



ΤΕΧΝΟΛΟΓΙΚΟ
ΕΚΠΑΙΔΕΥΤΙΚΟ
ΙΔΡΥΜΑ ΚΡΗΤΗΣ

ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ ΚΡΗΤΗΣ
ΣΧΟΛΗ ΕΦΑΡΜΟΣΜΕΝΩΝ ΕΠΙΣΤΗΜΩΝ
ΤΜΗΜΑ ΗΛΕΚΤΡΟΝΙΚΩΝ ΜΗΧΑΝΙΚΩΝ Τ.Ε.

Πτυχιακή Εργασία

Ανάπτυξη Web Interface για απομακρυσμένο έλεγχο με τη
χρήση του μικροελεγκτή RaspberryPi

Λυράκης Ευάγγελος

A.M. 4196

Επιβλέπων Καθηγητής: Πετρίδης Κωνσταντίνος

Χανιά, Σεπτέμβριος 2015

Πρόλογος:

Ο σκοπός της παρούσας Πτυχιακής Εργασίας πραγματεύεται και αναλύει την ανάπτυξη πλατφόρμας που βασίζεται στο μικροελεγκτή Raspberry Pi για τον απομακρυσμένο έλεγχο διαφόρων ηλεκτρονικών συστημάτων

Πιο συγκεκριμένα, παρουσιάζεται και αναπτύσσεται όλο το λογισμικό που χρειάζεται για τη δημιουργία ενός απλού Web Interface, το οποίο θα μας επιτρέπει τον απομακρυσμένο έλεγχο ηλεκτρικών συσκευών από τα GPIO του Raspberry Pi. Επίσης, γίνεται αναλυτική παρουσίαση για το πως ο αναγνώστης μπορεί να στήσει εξ αρχής ένα καινούριο Raspberry Pi και έναν Web Server.

Ευχαριστίες:

Θα ήθελα να ευχαριστήσω τον καθηγητή Κωνσταντίνο Πετρίδη για την εμπιστοσύνη που μου έδειξε δίδοντας μου την ευκαιρία να ασχοληθώ με το θέμα αυτής της εργασίας.

Περιεχόμενα

1	ΑΥΤΟΜΑΤΙΣΜΟΣ.....	6
1.1	Τι είναι ο Αυτοματισμός.....	6
1.2	Πλεονεκτήματα του Αυτοματισμού.....	6
1.3	Αρχές Αυτοματισμού.....	7
1.3.1	Μηχανοποίηση.....	7
1.3.2	Συνοχή Διαδικασίας.....	7
1.3.3	Αυτόματος Έλεγχος.....	8
1.3.4	Αιτιολόγηση Αυτοματισμού.....	8
1.4	Ιστορία του Αυτοματισμού.....	9
1.5	Ο αυτοματισμός στην βιομηχανία.....	12
1.6	Οικιακές εφαρμογές αυτοματισμού.....	13
2	ΘΕΩΡΗΤΙΚΟ ΥΠΟΒΑΘΡΟ.....	14
2.1	Μικροελεγκτές και μικροεπεξεργαστές.....	14
2.1.1	Γενικές πληροφορίες.....	14
2.1.2	Βασικά χαρακτηριστικά που καθιστούν προτιμότερο τον μικροελεγκτή.....	17
2.1.3	Αρχιτεκτονικές μικροελεγκτών και μικροεπεξεργαστών.....	19
2.1.4	Ανάλυση μικροελεγκτή.....	20
2.2	Raspberry Pi.....	22
2.2.1	Επισκόπηση του Raspberry Pi Board.....	23
2.2.2	Λειτουργικό σύστημα για το Raspberry Pi.....	25
2.2.3	Πλεονεκτήματα και Μειονεκτήματα του Raspberry.....	26
2.3	Web Server.....	26
2.3.1	Τι είναι ένας Server.....	27
2.3.2	Κατηγορίες Server.....	27
2.3.3	Χαρακτηριστικά ενός Server.....	28
2.4	Πρωτόκολλο Μεταφοράς Υπερκειμένου (HyperText Transfer Protocol - HTTP).....	29

**Ανάπτυξη Web Interface για απομακρυσμένο έλεγχο
με τη χρήση του μικροελεγκτή RaspberryPi**

2.4.1	Τι είναι το HTTP	30
2.4.2	Πως λειτουργεί το HTTP	30
2.4.3	Πρωτόκολλο ασφαλούς μεταφοράς υπερκειμένου (HyperText Transfer Protocol Secure - HTTPS)	35
2.4.4	Διαφορές HTTP και HTTPS.....	35
2.5	Η γλώσσα προγραμματισμού HTML	36
2.6	Η γλώσσα προγραμματισμού Javascript.....	37
2.6.1	Τι είναι η Javascript	37
2.6.2	Javascript και Java	38
2.6.3	Τρόποι εισαγωγής της Javascript	40
2.7	Η γλώσσα προγραμματισμού PHP (Hypertext Preprocessor)	40
2.7.1	Γενικά για την PHP.....	40
2.7.2	Τρόποι εισαγωγής της PHP	41
2.8	Το λειτουργικό σύστημα Raspbian	42
2.8.1	Γενικά για το Raspbian	42
2.8.2	Τι είναι το Debian	42
2.8.3	Τι είναι το Linux.....	43
2.8.4	Πλεονεκτήματα του Linux	43
3	ΠΕΡΙΓΡΑΦΗ ΔΙΑΤΑΞΗΣ.....	44
3.1	Χρησιμότητα του Raspberry Pi στην εφαρμογή μας	46
3.2	Χρησιμότητα του Modem-Router	46
3.3	Χρησιμότητα του υπολογιστή/κινητού τηλεφώνου	46
3.4	Χρησιμότητα του Relay	47
3.5	Περιγραφή διάταξης ιστοσελίδας.....	47
4	Υλοποίηση.....	48
4.1	Εγκατάσταση Raspbian στο Raspberry Pi.....	48
4.2	Απόδοση στατικής διεύθυνσης IP στο Raspberry Pi	51
4.3	Εγκατάσταση απαιτούμενων πακέτων και εφαρμογών	57

**Ανάπτυξη Web Interface για απομακρυσμένο έλεγχο
με τη χρήση του μικροελεγκτή RaspberryPi**

4.3.1	Εγκατάσταση βιβλιοθήκης WiringPi	57
4.3.2	Εγκατάσταση Apache Web Server και της γλώσσας PHP5	58
4.4	Αντιγραφή των αρχείων της ιστοσελίδας στο φάκελο του Web Server.....	59
4.5	Έλεγχος λειτουργίας.....	60
5	Παράρτημα 1	63
5.1	Περιγραφή βασικών εντολών της βιβλιοθήκης WiringPi	63
5.2	Τρόποι αρίθμησης των pins του Raspberry Pi	63
5.3	Τεχνικά χαρακτηριστικά των εκδόσεων του Raspberry Pi.....	64
5.4	Δευτερεύουσες λειτουργίες των ακροδεκτών του Raspberry Pi.....	65
5.5	GIT	66
6	Παράρτημα 2	67
6.1	Επεξήγηση κώδικα	67
6.1.1	Κώδικας αρχείου gpio.php.....	67
6.1.2	Κώδικας αρχείου index.php.....	68
6.1.3	Κώδικας αρχείου script.js.....	70
7	Βιβλιογραφία.....	82

1 ΑΥΤΟΜΑΤΙΣΜΟΣ

1.1 Τι είναι ο Αυτοματισμός

Αυτοματισμός σημαίνει η τυποποίηση μίας διαδικασίας μέσω της εύρεσης καλώς ορισμένων βημάτων τα οποία πρέπει να ακολουθηθούν για να παραχθεί κάποιο επιθυμητό αποτέλεσμα. Έτσι ο αυτοματισμός δεν είναι τίποτα άλλο παρά η εύρεση ενός αλγορίθμου για την επίλυση ενός προβλήματος, ή η κατασκευή ενός αυτόνομου μηχανισμού που εκτελεί αυτόν τον αλγόριθμο για κάποια είσοδο χωρίς ανθρώπινη παρέμβαση. Ο αυτοματισμός γενικά, υπονοεί την ανεξάρτητη διαχείριση ή δράση ή αυτό-ρύθμιση, χωρίς την ανθρώπινη επέμβαση. Ο αυτοματισμός περιλαμβάνει τις μηχανές, τα εργαλεία, τις συσκευές, τις εγκαταστάσεις και τα συστήματα που είναι όλα πλατφόρμες που αναπτύσσονται από τους ανθρώπους για να εκτελέσουν ένα δεδομένο σύνολο δραστηριοτήτων χωρίς την ανθρώπινη συμμετοχή κατά τη διάρκειά τους. Βέβαια, υπάρχουν πολλές παραλλαγές αυτού του ορισμού. Ο αυτοματισμός στηρίζεται εννοιολογικά στη θεωρία ελέγχου και στους μηχανισμούς ανάδρασης. Το αντικείμενο του αυτοματισμού είναι γενικό και πολύπλευρο, και οι εφαρμογές του βρίσκονται πολυάριθμες στην καθημερινή ζωή και στην βιομηχανία.

1.2 Πλεονεκτήματα του Αυτοματισμού

- Η αντικατάσταση του ανθρώπινου παράγοντα σε διαδικασίες που είτε απαιτούν δύσκολες χειρονακτικές εργασίες είτε πραγματοποιούνται σε επικίνδυνα περιβάλλοντα.
- Η πραγματοποίηση εργασιών που ξεπερνούν τις ανθρώπινες ικανότητες όσον αφορά το βάρος, το μέγεθος, τη ταχύτητα, την αντοχή και την ακρίβεια.
- Η αύξηση της παραγωγικότητας.

- Βελτίωση της ποιότητας παραγωγής

1.3 Αρχές Αυτοματισμού

Ο αυτοματισμός μπορεί να θεωρηθεί ως ο συνδυασμός τεσσάρων βασικών αρχών: Μηχανοποίηση, συνοχή διαδικασίας, αυτόματος έλεγχος και οικονομική, κοινωνική και τεχνολογική αιτιολόγηση.

1.3.1 Μηχανοποίηση

Η μηχανοποίηση ορίζεται ως η εφαρμογή των μηχανών για την εκτέλεση μιας εργασίας. Οι μηχανές μπορούν να εκτελέσουν διάφορους στόχους, σε διαφορετικά επίπεδα πολυπλοκότητας. Όταν η μηχανοποίηση σχεδιάζεται με γνωστικές λειτουργίες και λήψεων αποφάσεων, όπως ο έλεγχος διεργασίας και ο αυτόματος έλεγχος, ο όρος του σύγχρονου αυτοματισμού γίνεται κατάλληλος. Μερικές μηχανές, βασισμένες στη δύναμη, τη πυκνότητα και τη ταχύτητά τους, μπορούν να ολοκληρώσουν στόχους που δεν θα μπορούσαν ποτέ να εκτελεσθούν από την ανθρώπινη εργασία, ανεξάρτητα από το πόση εργασία θα πραγματοποιούνταν ή το πόσο αποτελεσματικά η λειτουργία θα μπορούσε να οργανωθεί και να ρυθμιστεί. Με αυξημένη διαθεσιμότητα και σοφιστική χρήση των πηγών ισχύος και του αυτόματου έλεγχου, το επίπεδο αυτονομίας των μηχανών και των μηχανικών συστημάτων δημιούργησε μια διάκριση μεταξύ της μηχανοποίησης και του αυτοματισμού.

1.3.2 Συνοχή Διαδικασίας

Η ιδέα της συνοχής είναι να αυξηθεί η παραγωγικότητα, δηλαδή η χρήσιμη παραγωγή ανά εργατοώρα. Για παράδειγμα, οι ανεμόμυλοι μπορούν να παρέχουν σχετικά συνεχείς κύκλους της άλεσης σιταριού. Στις αρχές του 20ου αιώνα, με την εμφάνιση της μαζικής παραγωγής, η

συνοχή της διαδικασίας οργάνωνε καλύτερα τη ροή της δουλειάς. Η οργάνωση της παραγωγικής ροής και των γραμμών συναρμολόγησης και οι αυτόματες ή ημιαυτόματες γραμμές μεταφοράς, αυξήσανε τη παραγωγικότητα πέρα από τη χρήση της μηχανοποίησης. Η αναδυόμενη αυτοκινητική βιομηχανία στην Ευρώπη και τις ΗΠΑ, στις αρχές της δεκαετίας του 20ου αιώνα, χρησιμοποίησαν την έννοια κινώντας την εργασία συνεχώς, αυτόματα ή ημιαυτόματα, στις εξειδικευμένους μηχανές και τους τερματικούς σταθμούς.

