0 8px rgba(0,0,0,0.3);
70 }
71 input[type="checkbox"].ios8-switch-lg:checked + label:after {
72     margin-left: 20px; /* x*2 */
73 }

```

Επιπλέον, κάθε φορά που ο χρήστης κάνει κλικ σε ένα κουμπί, η αντίστοιχη σελίδα PHP καλείται μέσω AJAX. Όλες αυτές οι PHP σελίδες έχουν δύο βασικούς ρόλους. Ο πρώτος είναι να ενεργοποιήσουν το σωστό python script (χρησιμοποιώντας την εντολή php `shell_exec()`) και το δεύτερο να εκτυπώσουν (`echo()`) το κατάλληλο μήνυμα στην οθόνη. Έτσι, η δομή τους είναι σχεδόν παρόμοια, αλλά με μικρές διαφοροποιήσεις στο κείμενο κάθε φορά.

**one-checked.php** → User turns ON the light switch 1

```

1 <?php
2 $output = shell_exec("sudo python ../scripts/switch-one-on.py");
3 echo "<pre>$output</pre>";
4 ?>

```

**one-unchecked.php** → User turns OFF the light switch 1

```

1 <?php
2 $output = shell_exec("sudo python ../scripts/switch-one-off.py");
3 echo "<pre>$output</pre>";
4 ?>

```

**two-checked.php** → User turns ON the light switch 2

```

1 <?php
2 $output = shell_exec("sudo python ../scripts/switch-two-on.py");
3 echo "<pre>$output</pre>";
4 ?>

```

**two-unchecked.php** → User turns OFF the light switch 2

```

1 <?php
2 $output = shell_exec("sudo python ../scripts/switch-two-off.py");
3 echo "<pre>$output</pre>";
4 ?>

```

**tree-checked.php** → User turns ON the light switch 3

```
1 <?php
2 $output = shell_exec("sudo python ../scripts/switch-three-on.py");
3 echo "<pre>$output</pre>";
4 ?>
```

**tree-unchecked.php** → User turns OFF the light switch 3

```
1 <?php
2 $output = shell_exec("sudo python ../scripts/switch-three-off.py");
3 echo "<pre>$output</pre>";
4 ?>
```

**four-checked.php** → User turns ON the light switch 4

```
1 <?php
2 $output = shell_exec("sudo python ../scripts/switch-four-on.py");
3 echo "<pre>$output</pre>";
4 ?>
```

**four-unchecked.php** → User turns OFF the light switch 4

```
1 <?php
2 $output = shell_exec("sudo python ../scripts/switch-four-off.py");
3 echo "<pre>$output</pre>";
4 ?>
```

**five-checked.php** → User turns ON the light switch 5

```
1 <?php
2 $output = shell_exec("sudo python ../scripts/switch-five-on.py");
3 echo "<pre>$output</pre>";
4 ?>
```

**five-unchecked.php** → User turns OFF the light switch 5

```
1 <?php
2 $output = shell_exec("sudo python ../scripts/switch-five-off.py");
3 echo "<pre>$output</pre>";
4 ?>
```

**six-checked.php** → User turns ON the light switch 6

```
1 <?php
2 $output = shell_exec("sudo python ../scripts/switch-six-on.py");
3 echo "<pre>$output</pre>";
4 ?>
```

**six-unchecked.php** → User turns OFF the light switch 6

```
1 <?php
2 $output = shell_exec("sudo python ../scripts/switch-six-off.py");
3 echo "<pre>$output</pre>";
4 ?>
```

**seven-checked.php** → User turns ON the light switch 7

```
1 <?php
2 $output = shell_exec("sudo python ../scripts/switch-seven-on.py");
3 echo "<pre>$output</pre>";
4 ?>
```

**seven-unchecked.php** → User turns OFF the light switch 7

```
1 <?php
2 $output = shell_exec("sudo python ../scripts/switch-seven-off.py");
3 echo "<pre>$output</pre>";
4 ?>
```

**eight-checked.php** → User turns ON the light switch 8

```
1 <?php
2 $output = shell_exec("sudo python ../scripts/switch-eight-on.py");
3 echo "<pre>$output</pre>";
4 ?>
```

**eight-unchecked.php** → User turns OFF the light switch 8