1.3.3 Αυτόματος Έλεγχος

Ένας βασικός μηχανισμός του αυτόματου ελέγχου είναι η ανατροφοδότηση, η όποια είναι ο κανονισμός μιας διαδικασίας, έτσι ώστε η παραγωγή να ικανοποιεί τους όρους ενός προκαθορισμένου στόχου. Ένα παράδειγμα είναι ο ανεμόμυλος ο οποίος μπορεί να ρυθμίσει τον προσανατολισμό των λεπίδων του μέσω ανατροφοδότησης, πληροφορώντας τον για τη μεταβαλλόμενη κατεύθυνση του αέρα. Ένα άλλο παράδειγμα είναι το σύστημα θέρμανσης που μπορεί να σταματήσει και να επανεκκινήσει τη διαδικασία θέρμανσης ή ψύξης ανάλογα με την ανατροφοδότηση από το θερμοστάτη του.

1.3.4 Αιτιολόγηση Αυτοματισμού

Η αιτιολόγηση σημαίνει μια λογική και συστηματική ανάλυση, κατανόηση, και αξιολόγηση των στόχων και των περιορισμών της αυτοματοποιημένης λύσης. Ο αυτοματισμός οργανώνεται ορθολογικά με την εξέταση των τεχνολογικών και μηχανικών πτυχών στα πλαίσια των οικονομικών, κοινωνικών και διευθυντικών εκτιμήσεων, που περιλαμβάνουν επίσης: ανθρώπινους παράγοντες, οργανωτικά ζητήματα, περιβαλλοντικούς περιορισμούς και συντήρηση πόρων και ενέργειας. Θεωρητικά, όλες οι συνεχείς διαδικασίες μπορούν να ελεγχθούν αυτόματα, αλλά πρακτικά μία τέτοια αυτοματοποίηση πρέπει

να οργανωθεί ορθολογικά πρώτα. Για παράδειγμα, οι αεριωθούμενες μηχανές μπορούν να προωθηθούν συνεχώς στους μεταφορείς στα κύτταρα συνελεύσεων, αλλά εάν η ζήτηση για αυτές τις μηχανές είναι χαμηλή, δεν υπάρχει καμία αιτιολόγηση για να αυτοματοποιήσει τη ροή τους. Επιπλέον, όλη η αυτοματοποίηση πρέπει να έχει ως σκοπό να λειτουργήσει μέσα στα ασφαλή όρια, έτσι δεν θέτει τους κινδύνους στους ανθρώπους και στο περιβάλλον.

1.4 Ιστορία του Αυτοματισμού

Ο αυτοματισμός είναι ένα από τα πιο ιστορικά πεδία της επιστήμης, διότι η ανάπτυξη του συνοδεύει την εξέλιξη όλων των άλλων τεχνολογιών. Η γνώση του αυτοματισμού επομένως αποτελεί γνώση της τεχνολογικής ιστορίας. Στην περίπτωση του αυτοματισμού, η ιστορία αυτή είναι ιδιαίτερα πλούσια καθώς ήδη από τους Αρχαίους Έλληνες υπήρξε ιδιαίτερη εφευρετικότητα και αναπτύχθηκαν πολλές και σημαντικές λύσεις αυτοματισμού που χρησιμοποιούμε μέχρι και σήμερα.

- Ατμοστρόβιλος του Ήρωνα



Η πρώτη ατμομηχανή ανακαλύφθηκε από τον Ήρωνα η οποία αποτελούταν από ένα κλειστό δοχείο το οποίο περιείχε νερό. Όταν το νερό άρχιζε να βράζει ο ατμός με την βοήθεια σωλήνων κατευθυνόταν στο πάνω μέρος σε μια γυάλινη σφαίρα με δύο αντιδιαμετρικές εξόδους. Η ταχύτητα εξόδου του ατμού είχε σαν συνέπεια την περιστροφή της γυάλινης σφαίρας έτσι αυτή η κυκλική κίνηση θα μπορούσε να δώσει κίνηση σε αρκετές μηχανές της εποχής.

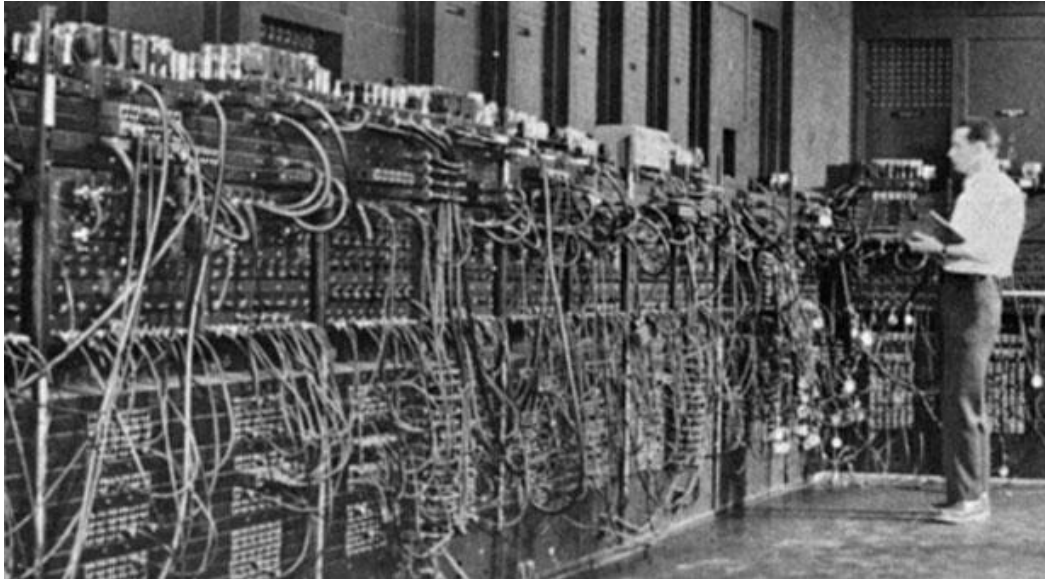
- Ο Μηχανισμός των Αντικυθήρων



Ο Μηχανισμός των Αντικυθήρων (≈ 100 π.Χ.) είναι ο αρχαιότερος αυτοματισμός που σώζεται ως σήμερα. Μπορούσε να προβλέψει τις θέσεις των πλανητών.

Όπως καταλαβαίνουμε ο άνθρωπος ήδη από την αρχαιότητα προσπαθούσε να ανακαλύψει τρόπους ούτως ώστε να διευκολύνει τη ζωή του. Η εξέλιξη του αυτοματισμού όμως ακολούθησε την πορεία εξέλιξης της τεχνολογίας. Το μεγάλο άλμα του αυτοματισμού έγινε όταν άρχισε να γίνεται η χρήση του ηλεκτρισμού και των ηλεκτρονικών εξαρτημάτων. Ας δούμε μερικές εφευρέσεις της σημερινής τεχνολογίας.

- Ο πρώτος ηλεκτρονικός υπολογιστής



Ο πρώτος ηλεκτρονικός υπολογιστής που δημιουργήθηκε το 1946 επονομάστηκε ENIAC. Αντί για μηχανικά μέρη χρησιμοποιούσε ηλεκτρονικές λυχνίες. Ο ENIAC ήταν τεράστιος καθώς καταλάμβανε 163 τετραγωνικά μέτρα χώρο και ζύγιζε 30 τόνους.

- PLC (Programmable Logic Controller)



Το PLC είναι μια ηλεκτρονική προγραμματιζόμενη συσκευή βασισμένη σε μικροεπεξεργαστή ο οποίος έχει την δυνατότητα να εκτελεί μόνο ένα πρόγραμμα κάθε φορά σε πραγματικό χρόνο. Ο σκοπός κάθε προγράμματος που αποθηκεύεται στην μνήμη είναι να εξηγεί στο PLC πώς να αλληλεπιδρά με το περιβάλλον του.

1.5 Ο αυτοματισμός στην βιομηχανία

Ο βιομηχανικός αυτοματισμός ασχολείται με την αυτοματοποιημένη λειτουργία και παρακολούθηση της βιομηχανικής παραγωγής, με χρήση τεχνολογιών μέτρησης, καταγραφής και ελέγχου. Επίσης, ενώ αρχικά ο αυτοματισμός εισήχθη στη βιομηχανία για την αύξηση της παραγωγής και τη μείωση του κόστους παραγωγής, πλέον χρησιμεύει περισσότερο για την καλύτερη αποτελεσματικότητα στη παραγωγή καθώς μειώνει αισθητά το ανθρώπινο σφάλμα και τις απώλειες των πρώτων υλών.

Όπως αναφέραμε και πιο πάνω, οι αυτοματισμοί εκτός από το κομμάτι της παραγωγής λαμβάνουν χώρα και στο κομμάτι του έλεγχου όπως είναι η ασφάλεια και η οικονομία. Για παράδειγμα είναι πολύ σημαντικό σε ένα εργοστάσιο παραγωγής καυσίμων να υπάρχουν αυτοματοποιημένες διαδικασίες ελέγχου με τις οποίες θα αποφεύγονται σοβαρά εργατικά ατυχήματα. Εξίσου σημαντικό είναι και ο έλεγχος του οικονομικού σκέλους σε μια επιχείρηση όπως για παράδειγμα ο έλεγχος κατανάλωσης ισχύος.

Υπάρχουν πολλά παραδείγματα εργοστασιακού αυτοματισμού εφόσον ότι παράγεται στις μέρες μας γίνεται μέσω αυτοματοποιημένων διαδικασιών, όπως για παράδειγμα η αυτοκινητοβιομηχανία ή η παραγωγή ειδών πρώτης ανάγκης.

1.6 Οικιακές εφαρμογές αυτοματισμού

Ο έλεγχος του οικιακού περιβάλλοντος αναφέρεται στον έλεγχο των διαφόρων παραμέτρων, συσκευών και συστημάτων, με τα οποία ο ένοικος αλληλεπιδρά. Οι αντιδράσεις υλοποιούνται κατά κανόνα με κινήσεις ανθρωπίνων μελών και με την ομιλία. Μέσω των αντιδράσεων αυτών μπορεί ο άνθρωπος να ενεργοποιήσει διάφορα συστήματα προς επιτέλεση διάφορων λειτουργιών. Τέτοια συστήματα είναι τα συστήματα κλιματισμού, ασφάλειας κ.α. Ιδιαίτερη σημασία αποκτούν οι διαδικασίες σε περιπτώσεις ανθρώπων με προβλήματα, όπως είναι τα άτομα με κινητικά προβλήματα.

Η τεχνολογία σήμερα προσφέρει τη δυνατότητα πλήρους αυτοματοποίησης των οικιακών λειτουργιών, καθόσον οι περισσότερες συσκευές ήδη ελέγχονται από εξελιγμένους επεξεργαστές συμβατικής λογικής και τεχνητής νοημοσύνης (ψυγεία, πλυντήρια, κουζίνες, κλιματιστικά κ.λπ.). Η διασύνδεση των ελεγκτών αυτών σε ένα σύστημα κατανομημένου ελέγχου, όπου ως κεντρικός υπολογιστής μπορεί να λειτουργήσει ένας προσωπικός υπολογιστής ή ένας προγραμματιζόμενος ελεγκτής είναι λύση, η οποία είχε προταθεί και παλαιότερα, είναι δε πιο ενδιαφέρουσα στα πλαίσια της λειτουργίας του διαδικτύου. Ο προγραμματισμός των συστημάτων αυτών είναι εύκολος ακόμα και από μη ειδικούς χρήστες, καθόσον έχουν αναπτυχθεί συστήματα γραφικού προγραμματισμού, τα οποία εξασφαλίζουν ένα συμπαγές, εύχρηστο και φιλικό περιβάλλον προγραμματισμού και λειτουργίας.

Χαρακτηριστικά παραδείγματα οικιακών εφαρμογών του αυτοματισμού είναι ο έλεγχος του φωτισμού, συστήματα συναγερμού (παραβίασης, φωτιάς κ.α.), αυτόνομη θέρμανση, αυτόματος κλιματισμός, διαχείριση και έλεγχος της κατανάλωσης ενέργειας κ.α.

2 ΘΕΩΡΗΤΙΚΟ ΥΠΟΒΑΘΡΟ

2.1 Μικροελεγκτές και μικροεπεξεργαστές

2.1.1 Γενικές πληροφορίες

Ο μικροελεγκτής είναι ένα αυτόνομο υπολογιστικό σύστημα, με πολύ μικρό μέγεθος, σε ένα και μοναδικό ολοκληρωμένο κύκλωμα. Όπως και όλα τα VLSI κυκλώματα, αποτελείται από μέρη που κατασκευάζονται με διάφορες λιθογραφικές μεθόδους πάνω σε πλάκες πυριτίου. Πάνω σε

αυτά σχηματίζονται χιλιάδες έως εκατομμύρια τρανζίστορ και κατ' επέκταση δημιουργούνται τα λεγόμενα ολοκληρωμένα κυκλώματα που είναι συνδυασμός λογικών πυλών. Συνδυάζοντας τις λογικές πύλες, δημιουργούνται υπομονάδες που επιτελούν ορισμένες πιο εξειδικευμένες λειτουργίες στον μικροελεγκτή. Η επιθυμία των κατασκευαστών συστημάτων να δημιουργήσουν συστήματα με περισσότερες δυνατότητες και με μικρότερο μέγεθος οδήγησε στην ανάγκη για ενσωμάτωση όλων των λειτουργιών ενός υπολογιστή σε ένα ή μερικά ολοκληρωμένα κυκλώματα. Η ανάπτυξη της τεχνολογίας που χρησιμοποιήθηκε για την κατασκευή των κυκλωμάτων αυτών, είχε ως αποτέλεσμα την ανάπτυξη του μικροεπεξεργαστή.

Παρότι μικροεπεξεργαστές και μικροελεγκτές είναι παραλλαγές στο ίδιο θέμα, δηλαδή, στην ολοκλήρωση ενός γενικής χρήσεως επεξεργαστή σε ένα ολοκληρωμένο κύκλωμα (IC - Integrated Circuit), υπάρχουν και διαφορές:

- Μικροεπεξεργαστής είναι η υλοποίηση ενός γενικής χρήσεως επεξεργαστή σε ένα ολοκληρωμένο κύκλωμα με κύριο κριτήριο την απόδοση. Η ολοκλήρωση σκοπό έχει την μεγιστοποίηση της ταχύτητας του επεξεργαστή. Εφόσον υπάρχει δυνατότητα (λόγω διαθέσιμων ημιαγωγών στοιχείων στο υπόστρωμα), επεκτείνεται με κριτήριο την απόδοση (π.χ. υποστήριξη πράξεων κινητής υποδιαστολής - Floating Point), με υποστήριξη της λειτουργικότητας της μνήμης (με ενσωμάτωση μονάδας διαχείρισης μνήμης MMU – Memory Management Unit), με υποστήριξη μέρους της ιεραρχίας μνήμης (με ενσωμάτωση κρυφής μνήμης ενός ή και δύο επιπέδων), κλπ.
- Μικροελεγκτής είναι η υλοποίηση ενός γενικής χρήσεως επεξεργαστή σε ένα ολοκληρωμένο κύκλωμα με κύριο γνώμονα την αυτοτέλεια. Η ολοκλήρωση σκοπό έχει την μεγιστοποίηση της δυνατότητας του μικροελεγκτή να χρησιμοποιηθεί με ελάχιστα ή και καθόλου κυκλώματα υποστήριξης. Επειδή σχεδόν όλοι οι μικροελεγκτές χρησιμοποιούνται σε ενσωματωμένες εφαρμογές, παρέχουν στήριξη σε επίπεδο υλικού αλλά και σε επίπεδο εντολών για πρωταρχικές λειτουργίες όπως η μέτρηση χρόνου (με

χρονιστές – timers), η μέτρηση συμβάντων (με μετρητές – counters), η σύγκριση αναλογικού σήματος με κάποιο προγραμματιζόμενο κατώφλι (analog signal comparators), η υποστήριξη επικοινωνίας μέσω τυποποιημένου σειριακού πρωτοκόλλου (όπως RS 232, USB, κλπ.), καθώς και υποστήριξη δομών που αυξάνουν την αξιοπιστία (π.χ. με μετρητές επανεκκίνησης - watchdog timers).

Από τις παραπάνω περιγραφές βλέπουμε πως παρά τα κοινά τους στοιχεία, μικροεπεξεργαστές και μικροελεγκτές δεν είναι ποιοτικά πανομοιότυποι. Παρόλο που και οι δύο έχουν σχεδιαστεί για εφαρμογές πραγματικού χρόνου, ένας μικροεπεξεργαστής είναι ένα ολοκληρωμένο κύκλωμα το οποίο περιέχει μόνο τη μονάδα επεξεργασίας (CPU - Central Processing Unit) ενώ οι μνήμες RAM, ROM και τα υπόλοιπα περιφερειακά κυκλώματα πρέπει να προστεθούν εξωτερικά. Αντιθέτως, οι μικροελεγκτές μαζί με τη μονάδα επεξεργασίας περιέχουν και ένα συγκεκριμένο ποσοστό μνήμης RAM και ROM μαζί με ορισμένα άλλα περιφερειακά κυκλώματα, όλα σε ένα chip. Επί πλέον, δεν είναι πανομοιότυποι και σε ποσοτικά στοιχεία όπως το datapath. Ειδικά για μικρών υπολογιστικών απαιτήσεων εφαρμογές (π.χ. φούρνους μικροκυμάτων), ένας μικροελεγκτής με 4-bit ή 8-bit που κοστίζει 1-2 Ευρώ επαρκεί πλήρως. Οι μικροελεγκτές έχουν ως κύριο σκοπό τους να επικοινωνούν με άλλες ηλεκτρονικές διατάξεις και όχι με τον άνθρωπο. Είναι αρκετά βεβαρημένοι με υπολογιστικό φόρτο υπολογιστές, κρυμμένοι σε διάφορες ηλεκτρονικές συσκευές και οι λειτουργίες τους είναι από περιορισμένες έως πολύπλοκες εργασίες ελέγχου. Κάποιες από τις καθημερινές συσκευές, που ελέγχονται εν μέρει ή και πλήρως ακόμη από κάποιον μικροελεγκτή, είναι ηλεκτρονικά ρολόγια, φούρνοι μικροκυμάτων, ηλεκτρικά πλυντήρια, «εγκέφαλοι» αυτοκινήτων κτλ. Επιπλέον, εκτός από εφαρμογές καθημερινής χρήσης, οι μικροελεγκτές χρησιμοποιούνται και κατά κόρον σε πληθώρα από επιστημονικούς τομείς για πειραματισμό και έρευνα. Ο προγραμματισμός αυτών των συσκευών γίνεται με ένα σετ εντολών που έχουν συγκεκριμένη σύνταξη και χρησιμοποιούν δύο καταστάσεις το 1 και το 0. Η γλώσσα που αντιλαμβάνεται ένας μικροελεγκτής όπως και ένας μικροεπεξεργαστής

ονομάζεται γλώσσα μηχανής. Στην πράξη ο χρήστης προγραμματίζει τον μικροελεγκτή σε μνημονική γλώσσα Assembly. Η γλώσσα Assembly γενικά θεωρείται χαμηλού επιπέδου καθώς βρίσκεται πολύ κοντά στην γλώσσα μηχανής. Είναι δυνατόν να γράψουμε προγράμματα σε αυτή τη γλώσσα με την βοήθεια εργαλείων που ονομάζονται assemblers. Έτσι, όταν ο προγραμματιστής δώσει εντολές όπως την MOV, ADD, LD, (Μετακίνηση, Πρόσθεση, Φόρτωση) ο assembler αντιστοιχεί την εντολή με μια ακολουθία 0 και 1 που είναι κατανοητή από τον μικροελεγκτή και τα υποσυστήματα του. Ωστόσο, ακόμη και με αυτή την διευκόλυνση που προσφέρουν τα εργαλεία αυτά, είναι αρκετά οδυνηρό από άποψη χρόνου να γραφεί ένα πρόγραμμα τέτοιου επιπέδου. Οι εντολές της γλώσσας μηχανής επιτελούν περιορισμένες διεργασίες και πολλές φορές χρειάζεται εκατοντάδες εντολών για να εκτελέσουν πράξεις και λειτουργίες πιο σύνθετες από τις συνηθισμένες. Υπάρχει όμως η δυνατότητα να γράψουμε το πρόγραμμα σε γλώσσα προγραμματισμού υψηλού επιπέδου. Οι γλώσσες υψηλού επιπέδου μπορούν να μεταφραστούν σε γλώσσα χαμηλού επιπέδου με τη βοήθεια μεταφραστικών εργαλείων, όπως είναι ο compiler (μεταγλωττιστής) και ο interpreter (διερμηνέας). Το πρώτο είναι εργαλείο λογισμικού, το οποίο δέχεται τις εντολές υψηλού επιπέδου που δίνει ο χρήστης και τις μετατρέπει σε γλώσσα μηχανής. Ο διερμηνέας από την άλλη, είναι υλικό (hardware) μέσα στον μικροελεγκτή το οποίο αποκωδικοποιεί τη γλώσσα υψηλού επιπέδου άμεσα σε γλώσσα μηχανής. Αυτό απαιτεί βεβαίως επεξεργαστική ισχύ από το μέρος του μικροελεγκτή και έχει την τάση να τρέχει πιο αργά από ένα πρόγραμμα που «μεταγλωττίστηκε» (μέσω compiler) σε γλώσσα μηχανής. Ωστόσο έχει το πλεονέκτημα της αμεσότητας, καθώς ο προγραμματιστής μπορεί να αλλάξει ένα σημείο στον κώδικα του και να δει τα αποτελέσματα χωρίς το βήμα της μεταγλώττισης.

2.1.2 Βασικά χαρακτηριστικά που καθιστούν προτιμότερο τον μικροελεγκτή

Λόγω του ολοκληρωμένου πακέτου που προσφέρει, ο μικροελεγκτής εμφανίζει κάποια χαρακτηριστικά τα οποία τον καθιστούν προτιμότερο για χρήση σε εφαρμογές από το να χρησιμοποιηθούν ξεχωριστά

περιφερειακά κυκλώματα. Ορισμένα βασικά χαρακτηριστικά είναι τα εξής:

- ✓ Χαμηλό κόστος: Το μεγαλύτερο πλεονέκτημα των μικροελεγκτών είναι το χαμηλό κόστος που έχουν έναντι των μικροεπεξεργαστών. Ακόμα, ένας μικροελεγκτής είναι φτηνός σε περίπτωση αντικατάστασης, ενώ ένας μικροεπεξεργαστής είναι δέκα φορές ακριβότερος.
- ✓ Χαμηλή κατανάλωση: Οι μικροελεγκτές χρησιμοποιούν την τεχνολογία CMOS και λειτουργούν σε συχνότητες που δε ξεπερνούν τα 32KHz, με αποτέλεσμα να καταναλώνουν χαμηλές ποσότητες ισχύος της τάξεως των mW.
- ✓ Μικρότερο μέγεθος: Η ολοκλήρωση των βασικών στοιχείων που απαρτίζεται ο ίδιος ο μικροελεγκτής μείωσε τις διαστάσεις σε σχέση με τη χρήση των επιμέρους στοιχείων ως σύνολο.
- ✓ Περισσότερη αυτονομία: Μέσω της ενσωμάτωσης σύνθετων περιφερειακών κυκλωμάτων, όπως οι μνήμες, οι μικροελεγκτές έχουν μεγαλύτερη αυτονομία και ορισμένοι δεν χρειάζονται κανένα άλλο ολοκληρωμένο κύκλωμα για να λειτουργήσουν.
- ✓ Ευκολότερη υλοποίηση εφαρμογών και μεγαλύτερη αξιοπιστία: Αυτά προκύπτουν λόγω των λιγότερων και απλούστερων διασυνδέσεων των ενσωματωμένων περιφερειακών κυκλωμάτων.
- ✓ Περισσότεροι ακροδέκτες για ψηφιακές εισόδους/εξόδους: Αυτό προκύπτει λόγω του ότι δε δεσμεύονται ακροδέκτες για τη διασύνδεση διάφορων περιφερειακών κυκλωμάτων, καθώς αυτά θα περιέχονται στο ίδιο το chip του μικροελεγκτή.
- ✓ Μειωμένες εκπομπές ηλεκτρομαγνητικών παρεμβολών και μειωμένη ευαισθησία σε αντίστοιχες παρεμβολές από άλλες ηλεκτρικές και ηλεκτρονικές συσκευές: Το πλεονέκτημα αυτό προκύπτει από το μικρότερο αριθμό και μήκος εξωτερικών διασυνδέσεων καθώς και τις χαμηλότερες ταχύτητες λειτουργίας.