```
1 <?php
2 $output = shell_exec("sudo python ../scripts/switch-eight-off.py");
3 echo "<pre>$output</pre>";
4 ?>
```

## 8.2. Python Scripts

**switch-one-on.py** → The First Relay Input Led Lamp - GPIO 12 turns ON

**switch-one-off.py** → The First Relay Input Led Lamp - GPIO 12 turns OFF

**switch-two-on.py** → The First Relay Input Led Lamp - GPIO 6 turns ON

**switch-two-off.py** → The First Relay Input Led Lamp - GPIO 6 turns OFF

**switch-three-on.py** → The First Relay Input Led Lamp - GPIO 13 turns ON

**switch-three-off.py** → The First Relay Input Led Lamp - GPIO 13 turns OFF

**switch-four-on.py** → The First Relay Input Led Lamp - GPIO 16 turns ON

**switch-four-off.py** → The First Relay Input Led Lamp - GPIO 16 turns OFF

**switch-five-on.py** → The First Relay Input Led Lamp - GPIO 19 turns ON

**switch-five-off.py** → The First Relay Input Led Lamp - GPIO 19 turns OFF

**switch-six-on.py** → The First Relay Input Led Lamp - GPIO 20 turns ON

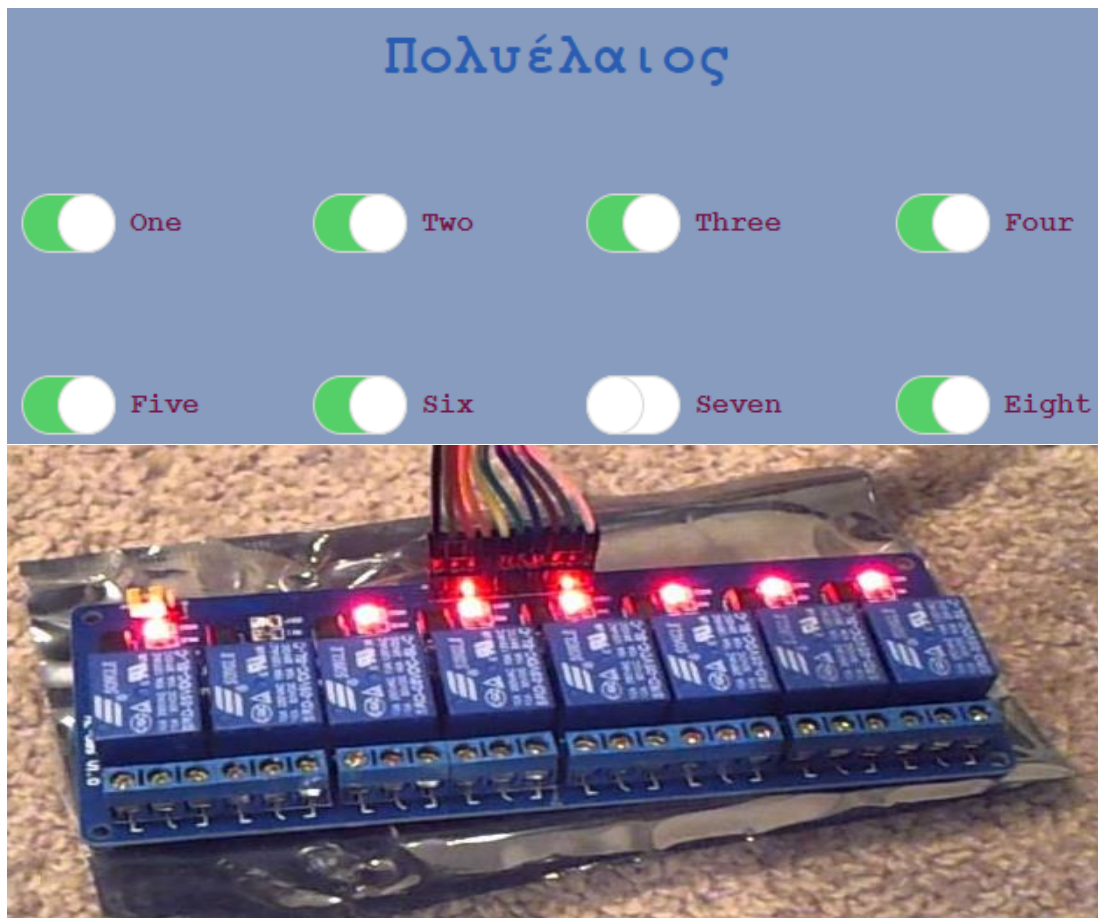
**switch-six-off.py** → The First Relay Input Led Lamp - GPIO 20 turns OFF

**switch-seven-on.py** → The First Relay Input Led Lamp - GPIO 26 turns ON

**switch-seven-off.py** → The First Relay Input Led Lamp - GPIO 26 turns OFF

**switch-eight-on.py** → The First Relay Input Led Lamp - GPIO 21 turns ON

**switch-eight-off.py** → The First Relay Input Led Lamp - GPIO 21 turns OFF



### switch-one-on.py

```
1 #!/usr/bin/python
2 import RPi.GPIO as GPIO
3 import time
4
5 GPIO.setmode(GPIO.BCM)
6 # init list with pin numbers
7
8 GPIO.setup(12, GPIO.OUT)
9 GPIO.output(12, GPIO.HIGH)
10 try:
11     GPIO.output(12, GPIO.LOW)
12     # print "Relay Input 1 - Gpio: 12 - Led is on"
13
14 # End program cleanly with keyboard
15 except KeyboardInterrupt:
16     print "  Quit"
17
18 # Reset GPIO settings
19 GPIO.cleanup()
```

### switch-one-off.py

```
1 #!/usr/bin/python
2 import RPi.GPIO as GPIO
3 import time
4
5 GPIO.setmode(GPIO.BCM)
6 # init list with pin numbers
7
8 GPIO.setup(12, GPIO.OUT)
9 GPIO.output(12, GPIO.HIGH)
10
11 # main loop
12 try:
13     GPIO.output(12, GPIO.HIGH)
14     # print "Relay Input 1 - Gpio: 12 - Led is off"
15
16 # End program cleanly with keyboard
17 except KeyboardInterrupt:
18     print "  Quit"
19
20 # Reset GPIO settings
21 GPIO.cleanup()
```

## switch-two-on.py