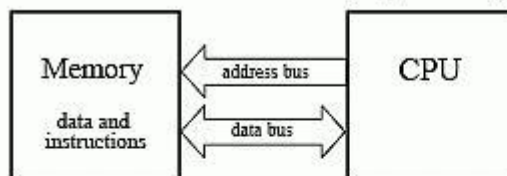


2.1.3 Αρχιτεκτονικές μικροελεγκτών και μικροεπεξεργαστών

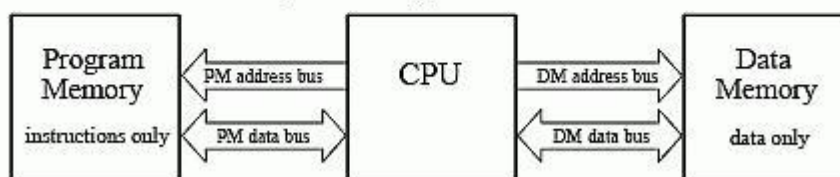
Υπάρχουν δύο αρχιτεκτονικές που συναντιούνται στους μικροελεγκτές και στους μικροεπεξεργαστές. Η αρχιτεκτονική Von Neumann και η αρχιτεκτονική Harvard. Αυτές οι δύο αρχιτεκτονικές διαφέρουν στον τρόπο που γίνεται η αποθήκευση και η προσπέλαση της μνήμης προγράμματος και της μνήμης δεδομένων.

- ❖ Στην αρχιτεκτονική Von Neumann τα δεδομένα και οι εντολές των εκτελούμενων προγραμμάτων αποθηκεύονται σε μια μοναδική μνήμη εγγραφής-ανάγνωσης, της οποίας τα περιεχόμενα μπορούν να διευθυνσιοδοτηθούν κατά κελί, χωρίς να μας ενδιαφέρει ο τύπος των δεδομένων που περιέχεται εκεί. Επίσης, η εκτέλεση των εντολών του προγράμματος πραγματοποιείται σειριακά (εκτός και αν υπάρχει ρητή διακλάδωση), από μια εντολή στην επόμενη. Υπάρχει κυρίως στους μικροεπεξεργαστές.
- ❖ Στην αρχιτεκτονική Harvard ο χώρος διευθύνσεων προγράμματος είναι ανεξάρτητος από εκείνον των δεδομένων. Το μήκος των λέξεων στο χώρο του προγράμματος μπορεί να είναι διαφορετικό από το μήκος των λέξεων στο χώρο των δεδομένων. Τη βρίσκουμε κυρίως στους μικροελεγκτές.

a. Von Neumann Architecture (*single memory*)



b. Harvard Architecture (*dual memory*)



2.1.4 Ανάλυση μικροελεγκτή

Στον μικροεπεξεργαστή, το ολοκληρωμένο κύκλωμα που τον αποτελεί περιέχει μόνο την Λογική και Αριθμητική Μονάδα (ALU), στοιχειώδεις καταχωρητές (registers), προσωρινή μνήμη RAM πολύ υψηλής ταχύτητας (cache memory) και, κάποιες φορές, τον ελεγκτή μνήμης (memory controller). Όμως σε ένα ολοκληρωμένο κύκλωμα ενός μικροελεγκτή περιέχονται ορισμένα ακόμα στοιχεία. Τα πιο βασικά είναι:

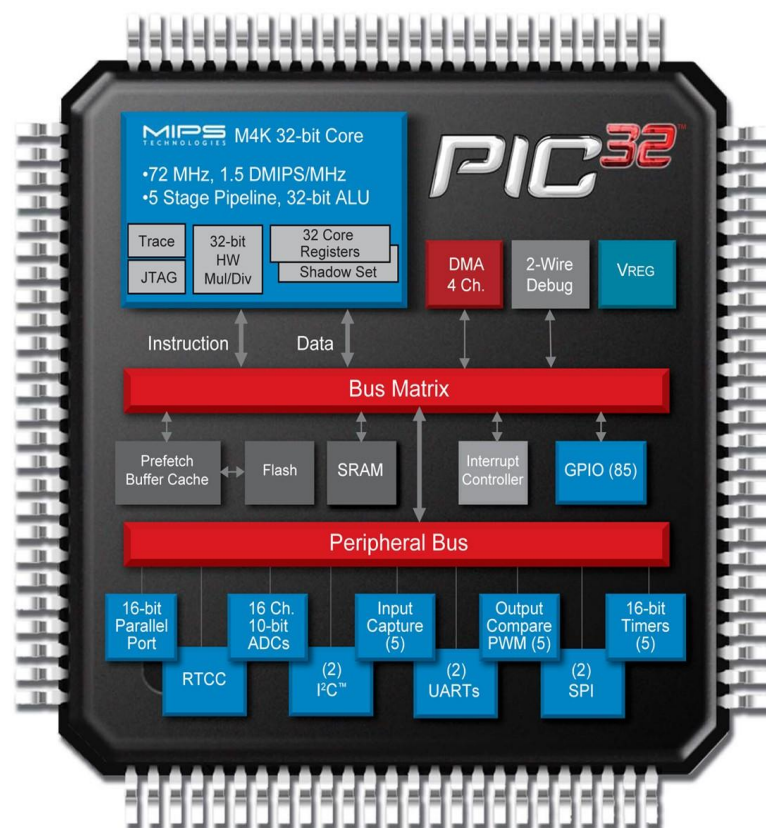
- Μνήμη προγράμματος (τύπου ROM, FLASH, EPROM κλπ.) η οποία περιέχει το λογισμικό του συστήματος. Σε κάποια μοντέλα, είναι δυνατό το κλείδωμα αυτής της μνήμης, μετά την εγγραφή της, ώστε να προστατευτεί το περιεχόμενό της από αντιγραφή.
- Μνήμη RAM.
- Μόνιμη μνήμη αποθήκευσης παραμέτρων λειτουργίας (τύπου EEPROM ή NVRAM) η οποία να μπορεί να γράφεται τον πυρήνα του μικροελεγκτή. Αυτή η μνήμη έχει, έναντι της FLASH, το πλεονέκτημα της δυνατότητας διαγραφής και εγγραφής οποιουδήποτε μεμονωμένου byte.
- Κύκλωμα αρχικοποίησης (reset).
- Διαχειριστή αιτήσεων διακοπής (interrupt request controller) από τα περιφερειακά.
- Κύκλωμα επιτήρησης τροφοδοσίας (brown-out detection) το οποίο παρακολουθεί την τροφοδοσία και αρχικοποιεί ολόκληρο το σύστημα όταν αυτή πέσει κάτω από τα ανεκτά όρια, προλαμβάνοντας έτσι την αλλοίωση των δεδομένων.
- Κύκλωμα επιτήρησης λειτουργίας (watchdog timer) το οποίο αρχικοποιεί το σύστημα, αν αυτό εμφανίσει σημάδια δυσλειτουργίας λόγω κολλήματος (hang).
- Τοπικό ταλαντωτή για την παροχή παλμών χρονισμού (clock).

- Έναν ή περισσότερους χρονιστές-απαριθμητές υψηλής ταχύτητας (hardware timer-counter) για τη δημιουργία καθυστερήσεων, μέτρηση διάρκειας γεγονότων, απαρίθμηση γεγονότων και άλλων λειτουργιών ακριβούς χρονισμού.
- Ρολόι πραγματικού χρόνου (Real Time Clock, RTC) το οποίο τροφοδοτείται από ανεξάρτητη μπαταρία και γι αυτό πρέπει να έχει πολύ χαμηλή κατανάλωση ρεύματος.
- Σειρά ανεξάρτητων ψηφιακών εισόδων και εξόδων (Parallel Input-Output, PIO).

Ανάλογα με την εφαρμογή για την οποία προορίζετε ένας μικροελεγκτής, τότε, ίσως, να περιέχει και:

- Μία ή περισσότερες ασύγχρονες σειριακές θύρες επικοινωνίας (Universal Asynchronous Receiver Transmitter, UART).
- Σύγχρονες σειριακές θύρες επικοινωνίας (πχ I²C, SPI, Ethernet).
- Ολόκληρα υποσυστήματα για την άμεση υποστήριξη από υλικολογισμικό (firmware) των πιο σύνθετων πρωτοκόλλων επικοινωνίας όπως CAN, HDLC, ISDN, ADSL.
- Μονάδα άμεσης εκτέλεσης πράξεων κινητής υποδιαστολής (Floating Point Processing Unit, FPU), η οποία είναι πάντοτε πιο γρήγορη από την ALU του επεξεργαστή. Τέτοιες μονάδες χαρακτηρίζουν τους μικροελεγκτές με δυνατότητες ψηφιακής επεξεργασίας σήματος (Digital Signal Processing, DSP). Τα τελευταία χρόνια, με την ευρύτατη διάδοση των φορητών συσκευών ήχου και εικόνας, παρατηρείται μια τάση σύγκλισης των μικροελεγκτών με τους DSP.
- Περισσότερες από μία εισόδους για μετατροπή αναλογικού σήματος σε ψηφιακό (Analog to Digital converter, ADC).
- Μετατροπέα ψηφιακού σε αναλογικό σήμα (Digital to Analog converter, DAC).

- Διαμορφωτής πλάτος παλμού (Pulse-width modulation, PWM), για υποστήριξη διάφορων συσκευών, όπως κινητήρες.
- Ελεγκτή οθόνης υγρών κρυστάλλων (Liquid Crystal Display, LCD).



Μπλοκ διάγραμμα του μικροελεγκτή PIC32 της Microchip

2.2 Raspberry Pi

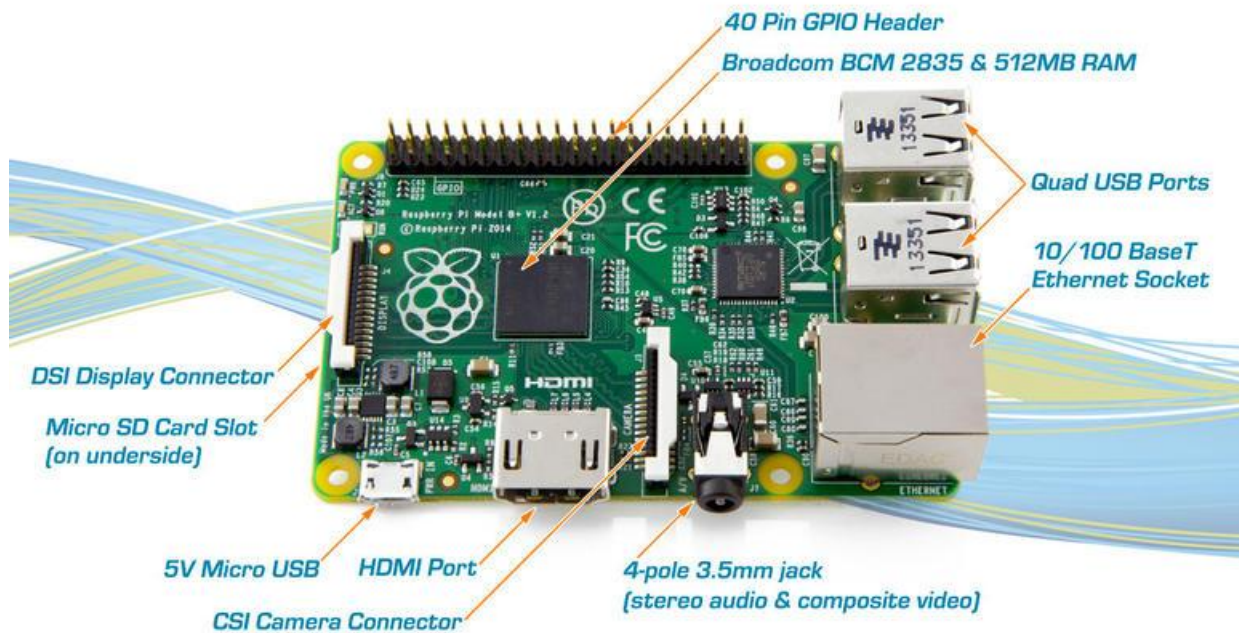
Το ρόλο του μικροελεγκτή στο σύστημα μας παίζει το Raspberry Pi, η έκδοση B+. Το Raspberry Pi αποτελεί έναν υπολογιστή σε μέγεθος πιστωτικής κάρτας ο οποίος τρέχει linux μέσω κάρτας SD (microSD στα νεότερα μοντέλα). Αναπτύχθηκε από την Raspberry Pi Foundation και σκοπό είχε την προώθηση της διδασκαλίας της επιστήμης των υπολογιστών σε σχολεία. Είναι μια συσκευή που παρουσιάζει πολλές δυνατότητες και είναι ικανή για λειτουργίες που συνήθως της κάνει ένας

προσωπικός υπολογιστής, όπως προβολή ταινιών, παιχνίδια, internet browsing κλπ. Λόγω του μεγέθους και των δυνατοτήτων του χρησιμοποιείται σε πολλά projects. Ο λόγος επιλογής αυτού του board ήταν ότι αποτελεί ένα σύγχρονο υπολογιστικό σύστημα με πολλές λειτουργικότητες και χαμηλό κόστος.

2.2.1 Επισκόπηση του Raspberry Pi Board

Έχουν κυκλοφορήσει τέσσερα μοντέλα Raspberry Pi. Το Model A, Model B και το αναβαθμισμένο μοντέλο του, Model B+, ενώ πρόσφατα κυκλοφόρησε και η έκδοση Raspberry Pi 2. Το Raspberry Pi B+, το οποίο θα χρησιμοποιήσουμε, διαθέτει τον μικροελεγκτή BCM2835 της Broadcom με CPU τον μονοπύρηνο ARM1176JZ-F (ARMv6) στα 700MHz, μνήμη RAM 512MB, τέσσερις θύρες USB2.0, θύρα stereo audio 3,5", 40-pin GPIO (General Purpose Input-Output), θύρα HDMI, θύρα Ethernet 100MBps, DSI και CSI cable υποδοχές (Display Serial Interface και Camera Serial Interface), υποδοχή microSD, υποστήριξη Full HD (1080p). Η τελευταία έκδοση, Raspberry Pi 2, αναβαθμίστηκε με μικροελεγκτή τον Broadcom BCM2836 (τετραπύρηνος ARM Cortex-A7 (ARMv7) στα 900MHz και μνήμη RAM 1GB), ενώ διαθέτει και όλα τα περιφερειακά της προηγούμενης έκδοσης (Raspberry Pi B+).

Ανάπτυξη Web Interface για απομακρυσμένο έλεγχο
με τη χρήση του μικροελεγκτή RaspberryPi

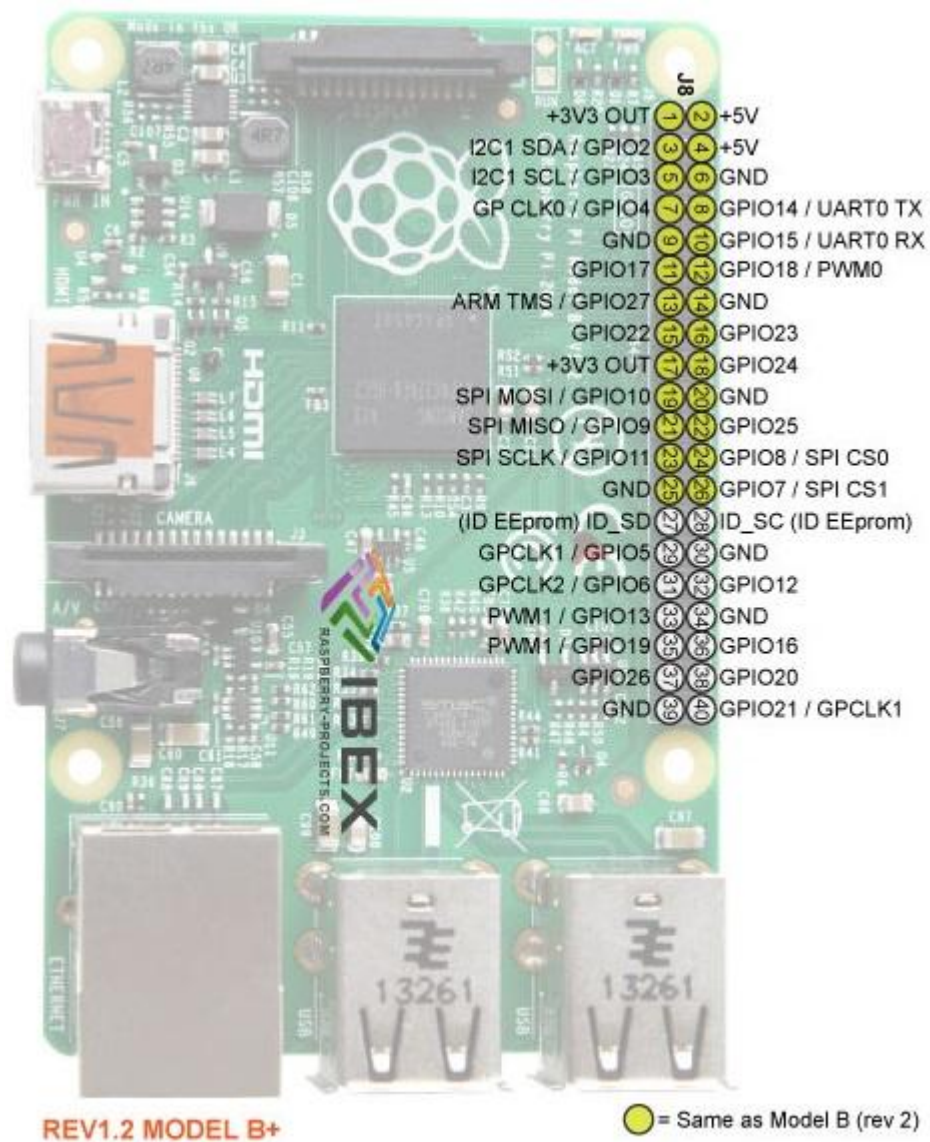


aspberry Pi B+

R

Το Raspberry Pi B+ (ομοίως και το Raspberry Pi 2) έχουν 40 pins γενικής χρήσης, που λειτουργούν στα 3,3V, εκτός από τα pins των 5V. Επίσης, αν εφαρμόσουμε τάση 5V σε οποιοδήποτε ακροδέκτη που λειτουργεί στα 3,3V, πιθανώς να καταστρέψουμε τη συσκευή. Η μέγιστη επιτρεπτή ροή ρεύματος από τους ακροδέκτες των 3.3V είναι 50mA συνολικά και 16mA από τον καθένα. Ορισμένοι ακροδέκτες μπορούν να επιτελούν και δευτερεύοντες λειτουργίες, όπως για παράδειγμα το pin 12 μπορεί να χρησιμοποιηθεί ως έξοδος PWM (Pulse Width Modulation).

Ανάπτυξη Web Interface για απομακρυσμένο έλεγχο με τη χρήση του μικροελεγκτή RaspberryPi



Raspberry Pi B+ J8 Header

2.2.2 Λειτουργικό σύστημα για το Raspberry Pi

Το Raspberry Pi μπορεί να χρησιμοποιήσει κάποιο linux-based λειτουργικό σύστημα. Το σύνθηες λειτουργικό σύστημα που χρησιμοποιεί είναι το Raspbian, το οποίο είναι ειδικά σχεδιασμένο για να χρησιμοποιηθεί με το Raspberry Pi. Το λειτουργικό σύστημα αποθηκεύεται στη κάρτα microSD. Πέραν από το Raspbian υπάρχουν και κάποια ακόμη λειτουργικά συστήματα που μπορούν να χρησιμοποιηθούν με το Raspberry Pi. Ο install manager για το Raspberry Pi είναι ο NOOBS (New Out Of Box Software), και στην

εκτέλεση του δίνει την δυνατότητα επιλογής ανάμεσα σε αυτά τα λειτουργικά συστήματα. Επίσης, λόγω της χρήσης εξωτερικής κάρτας μνήμης σαν το κύριο μέσο αποθήκευσης, μπορούμε να έχουμε σε διαφορετικές κάρτες μνήμης διαφορετικά λειτουργικά συστήματα. Επιπροσθέτως, έχουμε τη δυνατότητα να εγκαταστήσουμε σε μία κάρτα μνήμης ένα συγκεκριμένο λειτουργικό σύστημα και μία συγκεκριμένη εφαρμογή και λειτουργία, και απλά αλλάζοντας κάρτες μνήμης το Raspberry Pi να εκτελεί κάθε φορά την εκάστοτε λειτουργία αναφορικά με την κάρτα μνήμης. Τέλος να αναφερθεί ότι μπορεί να γίνει χρήση του Raspberry Pi χωρίς κάποιο λειτουργικό σύστημα, κάνοντας χρήση κάποιου compiler για ARM επεξεργαστές και τοποθετώντας το δυαδικό αρχείο με το πρόγραμμα στη κάρτα microSD.

2.2.3 Πλεονεκτήματα και Μειονεκτήματα του Raspberry

Τα βασικά πλεονεκτήματα του Raspberry Pi είναι τα εξής:

- Πολύ μικρό κόστος (περίπου 40€)
- Μέγεθος αντίστοιχο μια πιστωτικής κάρτας (85.6x56mm)
- Χαμηλή κατανάλωση ενέργειας
- Ευκολία στη χρήση
- Αναπαραγωγή Video Full HD (1080p)
- Επέκταση λειτουργιών μέσω των ακροδεκτών
- Δυνατότητα αύξησης συχνότητας μικροεπεξεργαστή (overclocking)
- Συμβατότητα με το λειτουργικό σύστημα της Microsoft, Windows 10 (μόνο το Raspberry Pi 2)

Το βασικότερο μειονέκτημα του Raspberry Pi είναι η μη δυνατότητα της επέκτασης, και περαιτέρω αύξησης, της μνήμης RAM. Επίσης, λόγω του ότι οι ακροδέκτες του είναι σε τάση 3,3V, ο μικροελεγκτής δεν είναι συμβατός με αισθητήρες και εξαρτήματα των 5V.

2.3 Web Server

2.3.1 Τι είναι ένας Server

Ο Server ή αλλιώς εξυπηρετητής ή διακομιστής, στην πιο απλή του μορφή είναι ένας ηλεκτρονικός υπολογιστής που τρέχει κατάλληλο λογισμικό ώστε να εξυπηρετεί τους χρήστες που συνδέονται με αυτόν για κάποιο σκοπό. Ανάλογα με τον σκοπό, ο Server τρέχει και τις κατάλληλες υπηρεσίες και έχει και κατάλληλη ονομασία. Για παράδειγμα αν ο Server εξυπηρετεί ιστοσελίδες, ονομάζεται Web Server και τρέχει λογισμικό κατάλληλο για να εξυπηρετεί τα αρχεία με τα οποία είναι κατασκευασμένη μία ιστοσελίδα ενώ αν ο Server εξυπηρετεί e-mail λογαριασμούς ώστε οι χρήστες να έχουν το e-mail τους σε αυτόν, τότε θα είναι Mail Server και θα τρέχει κατάλληλο λογισμικό για αποστολή και λήψη e-mail μηνυμάτων.

Υπόψιν ότι μέσα σε έναν Server μπορεί να υπάρχουν στημένοι πολλών ειδών τέτοιοι, πχ Mail Server, Web Server, Ftp Server, κλπ και το μηχάνημα αυτό να εκπληρώνει πολλαπλούς ρόλους ταυτόχρονα.

2.3.2 Κατηγορίες Server

Οι βασικότερες κατηγορίες Server που χρησιμοποιούνται είναι οι εξής:

- **Web Server:** Στον Web Server φιλοξενούνται ιστοσελίδες και όταν κάποιος καλέσει μία τέτοια (γράφοντας το url της), τότε ο Web Server θα αναλάβει να την φορτώσει στον browser του επισκέπτη.
- **Mail Server:** Ο Mail Server εξυπηρετεί αποκλειστικά υπηρεσίες mail. Ο Mail Server εξυπηρετεί και την εισερχόμενη και την εξερχόμενη αλληλογραφία και διαχειρίζεται γενικά όλα τα σχετικά με το e-mail μας.
- **Proxy Server:** Ο Proxy Server μεσολαβεί μεταξύ ενός χρήστη και του Internet. Έχει στόχο να βελτιώσει την ταχύτητα πλοήγησης στο Internet και παράλληλα να μειώσει την κίνηση του δικτύου προς το Internet.
- **Application Server:** Λειτουργεί σαν μεσάζοντας, καλύπτοντας μεγάλο όγκο υπολογισμών, μεταξύ μίας βάσης δεδομένων και του τελικού χρήστη.

- IRC (Internet Relay Chat) Server: Επιτρέπει την άμεση ανταλλαγή μηνυμάτων κειμένου.
- Telnet Server: Επιτρέπει στους χρήστες να συνδεθούν σε ένα κεντρικό υπολογιστή και να δουλέψουν σε αυτόν απομακρυσμένα, σαν να ήταν εκεί.
- FTP Server: Καθιστά δυνατή την ασφαλή ανταλλαγή αρχείων χρησιμοποιώντας το πρωτόκολλο FTP (File Transfer Protocol).
- Audio/Video Server: Βοηθάει στη μετάδοση συνεχούς ροής βίντεο και ήχου, παρέχοντας δυνατότητες πολυμέσων σε ιστιότοπους.