```
1 #!/usr/bin/python
2 import RPi.GPIO as GPIO
3 import time
4
5 GPIO.setmode(GPIO.BCM)
6 # init list with pin numbers
7
8 GPIO.setup(6, GPIO.OUT)
9 GPIO.output(6, GPIO.HIGH)
10
11 try:
12     GPIO.output(6, GPIO.LOW)
13     # print "Relay Input 1 - Gpio: 6 - Led is on"
14
15 # End program cleanly with keyboard
16 except KeyboardInterrupt:
17     print "  Quit"
18
19 # Reset GPIO settings
20 GPIO.cleanup()
```

## switch-two-off.py

```
1 #!/usr/bin/python
2 import RPi.GPIO as GPIO
3 import time
4
5 GPIO.setmode(GPIO.BCM)
6 # init list with pin numbers
7
8 GPIO.setup(6, GPIO.OUT)
9 GPIO.output(6, GPIO.HIGH)
10
11 try:
12     GPIO.output(6, GPIO.HIGH)
13     # print "Relay Input 1 - Gpio: 6 - Led is off"
14
15 # End program cleanly with keyboard
16 except KeyboardInterrupt:
17     print "  Quit"
18
19 # Reset GPIO settings
20 GPIO.cleanup()
```



### switch-tree-on.py

```
1 #!/usr/bin/python
2 import RPi.GPIO as GPIO
3 import time
4
5 GPIO.setmode(GPIO.BCM)
6 # init list with pin numbers
7
8 GPIO.setup(13, GPIO.OUT)
9 GPIO.output(13, GPIO.HIGH)
10
11 try:
12     GPIO.output(13, GPIO.LOW)
13     # print "Relay Input 1 - Gpio: 13 - Led is on"
14
15 # End program cleanly with keyboard
16 except KeyboardInterrupt:
17     print "  Quit"
18
19 # Reset GPIO settings
20 GPIO.cleanup()
```

### switch- tree -off.py

```
1 #!/usr/bin/python
2 import RPi.GPIO as GPIO
3 import time
4
5 GPIO.setmode(GPIO.BCM)
6 # init list with pin numbers
7
8 GPIO.setup(13, GPIO.OUT)
9 GPIO.output(13, GPIO.HIGH)
10
11 try:
12     GPIO.output(13, GPIO.HIGH)
13     # print "Relay Input 1 - Gpio: 13 - Led is off"
14
15 # End program cleanly with keyboard
16 except KeyboardInterrupt:
17     print "  Quit"
18
19 # Reset GPIO settings
20 GPIO.cleanup()
```

### switch-four-on.py

```
1 #!/usr/bin/python
2 import RPi.GPIO as GPIO
3 import time
4
5 GPIO.setmode(GPIO.BCM)
6 # init list with pin numbers
7
8 GPIO.setup(16, GPIO.OUT)
9 GPIO.output(16, GPIO.HIGH)
10
11 try:
12     GPIO.output(16, GPIO.LOW)
13     # print "Relay Input 1 - Gpio: 16 - Led is on"
14
15 # End program cleanly with keyboard
16 except KeyboardInterrupt:
17     print "  Quit"
18
19 # Reset GPIO settings
20 GPIO.cleanup()
```

### switch- four -off.py

```
1 #!/usr/bin/python
2 import RPi.GPIO as GPIO
3 import time
4
5 GPIO.setmode(GPIO.BCM)
6 # init list with pin numbers
7
8 GPIO.setup(16, GPIO.OUT)
9 GPIO.output(16, GPIO.HIGH)
10
11 try:
12     GPIO.output(16, GPIO.HIGH)
13     # print "Relay Input 1 - Gpio: 16 - Led is off"
14
15 # End program cleanly with keyboard
16 except KeyboardInterrupt:
17     print "  Quit"
18
19 # Reset GPIO settings
20 GPIO.cleanup()
```

### switch-five-on.py

```
1 #!/usr/bin/python
2 import RPi.GPIO as GPIO
3 import time
4
5 GPIO.setmode(GPIO.BCM)
6 # init list with pin numbers
7
8 GPIO.setup(19, GPIO.OUT)
9 GPIO.output(19, GPIO.HIGH)
10
11 try:
12     GPIO.output(19, GPIO.LOW)
13     # print "Relay Input 1 - Gpio: 19 - Led is on"
14
15 # End program cleanly with keyboard
16 except KeyboardInterrupt:
17     print "  Quit"
18
19 # Reset GPIO settings
20 GPIO.cleanup()
```

### switch-five-off.py

```
1 #!/usr/bin/python
2 import RPi.GPIO as GPIO
3 import time
4
5 GPIO.setmode(GPIO.BCM)
6 # init list with pin numbers
7 GPIO.setup(19, GPIO.OUT)
8 GPIO.output(19, GPIO.HIGH)
9
10 try:
11     GPIO.output(19, GPIO.HIGH)
12     # print "Relay Input 1 - Gpio: 19 - Led is off"
13
14 # End program cleanly with keyboard
15 except KeyboardInterrupt:
16     print "  Quit"
17
18 # Reset GPIO settings
19 GPIO.cleanup()
```

### switch-six-on.py

```
1 #!/usr/bin/python
2 import RPi.GPIO as GPIO
3 import time
4
5 GPIO.setmode(GPIO.BCM)
6 # init list with pin numbers
7
8 GPIO.setup(20, GPIO.OUT)
9 GPIO.output(20, GPIO.HIGH)
10
11 try:
12     GPIO.output(20, GPIO.LOW)
13     # print "Relay Input 1 - Gpio: 20 - Led is on"
14
15 # End program cleanly with keyboard
16 except KeyboardInterrupt:
17     print "  Quit"
18
19 # Reset GPIO settings
20 GPIO.cleanup()
```

### switch-six-off.py

```
1 #!/usr/bin/python
2 import RPi.GPIO as GPIO
3 import time
4
5 GPIO.setmode(GPIO.BCM)
6 # init list with pin numbers
7
8 GPIO.setup(20, GPIO.OUT)
9 GPIO.output(20, GPIO.HIGH)
10
11 try:
12     GPIO.output(20, GPIO.HIGH)
13     # print "Relay Input 1 - Gpio: 20 - Led is off"
14
15 # End program cleanly with keyboard
16 except KeyboardInterrupt:
17     print "  Quit"
18
19 # Reset GPIO settings
20 GPIO.cleanup()
```

### switch-seven-on.py

```
1 #!/usr/bin/python
2 import RPi.GPIO as GPIO
3 import time
4
5 GPIO.setmode(GPIO.BCM)
6 # init list with pin numbers
7
8 GPIO.setup(26, GPIO.OUT)
9 GPIO.output(26, GPIO.HIGH)
10
11 try:
12     GPIO.output(26, GPIO.LOW)
13     # print "Relay Input 1 - Gpio: 26 - Led is on"
14
15 # End program cleanly with keyboard
16 except KeyboardInterrupt:
17     print "  Quit"
18
19 # Reset GPIO settings
20 GPIO.cleanup()
```

### switch-seven-off.py

```
1 #!/usr/bin/python
2 import RPi.GPIO as GPIO
3 import time
4
5 GPIO.setmode(GPIO.BCM)
6 # init list with pin numbers
7
8 GPIO.setup(26, GPIO.OUT)
9 GPIO.output(26, GPIO.HIGH)
10
11 try:
12     GPIO.output(26, GPIO.HIGH)
13     # print "Relay Input 1 - Gpio: 26 - Led is off"
14
15 # End program cleanly with keyboard
16 except KeyboardInterrupt:
17     print "  Quit"
18
19 # Reset GPIO settings
20 GPIO.cleanup()
```

### switch-eight-on.py

```
1 #!/usr/bin/python
2 import RPi.GPIO as GPIO
3 import time
4
5 GPIO.setmode(GPIO.BCM)
6 # init list with pin numbers
7
8 GPIO.setup(21, GPIO.OUT)
9 GPIO.output(21, GPIO.HIGH)
10
11 try:
12     GPIO.output(21, GPIO.LOW)
13     # print "Relay Input 1 - Gpio: 21 - Led is on"
14
15 # End program cleanly with keyboard
16 except KeyboardInterrupt:
17     print "  Quit"
18
19 # Reset GPIO settings
20 GPIO.cleanup()
```

### switch-eight-off.py