2.3.3 Χαρακτηριστικά ενός Server

Όπως είπαμε, στην πιο απλή του μορφή ένας Server είναι ένας ηλεκτρονικός υπολογιστής ο οποίος τρέχει κατάλληλο λογισμικό. Στην πράξη βέβαια, δεν είναι απλοί υπολογιστές αλλά είναι ηλεκτρονικοί υπολογιστές αυξημένων δυνατοτήτων. Ακριβώς επειδή η δουλειά ενός server είναι να εξυπηρετεί τους χρήστες που τον επισκέπτονται, ανάλογα με το πόσο μεγάλη είναι αυτή η επισκεψιμότητα αλλά και το τι εργασίες πρέπει να εκτελέσει ο Server για κάθε επισκέπτη που συνδέεται με αυτόν ώστε να του δώσει αυτό που θέλει και μάλιστα αρκετά γρήγορα, οι Servers έχουν αυξημένες δυνατότητες σε:

- χωρητικότητα και ταχύτητα δίσκων ενός Server αφού αυτοί αναλαμβάνουν την αποθήκευση των ηλεκτρονικών αρχείων τα οποία διαμοιράζει ο Server στους χρήστες του. Η ταχύτητα όμως των δίσκων παίζει και αυτή καθοριστικό ρόλο στην ταχύτητα απόκρισης ενός Server, μιας και τα ηλεκτρονικά αρχεία που σερβίρει ο διακομιστής και είναι αποθηκευμένα στους δίσκους θα παραδίδονται αργά στους επισκέπτες-χρήστες του. Αν μάλιστα τον Server τον επισκέπτονται πολλοί χρήστες ταυτόχρονα και οι δίσκοι του Server είναι αργοί, τότε οι χρήστες θα διαπιστώνουν πολύ αργή ταχύτητα φόρτωσης των σελίδων ή στην χειρότερη περίπτωση και πλήρη αδυναμία εξυπηρέτησης των σελίδων. Οι ταχύτεροι δίσκοι είναι οι δίσκοι τεχνολογίας SSD. Ένας καλής ποιότητας SSD έχει μεγαλύτερη αξιοπιστία από έναν δίσκο

κλασσικής τεχνολογίας και κατά πολύ μεγαλύτερη ταχύτητα (~100 φορές μεγαλύτερη ταχύτητα).

- μνήμη RAM μιας και οτιδήποτε σερβίρεται στους επισκέπτες φορτώνεται από τους δίσκους στην μνήμη RAM και μετά καταλήγει στον τελικό χρήστη. Αν η μνήμη RAM είναι αρκετά μεγάλη, τότε μέρος των δεδομένων του Server μπορούν να αποθηκεύονται στην RAM, ώστε να σερβίρονται κατευθείαν από εκεί στους χρήστες, μιας και είναι πολύ πιο γρήγορο μέσο απ' ό τι είναι οι δίσκοι.
- επεξεργαστική ισχύ (CPU) αφού παίζει συνολικά ρόλο στην ταχύτητα εξυπηρέτησης του Server μιας και είναι το κεντρικό σημείο από το οποίο περνούν και επεξεργάζονται όλες οι πληροφορίες για τον διαμοιρασμό τους στους χρήστες. Σε περιβάλλοντα Servers που εξυπηρετούνται ταυτόχρονα πολλοί χρήστες, επεξεργαστές με πολλούς πυρήνες συνιστώνται μιας και βοηθούν στην ταυτόχρονη επεξεργασία πληροφοριών.
- χωρητικότητα δικτύου μιας και αυτή θα πρέπει να επαρκεί ώστε να εξυπηρετήσει με την ταχύτητα που θέλουμε, τον αριθμό των ταυτόχρονων χρηστών που έχουμε υπολογίσει ότι θα συνδεθούν πάνω του στην ώρα αιχμής. Αν για παράδειγμα, σε ώρα αιχμής συνδέονται πάνω σε έναν Server 1000 άτομα και θέλουμε να τα εξυπηρετήσουμε με ταχύτητα 10Mbps το καθένα, τότε ο Server θα πρέπει να έχει χωρητικότητα δικτύου 10Gbps.
- αξιοπιστία υλικών καθώς η κατασκευή ενός Server πρέπει να γίνει με υλικά καλής ποιότητας και αξιοπιστίας ώστε ο Server να μην αντιμετωπίσει προβλήματα και σταματήσει η λειτουργία του.

2.4 Πρωτόκολλο Μεταφοράς Υπερκειμένου (HyperText Transfer Protocol - HTTP)

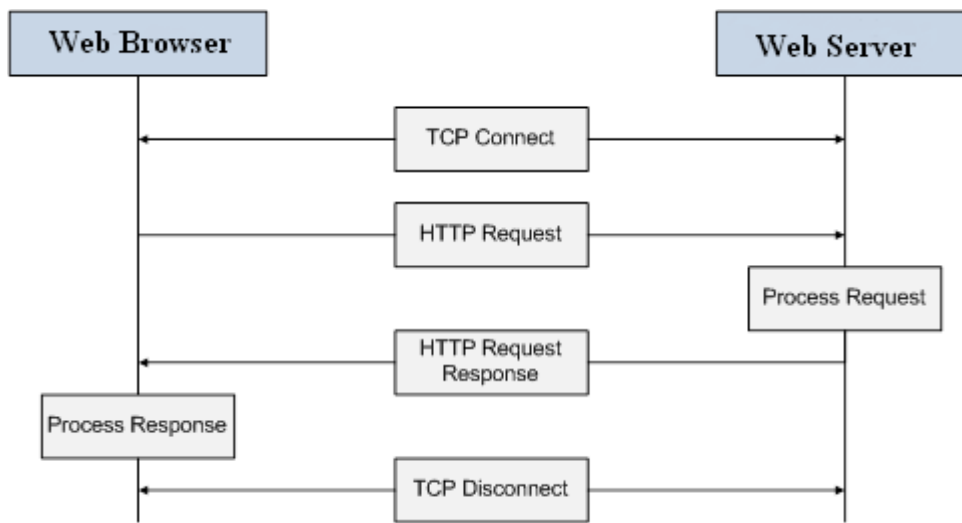
2.4.1 Τι είναι το HTTP

Το HTTP αποτελεί συντομογραφία της φράσης: HyperText Transfer Protocol (Πρωτόκολλο Μεταφοράς Υπερκειμένου). Είναι ένα σύνολο κανόνων, που καθορίζει τον τρόπο με τον οποίο θα γίνει η μεταφορά του υπερκειμένου (hypertext) μεταξύ δύο ή περισσότερων υπολογιστών. Το πρωτόκολλο HTTP είναι το πιο συνηθισμένο στον ηλεκτρονικό χώρο του Διαδικτύου και αποτελεί ένα πρωτόκολλο του επιπέδου εφαρμογών στα δίκτυα υπολογιστών. Είναι ένα γενικό, αντικειμενοστραφές πρωτόκολλο που μπορεί να χρησιμοποιηθεί σε ένα πλήθος εφαρμογών, για παράδειγμα σε Servers. Όλοι οι φυλλομετρητές (Web Browsers) το θεωρούν δεδομένο και το χρησιμοποιούν ακόμα και σε περιπτώσεις που ο χρήστης δεν καθορίσει ποιο πρωτόκολλο θέλει να χρησιμοποιήσει. Δηλαδή, αν ο χρήστης δεν γράψει το `http://` πριν το url της Ιστοσελίδας που θέλει, αλλά μονάχα το url, δεν θα υπάρξει πρόβλημα, καθώς όλοι οι Web Browsers θεωρούν σαν προεπιλογή το HTTP και όχι κάποιο άλλο πρωτόκολλο. Το απλό πρωτόκολλο HTTP δεν εγγυάται καμία ασφάλεια. Η έκδοση που χρησιμοποιούμε είναι η HTTP/1.1.

2.4.2 Πως λειτουργεί το HTTP

Το βασικότερο και πιο σημαντικό ίσως χαρακτηριστικό του πρωτοκόλλου αυτού είναι ότι επιτρέπει στα διάφορα συστήματα μετάδοσης δεδομένων να υφίστανται ανεξάρτητα από τα δεδομένα που αυτά μεταφέρουν. Πιο ειδικά αναφέρουμε ότι στηρίζεται στο μοντέλο πελάτη-διανομέα (client-server). Έτσι, λοιπόν, το ρόλο του διανομέα παίζει ο Web Server και του πελάτη ο Web Browser. Όταν ο χρήστης ζητά μία Ιστοσελίδα, ο Web Browser στέλνει ένα μήνυμα HTTP αίτησης (HTTP request), για τα διάφορα αντικείμενα της σελίδας, στον εξυπηρετητή (Web Server). Ο εξυπηρετητής όταν λάβει το μήνυμα αυτό ανταποκρίνεται με μηνύματα HTTP απόκρισης (HTTP response), στα οποία περιέχονται τα αιτούμενα αντικείμενα.

Το HTTP χρησιμοποιεί το πρωτόκολλο TCP (Transmission Control Protocol) για μεταφορά. Αφού ο πελάτης (Web Browser) εγκαταστήσει μία σύνδεση TCP με τον εξυπηρετητή (Web Server) αρχίζει την αποστολή μηνυμάτων αιτήσεων προς αυτόν και τη λήψη μηνυμάτων αποκρίσεων από αυτόν.



Πως λειτουργεί το HTTP

Λόγω της χρήσης του TCP το HTTP δεν χρειάζεται να ασχοληθεί καθόλου με τη μεταφορά των δεδομένων. Το μόνο που πρέπει να κάνει είναι να στείλει τις αιτήσεις μέσω της TCP σύνδεσης και να περιμένει τις αποκρίσεις. Το TCP εγγυάται την αξιόπιστη μεταφορά των δεδομένων καθώς και τον έλεγχο της συμφόρησης. Οι εξυπηρετητές του HTTP δεν κρατάνε καθόλου στοιχεία για την κατάσταση του πελάτη. Επομένως, αν ένας πελάτης στείλει μία αίτηση για ένα αρχείο δύο φορές, ο εξυπηρετητής θα του στείλει το αρχείο αυτό δύο φορές. Το HTTP μπορεί να χρησιμοποιήσει τόσο μόνιμες (persistent) όσο και μη μόνιμες (non-persistent) συνδέσεις.

2.4.2.1 Μη μόνιμες συνδέσεις

Έστω ότι θέλουμε να εμφανίσουμε στην οθόνη μία ιστοσελίδα που αποτελείται από το βασικό HTML αρχείο και άλλα τέσσερα αρχεία εικόνων. Έστω, επίσης, ότι το URL του βασικού HTML αρχείου είναι το `www.ntua.gr/index.htm`. Τότε πρέπει να ακολουθηθούν τα επόμενα βήματα για την λήψη της:

1. Ο HTTP πελάτης εγκαθιστά μία σύνδεση TCP στην θύρα 80 του εξυπηρετητή HTTP. Η θύρα 80 είναι η προκαθορισμένη θύρα στην οποία ο εξυπηρετητής HTTP περιμένει για αφίξεις αιτήσεων HTTP από διάφορους πελάτες HTTP.
2. Στη συνέχεια ο πελάτης HTTP στέλνει ένα μήνυμα HTTP αίτησης ζητώντας το αρχείο `index.htm`.
3. Ο εξυπηρετητής στέλνει το αρχείο `index.htm` στον πελάτη μέσω ενός HTTP μηνύματος απόκρισης.
4. Ο HTTP εξυπηρετητής κλείνει την TCP σύνδεση (στην πραγματικότητα η TCP σύνδεση κλείνει όταν επιβεβαιωθεί η λήψη του αρχείου από τον πελάτη).
5. Ο Web Browser λαμβάνει το αρχείο `index.htm`, κλείνει τη σύνδεση, και απεικονίζει τα περιεχόμενα του στην οθόνη του χρήστη. Επίσης ο Web Browser αναγνωρίζει και τα URL των τεσσάρων αναφερομένων εικόνων της Ιστοσελίδας.
6. Τα τέσσερα πρώτα βήματα επαναλαμβάνονται για καθένα από τα αναφερόμενα αντικείμενα (στην περίπτωσή μας εικόνες).

Από το παραπάνω παράδειγμα γίνεται φανερό γιατί ο τύπος αυτός της σύνδεσης ονομάζεται μη μόνιμος – μετά από την μεταφορά ενός αντικειμένου της Ιστοσελίδας η TCP σύνδεση τερματίζεται και για κάθε άλλο αντικείμενο της Ιστοσελίδας πρέπει να εγκατασταθεί μία καινούργια TCP σύνδεση. Στο παραπάνω παράδειγμα για την μεταφορά

της Ιστοσελίδας χρειάζεται η εγκατάσταση συνολικά 5 TCP συνδέσεων. Παρά το γεγονός ότι πρέπει να εγκατασταθούν τέσσερις διαφορετικές TCP συνδέσεις για τη μεταφορά των τεσσάρων εικόνων, οι σύγχρονοι Web Browser υποστηρίζουν την παράλληλη μεταφορά (parallel transfer) των διαφόρων αντικειμένων. Σε αυτή την περίπτωση οι εικόνες μπορεί να μεταφερθούν παράλληλα με αποτέλεσμα τη μείωση του συνολικού χρόνου που χρειάζεται για τη ολόκληρης της σελίδας στον Web Browser. Όταν εγκαθίστανται παράλληλες TCP συνδέσεις το συνολικό εύρος ζώνης (bandwidth) μοιράζεται ανάμεσα στις παράλληλες συνδέσεις.