```
1 #!/usr/bin/python
2 import RPi.GPIO as GPIO
3 import time
4
5 GPIO.setmode(GPIO.BCM)
6 # init list with pin numbers
7
8 GPIO.setup(21, GPIO.OUT)
9 GPIO.output(21, GPIO.HIGH)
10
11 try:
12     GPIO.output(21, GPIO.HIGH)
13     # print "Relay Input 1 - Gpio: 21 - Led is off"
14
15 # End program cleanly with keyboard
16 except KeyboardInterrupt:
17     print "  Quit"
18
19 # Reset GPIO settings
20 GPIO.cleanup()
```

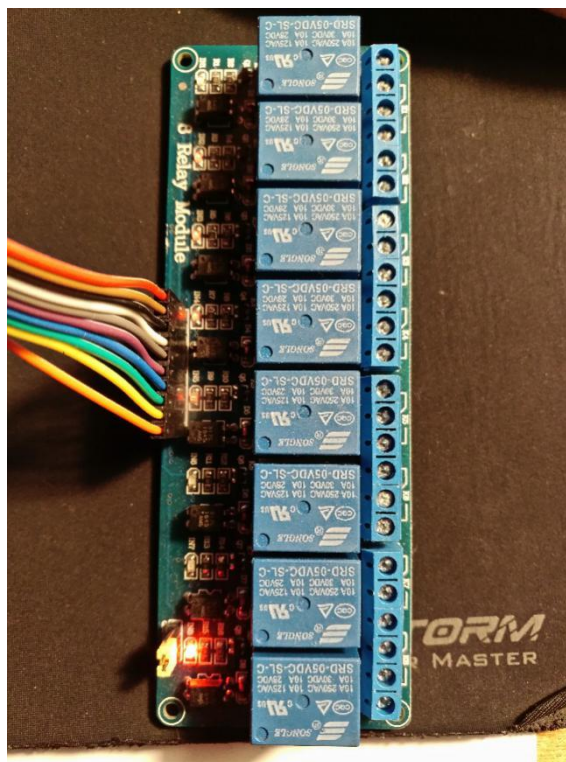
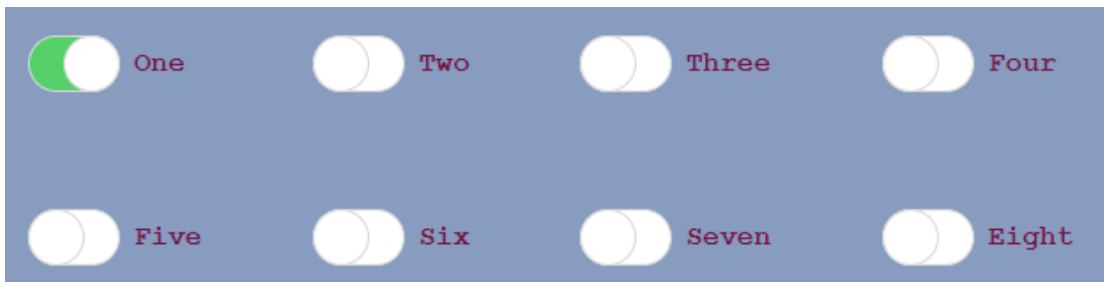
### Σημαντική σημείωση:

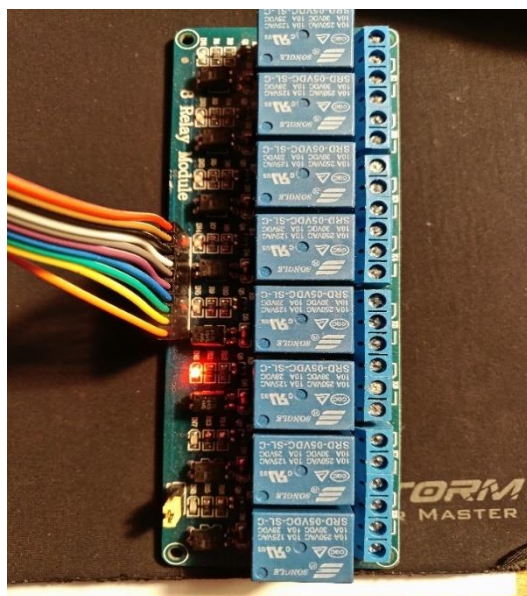
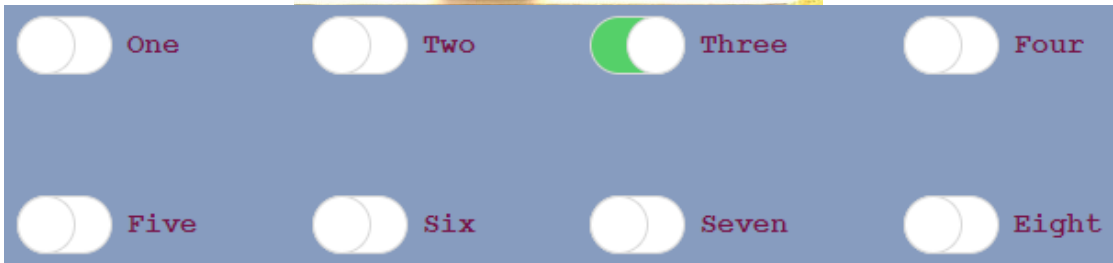
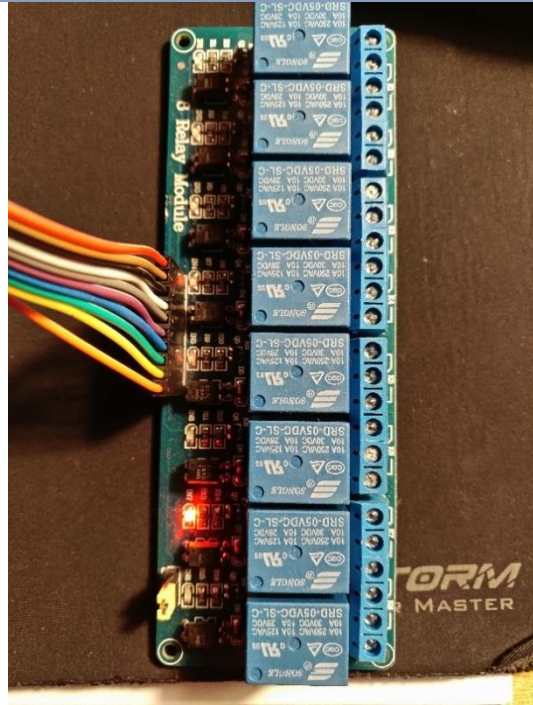
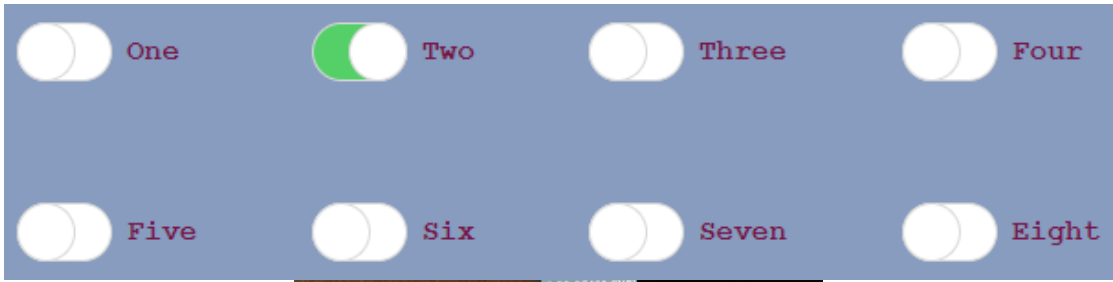
Τον Διακομιστή Apache τον διαχειρίζεται ο “**www-data**” χρήστης ο οποίος δεν συγκαταλέγεται μέσα στο αρχείο “**/etc/sudoers**”, οπότε και δεν έχει τα δικαιώματα να “τρέξει” τα παραπάνω Python Scripts. Για να μπορέσει να εκτελέσει τα παραπάνω Python Scripts, θα πρέπει να του δωθούν “**sudo**” δικαιώματα. Για να επιτευχθεί αυτό απλώς προσθέτει τον χρήστη “**www-data**” μέσα στο αρχείο με τους “**sudoers**” κάνοντας επεξεργασία στο αρχείο “**/etc/sudoers**” και προσθέτοντας την παρακάτω γραμμή από κώδικα στο τέλος του αρχείου:

```
www-data ALL=(ALL) NOPASSWD:ALL
```

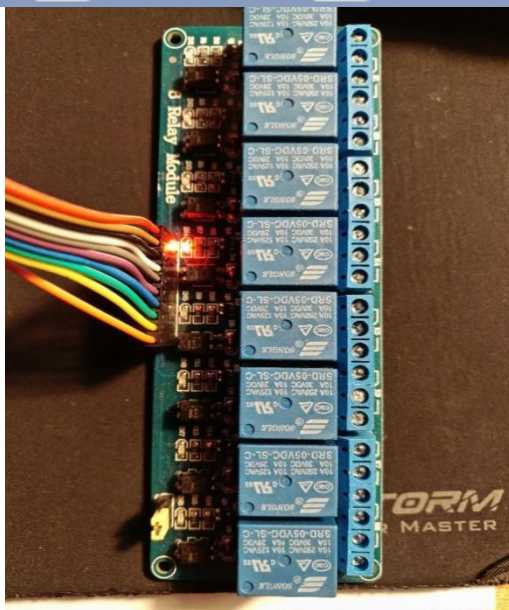
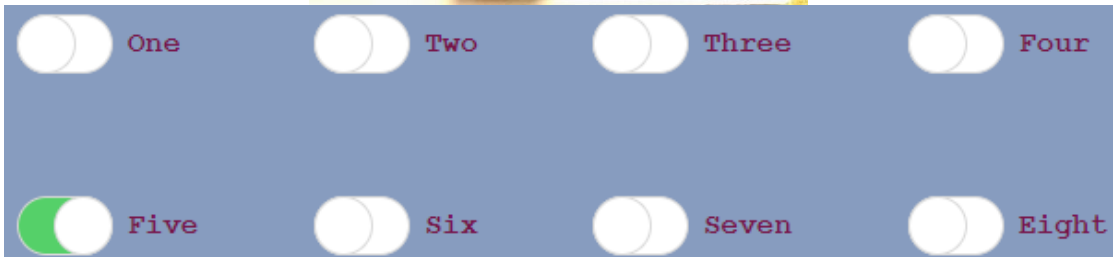
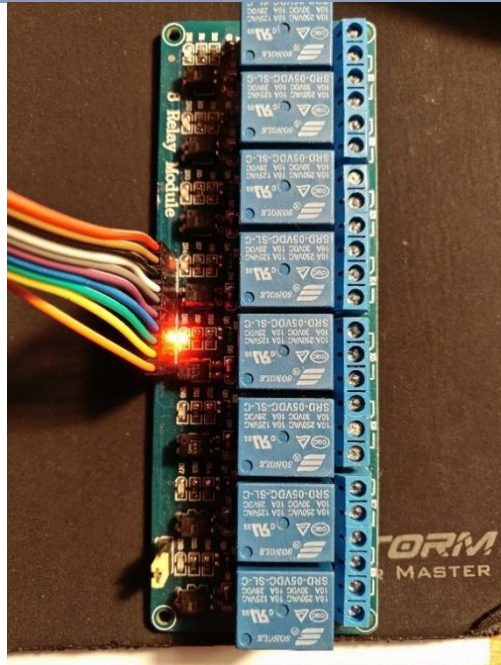
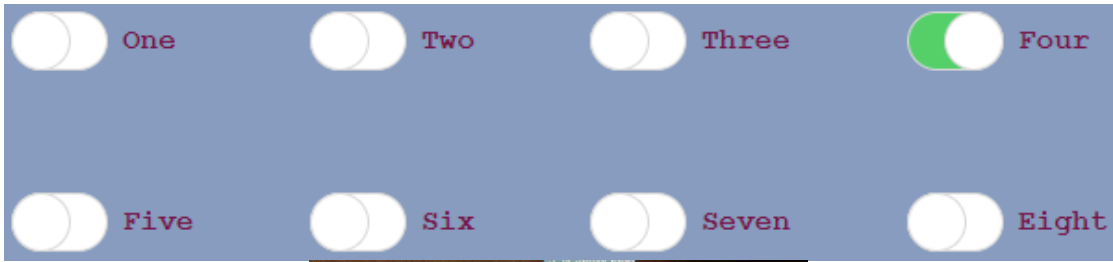
### 8.3. Δοκιμάζοντας την WEB εφαρμογή

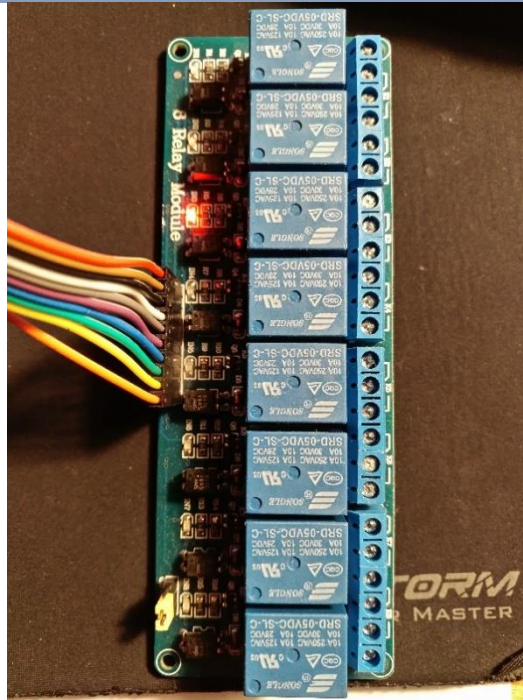
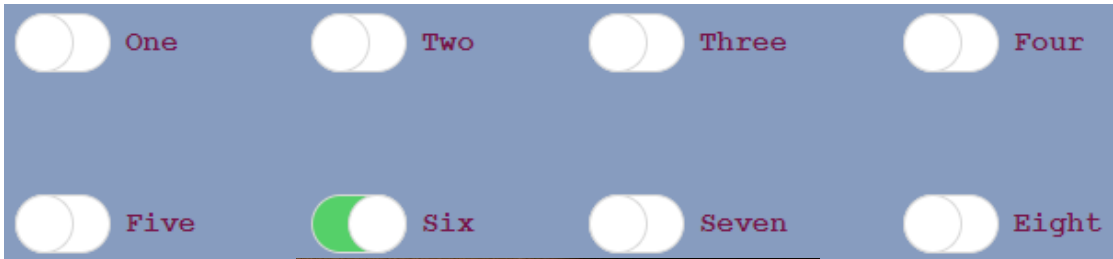
Παρακάτω, εμφανίζονται κάποια screenshot διαφορετικών σεναρίων της εφαρμογής

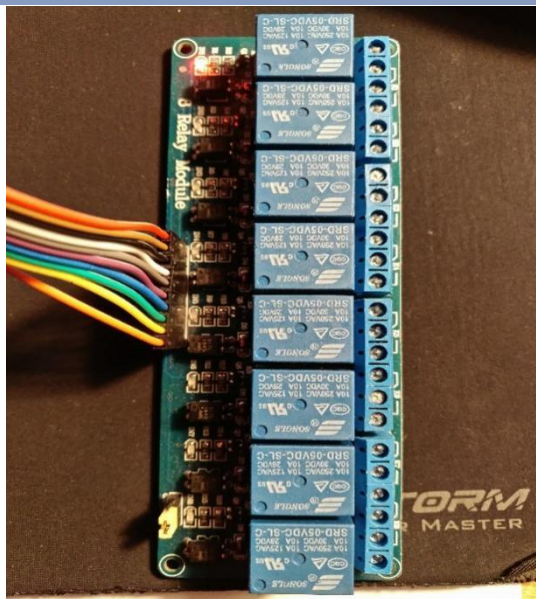
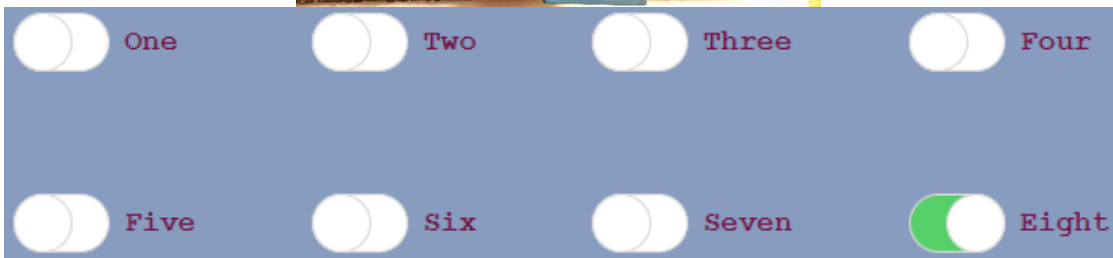
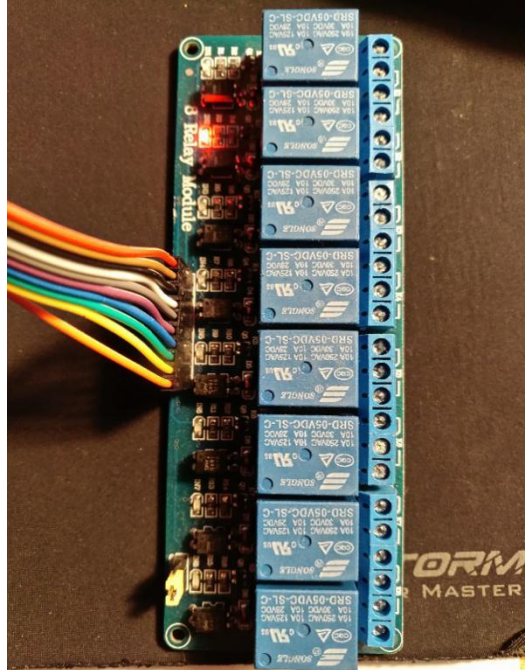
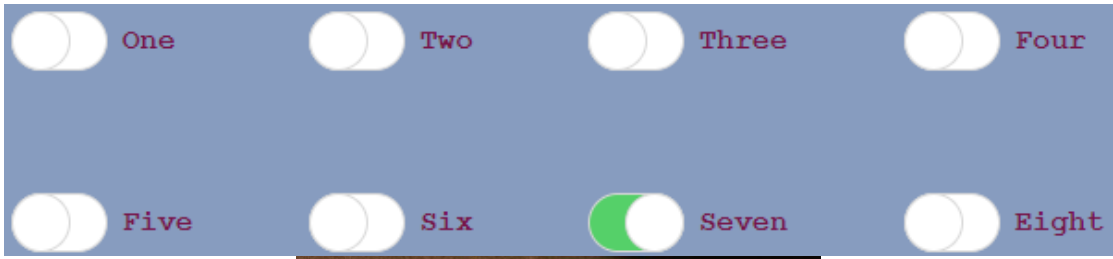












## 9. Αποτελέσματα

Στην παρούσα πτυχιακή καθίσταται σαφές ότι με ένα σχετικά μικρό γνωσιακό υπόβαθρο είναι εφικτό ο χρήστης να είναι σε θέση να πειραματιστεί με μια πληθώρα αντικειμένων που συνθέτουν με τη σειρά τους ένα μεγαλύτερο και πλήρως λειτουργικό project. Η παρούσα πτυχιακή αποσκοπούσε, όπως και επετεύχθη, στην παροχή τρόπων και μέσων ελέγχου για την αυτοματοποίηση καθημερινών αναγκών. Αξίζει να σημειωθεί ότι το κόστος της εργασίας παρέμεινε χαμηλό χρησιμοποιώντας φθηνό υλικό hardware που παρέχεται ευρέως από την αγορά σε μεγάλη κλίμακα προσιτό στον μέσο πολίτη. Ιδιαίτερα βοηθητικό ρόλο στην εκπόνηση της εργασίας διατέλεσε η τεχνολογία ανοιχτού κώδικα που επιτρέπει στον χρήστη την διαμόρφωση