2.4.2.2 Μόνιμες συνδέσεις

Πριν να περιγράψουμε τις μόνιμες συνδέσεις θα κάνουμε ένα χοντρικό υπολογισμό του χρόνου που χρειάζεται για τη μεταφορά ολόκληρης της Ιστοσελίδας. Για τον σκοπό αυτό ορίζουμε ως RTT (Round Trip Time) το χρόνο που χρειάζεται ένα πακέτο να μεταφερθεί από τον εξυπηρετητή (Web Server) στον Web Browser και μετά πάλι πίσω. Ο χρόνος αυτός περιλαμβάνει την καθυστέρηση διάδοσης (propagation delay), την καθυστέρηση του πακέτου στις ουρές των δρομολογητών (queuing delay), καθώς και την καθυστέρηση επεξεργασίας του πακέτου (packet processing delay). Ας δούμε τι συμβαίνει όταν ο χρήστης ζητά μία Ιστοσελίδα. Αρχικά ο Web Browser εγκαθιστά μία TCP σύνδεση με τον εξυπηρετητή (Web Server). Για το σκοπό αυτό λαμβάνει χώρα μία τριμερής χειραψία (three-way handshake): πρώτα ο πελάτης στέλνει ένα μικρό TCP segment στον εξυπηρετητή, στη συνέχεια ο εξυπηρετητής επιβεβαιώνει τη λήψη του παραπάνω μηνύματος με ένα νέο TCP segment που στέλνει πίσω στον πελάτη και τέλος ο πελάτης επιβεβαιώνει τη λήψη του τελευταίου αυτού

μηνύματος. Η τελευταία επιβεβαίωση μπορεί να είναι εμβόλιμη (piggy backing) στην HTTP αίτηση που κάνει ο πελάτης. Όταν η αίτηση αυτή φτάσει στον εξυπηρετητή αυτός αποκρίνεται με μία απόκριση HTTP που περιλαμβάνει το ζητούμενο αρχείο. Με μια πρώτη προσέγγιση ο συνολικός χρόνος είναι 2RTT, συν το χρόνο μετάδοσης του αρχείου από τον εξυπηρετητή, για κάθε αρχείο.

Από τα παραπάνω γίνεται φανερό ότι οι μη μόνιμες συνδέσεις παρουσιάζουν κάποια σημαντικά προβλήματα: πρώτον, για κάθε αντικείμενο της Ιστοσελίδας πρέπει να εγκατασταθεί μία ξεχωριστή TCP σύνδεση με αποτέλεσμα να χρειάζεται κάποιος χώρος αποθήκευσης για την κατάστασή της τόσο στην πλευρά του πελάτη όσο και στην πλευρά του εξυπηρετητή. Το πρόβλημα αυτό παρουσιάζεται κυρίως στους εξυπηρετητές που συνήθως πρέπει να είναι ικανοί να εξυπηρετήσουν πολλούς πελάτες ταυτόχρονα. Επιπλέον, το κάθε αντικείμενο της Ιστοσελίδας δέχεται καθυστέρηση 2RTT, ενώ το TCP έχει το χαρακτηριστικό της αργής εκκίνησης (slow start) που προσδίδει ακόμη μεγαλύτερη καθυστέρηση στη μεταφορά των αντικειμένων.

Αν και τα μειονεκτήματα αυτά μπορεί να περιοριστούν αν γίνει χρήση παράλληλων συνδέσεων, το HTTP υποστηρίζει μία πιο αποδοτική τεχνική, τις μόνιμες συνδέσεις. Στις μόνιμες συνδέσεις ο εξυπηρετητής TCP δεν τερματίζει τη σύνδεση αμέσως μετά την αποστολή της απόκρισης. Αντίθετα, την τερματίζει μετά από κάποιο συγκεκριμένο χρονικό διάστημα κατά το οποίο η σύνδεση είναι ανενεργή (δηλαδή δεν γίνεται μεταφορά δεδομένων). Μέσα στο χρονικό διάστημα αυτό ο Web Browser μπορεί να ζητήσει την αποστολή και άλλων αντικειμένων από την ίδια Ιστοσελίδα αλλά και την αποστολή και άλλων Ιστοσελίδων από τον ίδιο εξυπηρετητή Ιστού.

2.4.3 Πρωτόκολλο ασφαλούς μεταφοράς υπερκειμένου (HyperText Transfer Protocol Secure - HTTPS)

Το HTTPS δεν είναι ξεχωριστό πρωτόκολλο αλλά αναφέρεται στο συνδυασμό του απλού HTTP πρωτοκόλλου και των δυνατοτήτων κρυπτογράφησης που παρέχει το πρωτόκολλο Secure Sockets Layer (SSL) ή το πρωτόκολλο Transport Layer Security (TLS). Περισσότερο χρησιμοποιείται το πρωτόκολλο SSL, μιας και είναι ιδιαίτερα κατάλληλο για το HTTP, δεδομένου ότι μπορεί να προσφέρει προστασία, ακόμη και αν η μία πλευρά μονάχα της επικοινωνίας είναι επικυρωμένη. Η κρυπτογράφηση που χρησιμοποιείται διασφαλίζει ότι τα κρυπτογραφημένα δεδομένα δε μπορούν να υποκλαπούν από άλλους κακόβουλους χρήστες. Για να χρησιμοποιηθεί το HTTPS σε κάποιο Web Server θα πρέπει ο διαχειριστής του να εκδώσει ένα πιστοποιητικό δημοσίου κλειδιού. Στην συνέχεια το πιστοποιητικό αυτό θα πρέπει να υπογραφεί από μία αρχή πιστοποίησης (certificate authority), η οποία πιστοποιεί ότι ο εκδότης του πιστοποιητικού είναι νομότυπος και ότι το πιστοποιητικό είναι έγκυρο. Με τον τρόπο αυτό οι χρήστες μπορούν να δουν την υπογραφή της αρχής πιστοποίησης και να βεβαιωθούν ότι το πιστοποιητικό είναι έγκυρο και ότι κανένας κακόβουλος χρήστης δεν το έχει πλαστογραφήσει. Το HTTPS χρησιμοποιείται κυρίως όταν απαιτείται μεταφορά ευαίσθητων προσωπικών δεδομένων. Το επίπεδο προστασίας των δεδομένων εξαρτάται από το πόσο ισχυροί είναι οι αλγόριθμοι κρυπτογράφησης που χρησιμοποιούνται.

2.4.4 Διαφορές HTTP και HTTPS

Υπάρχουν κάποιες βασικές διαφορές μεταξύ των πρωτοκόλλων HTTP και HTTPS αρχίζοντας με την προεπιλεγμένη θύρα, η οποία είναι 80 για HTTP και 443 για HTTPS. Επίσης, στο HTTPS πρωτόκολλο, οι διευθύνσεις ιστοτόπων (url) ξεκινούν με https:// αντί με http://. Όταν χρησιμοποιούμε μια σύνδεση HTTPS, ο διακομιστής απαντά με την αρχική σύνδεση, προσφέροντας μια λίστα με τις μεθόδους

κρυπτογράφησης που υποστηρίζει. Σε απάντηση, ο πελάτης επιλέγει μια μέθοδο σύνδεσης, και τα πιστοποιητικά ανταλλαγή client και server για την επικύρωση των ταυτοτήτων τους. Αφού γίνει αυτό, αμφότερα τα μέρη ανταλλάσσουν τις κρυπτογραφημένες πληροφορίες μετά από την εξασφάλιση ότι και τα δύο χρησιμοποιούν το ίδιο κλειδί, και η σύνδεση είναι κλειστή. Προκειμένου να φιλοξενήσει συνδέσεις HTTPS, ένας διακομιστής πρέπει να έχει ένα πιστοποιητικό με δημόσιο κλειδί, το οποίο ενσωματώνει τις βασικές πληροφορίες σε επαλήθευση της ταυτότητας του ιδιοκτήτη κλειδί.

2.5 Η γλώσσα προγραμματισμού HTML

Η HTML (HyperText Markup Language) είναι η κύρια γλώσσα σήμανσης για τις ιστοσελίδες, και τα στοιχεία της είναι τα βασικά δομικά στοιχεία των ιστοσελίδων. Κάθε σελίδα που εμφανίζεται στο Internet είναι ένα αρχείο γραμμένο με τη γλώσσα HTML, που περιλαμβάνει το κείμενο της σελίδας, τη δομή της και τους συνδέσμους προς άλλα έγγραφα, εικόνες ή άλλα μέσα. Η HTML γράφεται υπό μορφή στοιχείων HTML τα οποία αποτελούνται ετικέτες (tags), οι οποίες συνήθως λειτουργούν ανά ζεύγη με την πρώτη να ονομάζεται ετικέτα έναρξης (<html>, <body>, <head> κ.α.) και τη δεύτερη ετικέτα λήξης (</html>, </body>, </head> κ.α.) (ή σε άλλες περιπτώσεις ετικέτα ανοίγματος και ετικέτα κλεισίματος αντίστοιχα). Ανάμεσα στις ετικέτες, οι σχεδιαστές ιστοσελίδων μπορούν να τοποθετήσουν κείμενο, πίνακες, εικόνες κλπ.

Ο σκοπός ενός Web Browser είναι να διαβάσει τα έγγραφα HTML και τα συνθέτει σε σελίδες που μπορεί κανείς να διαβάσει ή να ακούσει. Ο Web Browser δεν εμφανίζει τις ετικέτες HTML, αλλά τις χρησιμοποιεί για να ερμηνεύσει το περιεχόμενο της σελίδας.

Τα στοιχεία της HTML χρησιμοποιούνται για να κτίσουν όλους τους ιστιότοπους. Η HTML επιτρέπει την ενσωμάτωση εικόνων και άλλων αντικειμένων μέσα στη σελίδα, και μπορεί να χρησιμοποιηθεί για να

εμφανίσει διαδραστικές φόρμες. Παρέχει τις μεθόδους δημιουργίας δομημένων εγγράφων (δηλαδή εγγράφων που αποτελούνται από το περιεχόμενο που μεταφέρουν και από τον κώδικα μορφοποίησης του περιεχομένου) καθορίζοντας δομικά σημαντικά στοιχεία για το κείμενο, όπως κεφαλίδες, παραγράφους, λίστες, συνδέσμους, παραθέσεις και άλλα. Μπορούν επίσης να ενσωματώνονται σενάρια εντολών σε γλώσσες όπως η JavaScript, τα οποία επηρεάζουν τη συμπεριφορά των ιστοσελίδων HTML, ενώ με την ενσωμάτωση κώδικα σε PHP μπορεί να ανταλλάσει πληροφορίες όπως για παράδειγμα με βάσεις δεδομένων της MySQL.

2.6 Η γλώσσα προγραμματισμού Javascript

2.6.1 Τι είναι η Javascript

Η Javascript είναι μια γλώσσα σεναρίου (script language) που χρησιμοποιείται για να δώσει διαδραστικότητα μεταξύ χρήστη και HTML εγγράφου. Αυτή η γλώσσα είναι client side που σημαίνει ότι εκτελείται και παράγεται το αποτέλεσμα στον Web Browser του client - του επισκέπτη δηλαδή. Δεν έχει σχέση με την Java. Δεν είναι κατάλληλη στο να φτιάξουμε εξ' ολοκλήρου μία δυναμική ιστοσελίδα με αυτήν, απλά λειτουργεί "συμπληρωματικά". Επιτρέπει δηλαδή στον επισκέπτη να κάνει κάποια πράγματα δυναμικά σε μία ιστοσελίδα, σε μικρό βαθμό και άμεσα, χωρίς να επικοινωνήσει το σύστημα με τον Web Server για να εκτελέσει το αίτημά του. Ο κώδικας JavaScript συνήθως ενσωματώνεται στον HTML κώδικα. Με απλά λόγια, πχ μπορεί να φτιάξουμε ένα script σε Javascript το οποίο να κάνει το εξής: έστω σε μία ιστοσελίδα έχουμε μία φόρμα. Θέλουμε όμως να μην επιτρέπουμε στο πεδίο με το e-mail να γράψει πάνω από 50 χαρακτήρες. Αντίστοιχα θέλουμε πριν αποσταλεί η φόρμα, να ελέγξει αν το e-mail που πληκτρολόγησε ο χρήστης είναι σωστό, δηλαδή της μορφής πχ

kati@kati.com. Έτσι μόλις πατήσουμε το κουμπί αποστολή, η Javascript εφαρμογή θα ελέγξει αν τα ηλεκτρολόγησε ο χρήστης σωστά και είτε θα τον αφήσει να συνεχίσει, είτε όχι, βγάζοντάς του πχ ένα μήνυμα: το e-mail που ηλεκτρολογήσατε δεν είναι σωστό.