ήδη υπάρχοντα κώδικα σύμφωνα με τις εκάστοτε ανάγκες του. Η ενεργοποίηση και απενεργοποίηση των καναλιών του ρελέ αφήνει τον εκάστοτε χρήστη να επιλέξει ανάμεσα στις απεριόριστες δυνατότητες που διαθέτει το IoT, και να συνδέσει οποιαδήποτε συσκευή πάνω σε αυτό σύμφωνα με τις ανάγκες του. Κατα αυτόν τον τρόπο επετεύχθη η ενεργοποίηση και απενεργοποίηση των καναλιών του ρελέ που επιτρέπει στον χρήστη να επιλέξει ανάμεσα στις απεριόριστες δυνατότητες που διαθέτει το IoT, και να συνδέσει οποιαδήποτε συσκευή πάνω σε αυτό χωρίς να είναι αναγκαία η φυσική του παρουσία στο χώρο.

## 10.Βιβλιογραφία

1. Παπαθαστοπούλου Αλεξάνδρα-Internet of Things
2. Δ.Καλλιγερόπουλος-Σ.βασιλειάδου-Ιστορία των αυτομάτων
3. Internet of medical things , thematic report By Dr.Yogesh Shelke, Arpit Sharma
4. Nathan Chandler-what is the history of the remote control?
5. Raspberry Pi Basics. (n.d.).
6. Hakima Chaouchi-The internet of things connecting objects
7. Timothy L. Warner -Hacking Raspberry Pi.
8. Peter Membrey and David Hows - Learn Raspberry Pi with Linux.
9. Agus Kurniawan (2014). Raspberry Pi I/O Programming Using Python.
10. Dave Evans (2011). The Internet of Things, How the Next Evolution of the Internet Is Changing Everything.
11. Hannelore Marginean, Daniel Karzel, Tuan-Si Tran (2016). A Reference Architecture for the Internet of Things.
12. Friedemann Mattern and Christian Floerkemeier (2015). From the Internet of Computers to the Internet of Things.
13. Brandon Rhodes and John Goerzen (2014). Foundations of Python Network Programming.
14. Samuel Greengard (2015). The Internet of Things.
15. Πετρος κουλαλης.Το IOT σαν εργαλείο μάθησης
16. Βασικές εντολές linux
17. Ιωάννης Ε. Λιβιέρης-Εγχειρίδιο για το λειτουργικό σύστημα Linux
18. Steven Goodwin (2015). Smart Home Automation with Linux and Raspberry Pi.
19. Python software foundation-command line and environment
20. Python software foundation-Python libraries
21. Christian Benvenuti (2006). Understanding Linux Network Internals.
22. Alessandro Bassi, Martin Bauer, Martin Fiedler, Thorsten Kramp, Rob van Kranenburg, Sebastian Lange, Stefan Meissner (2013). Enabling Things to Talk: Designing IoT solutions with the IoT Architectural Reference Model.

23. Sergey Balandin, Yevgeni Koucheryavi, Sergey Andreev (2015). Internet of Things, Smart Spaces and Next Generation Networks and Systems.
24. Adrian McEwen & Hakim Cassimally (2014). Designing the Internet of Things.
25. Sparkfun-learn gpio library
26. Ernie Hayden-An abbreviated history of automation & industrial control systems and Cybersecurity