Επίσης, με τη Javascript μπορούμε να κάνουμε αρκετά πράγματα ακόμα μέσα σε μία ιστοσελίδα: μπορούμε να εκτελέσουμε κάποια πράγματα όταν συμβαίνει ένα γεγονός, για παράδειγμα όταν ο χρήστης κάνει κλικ σε ένα HTML στοιχείο, να εκτελείται κάποιο script και λαμβάνουμε τα αντίστοιχα αποτελέσματα, μπορούμε να διαβάσουμε και να αλλάξουμε τα περιεχόμενα ενός HTML στοιχείου, μπορούμε να την χρησιμοποιήσουμε για να επικυρώσουμε δεδομένα μιας φόρμας (validate) προτού αυτά να υποβληθούν στον Web Server (γλυτώνουμε και τον Web Server από επιπλέον έλεγχο και μεγαλύτερο δικτυακό φόρτο), μπορούμε να εντοπίσουμε τον Web Browser του επισκέπτη και ανάλογα με τον Web Browser να φορτώσουμε την αντίστοιχη σελίδα που είναι φτιαγμένη για αυτόν, μπορούμε να δημιουργήσουμε cookies (να αποθηκεύουμε και να λαμβάνουμε πληροφορίες στον υπολογιστή του επισκέπτη), ακόμα και έγγραφα που μπορούν να αυτοενημερωθούν.

Η Javascript είναι case sensitive ,δηλαδή να προσέχουμε τον τρόπο που γράφουμε, δημιουργούμε ή καλούμε μεταβλητές, αντικείμενα ή συναρτήσεις. Για παράδειγμα η συνάρτηση function() και η συνάρτηση Function() είναι δύο εντελώς διαφορετικές συναρτήσεις.

2.6.2 Javascript και Java

Η Javascript και η Java είναι δύο εντελώς διαφορετικές γλώσσες. Η Javascript δεν θα πρέπει να συγχέεται με τη Java, που είναι διαφορετική γλώσσα προγραμματισμού και με διαφορετικές εφαρμογές. Η χρήση της λέξης “Java” στο όνομα της γλώσσας έχει περισσότερη σχέση με το προφίλ του προϊόντος και λιγότερο με κάποια πιθανή συμβατότητα ή άλλη στενή σχέση με τη Java. Ρόλο σε αυτήν τη σύγχυση έπαιξε και ότι η Java και η Javascript έχουν δεχτεί σημαντικές

επιρροές από τη γλώσσα προγραμματισμού C, ειδικά στο συντακτικό, ενώ και οι δύο είναι αντικειμενοστραφείς γλώσσες.

2.6.2.1 Διαφορές Javascript και Java

Μερικές διαφορές της Javascript και της Java είναι οι παρακάτω:

- Στη Javascript δε χρειαζόμαστε κάποιο ειδικό εργαλείο ή εφαρμογή για να γράψουμε τα scripts. Στη Java χρειαζόμαστε κάποια εφαρμογή, όπως το JDK (Java Developers Kit).
- Τα Javascripts που δημιουργούμε ενσωματώνονται στην ιστοσελίδα, ενώ στη Java μεταγλωττίζονται σε εκτελέσιμα αρχεία.
- Η Javascript τρέχει πολύ πιο γρήγορα από τη Java.
- Η Javascript είναι πιο κατάλληλη για απλές εφαρμογές, σε αντίθεση με τη Java που έχει περισσότερες δυνατότητες.
- Η JavaScript είναι πιο χαλαρή στη χρήση μεταβλητών σε σχέση με τη Java, η οποία έχει αυστηρούς κανόνες γύρω από τη χρήση μεταβλητών.
- Η JavaScript χρησιμοποιεί δυναμική σύνδεση, αφού οι αναφορές στα αντικείμενα ελέγχονται όταν το σενάριο εκτελείται. Στη Java οι αναφορές στα αντικείμενα πρέπει να υπάρχουν όταν μεταγλωττίζεται το πρόγραμμα. Χρησιμοποιεί, δηλαδή, στατική σύνδεση.
- Γενικά η Javascript είναι πιο εύκολη στη χρήση σε σχέση με τη Java.

2.6.3 Τρόποι εισαγωγής της Javascript

Υπάρχουν δύο τρόποι να εισάγουμε Javascript κώδικα μέσα σε ένα κώδικα ιστοσελίδας: εσωτερικά και εξωτερικά.

Για να εισάγουμε εσωτερικά στο αρχείο της ιστοσελίδας τον κώδικα της Javascript θα πρέπει να τον γράψουμε μέσα στις ετικέτες `<body>` ή `<head>`. Η μορφή που θα έχει θα είναι:

```
<script type="text/javascript">  
Εδώ ο κώδικας  
</script>
```

Ο κώδικας Javascript μπορεί να γραφτεί και ξεχωριστά από τον κώδικα της ιστοσελίδας. Τα εξωτερικά αρχεία Javascript έχουν κατάληξη `.js`. Συνήθως αυτά τα αρχεία Javascript περιέχουν κώδικα ο οποίος μπορεί να χρησιμοποιηθεί σε διαφορετικές ιστοσελίδες. Το μόνο που χρειάζεται να γράψουμε στον κώδικα της ιστοσελίδας είναι η παραπομπή για το αρχείο `.js`. Η μορφή είναι η εξής:

```
<script type="text/javascript" src="onoma_arxeiou.js"></script>
```

Τα εξωτερικά αρχεία `.js` δεν πρέπει να περιέχουν τις ετικέτες `<script>...</script>`.

2.7 Η γλώσσα προγραμματισμού PHP (Hypertext Preprocessor)

2.7.1 Γενικά για την PHP

Η PHP είναι μια γλώσσα προγραμματισμού για τη δημιουργία ιστοσελίδων με δυναμικό περιεχόμενο και μπορεί να εισαχθεί σε ιστοσελίδες HTML. Συγκεκριμένα, εκτελείται στη μεριά του Web

Server με σκοπό να παράγουμε με δυναμικό τρόπο τις HTML σελίδες (πχ δείξε μου τι καιρό κάνει τώρα), μιας και η HTML είναι μια στατική γλώσσα προγραμματισμού ιστοσελίδων (αναπαράσταση των δεδομένων μιας ιστοσελίδας).

Ένας κώδικας PHP περνά από επεξεργασία από ένα συμβατό διακομιστή του Παγκόσμιου Ιστού (π.χ. Apache), ώστε να παραχθεί σε πραγματικό χρόνο το τελικό περιεχόμενο, που θα σταλεί στο πρόγραμμα περιήγησης των επισκεπτών σε μορφή κώδικα HTML. Αυτό έχει ως αποτέλεσμα, να μην λειτουργήσει η ενσωμάτωση ενός αρχείου με κώδικα PHP σε μία ιστοσελίδα HTML αν δεν έχει την κατάλληλη επέκταση (π.χ. *.php, *.php4, *.phtml κ.ά.). Επίσης ακόμη κι όταν ένα αρχείο έχει την επέκταση .php, θα πρέπει ο Web Server να είναι ρυθμισμένος για να επεξεργάζεται και να μεταγλωττίζει τον κώδικα PHP σε HTML που καταλαβαίνει Web Browser του επισκέπτη. Όταν ο επισκέπτης ζητάει την εμφάνιση μιας ιστοσελίδας της μορφής PHP, τότε ο Web Server επεξεργάζεται εκείνη τη στιγμή τον κώδικα της ιστοσελίδας και δημιουργεί μια καινούργια σελίδα με καθαρό κώδικα HTML, ο οποίος θα διαφέρει ανάλογα με κάποιες παραμέτρους που έχουμε επιλέξει. Έτσι, είναι δυνατό διαφορετικοί χρήστες να λαμβάνουν τελείως διαφορετικά αποτελέσματα από την ίδια διεύθυνση ιστοσελίδας. Ο επισκέπτης βλέπει τον παραγόμενο HTML κώδικα αλλά δεν είναι σε θέση να γνωρίζει ποιος μπορεί να είναι ο αρχικός κώδικας PHP που δημιούργησε τον HTML κώδικα.

2.7.2 Τρόποι εισαγωγής της PHP

Οι τρόποι ενσωμάτωσης του PHP κώδικα είναι παρόμοιοι με τους αντίστοιχους του κώδικα Javascript που περιγράψαμε παραπάνω:
Ή γράφουμε τον κώδικα PHP μέσα στον αντίστοιχο HTML όπως:

```
<?php  
Ο PHP κώδικας εδώ  

```

Ή φτιάχνουμε βιβλιοθήκες σε ξεχωριστό αρχείο (πάντα με τη σωστή επέκταση έτσι ώστε να αναγνωρίζεται ως κώδικας PHP) και τις εισάγουμε στην HTML ιστοσελίδα:

```
<? Php  
include 'ονομα_vivliothikis.php';  
>
```

2.8 Το λειτουργικό σύστημα Raspbian

2.8.1 Γενικά για το Raspbian

Το Raspbian είναι ένα ελεύθερο λειτουργικό σύστημα βασισμένο στο Debian, ειδικά διαμορφωμένο για το Raspberry Pi. Περιέχει περισσότερα από 35.000 πακέτα και προ-μεταγλωττισμένες εφαρμογές για εύκολη εγκατάσταση και διαχείριση του Raspberry Pi. Αν και το Raspbian είναι ειδικά σχεδιασμένο και διαμορφωμένο για το Raspberry Pi, δεν προέρχεται από τους δημιουργούς του μικροελεγκτή, αλλά από μία ομάδα από προγραμματιστές που είναι οπαδοί του Raspberry Pi.

2.8.2 Τι είναι το Debian

Το Debian, αποτέλεσμα του Debian Project, είναι μια δημοφιλής διανομή Linux, δηλαδή ελεύθερο λογισμικό που αναπτύσσεται μέσω της συνεργασίας εθελοντών από όλο τον κόσμο. Βασίζεται στον πυρήνα Linux και στην ομάδα βασικών εργαλείων του εγχειρήματος GNU. Το Debian παρέχει πολύ περισσότερα από απλώς ένα λειτουργικό σύστημα. Έρχεται με συνολικά πάνω από 29.000 πακέτα προγραμμάτων, μεταγλωττισμένα και έτοιμα σε μια ωραία μορφή για εύκολη εγκατάσταση στο μηχανήμά σας.

2.8.3 Τι είναι το Linux

Το Linux είναι ένα ελεύθερο τύπου-Unix λειτουργικό σύστημα που αρχικά δημιουργήθηκε από τον Linus Torvalds και στη συνέχεια αναπτύχθηκε με τη βοήθεια προγραμματιστών από όλον τον κόσμο. Το Linux είναι μία ανεξάρτητη POSIX υλοποίηση και στα χαρακτηριστικά του συμπεριλαμβάνει πραγματικό multitasking, πραγματικό πολυχρηστικό περιβάλλον, virtual memory, shared libraries, demand loading, TCP/IP networking και πολλά άλλα χαρακτηριστικά που δικαιολογούν τον τίτλο "τύπου-Unix". Είναι κατασκευασμένο υπό την GPL άδεια, δηλαδή ο πηγαίος κώδικάς του είναι διαθέσιμος στον καθένα (opensource).

Το Linux δεν είναι ακριβώς ένα ενιαίο λειτουργικό σύστημα, όπως είναι τα Windows και το Mac OS. Είναι απλά ο πυρήνας (kernel), που σε συνδυασμό με τρίτες εφαρμογές δημιουργεί το πλήρες λειτουργικό σύστημα. Αυτός ο συνδυασμός ονομάζεται “διανομή Linux” (όπως το Debian), και υπάρχουν πολλές διαφορετικές διανομές που αναπτύσσονται από διαφορετικές εταιρείες ενώ απευθύνονται σε διαφορετικούς χρήστες και καλύπτουν διαφορετικές ανάγκες.

2.8.4 Πλεονεκτήματα του Linux

Οι βασικότεροι λόγοι για να χρησιμοποιήσουμε μια διανομή Linux είναι οι εξής:

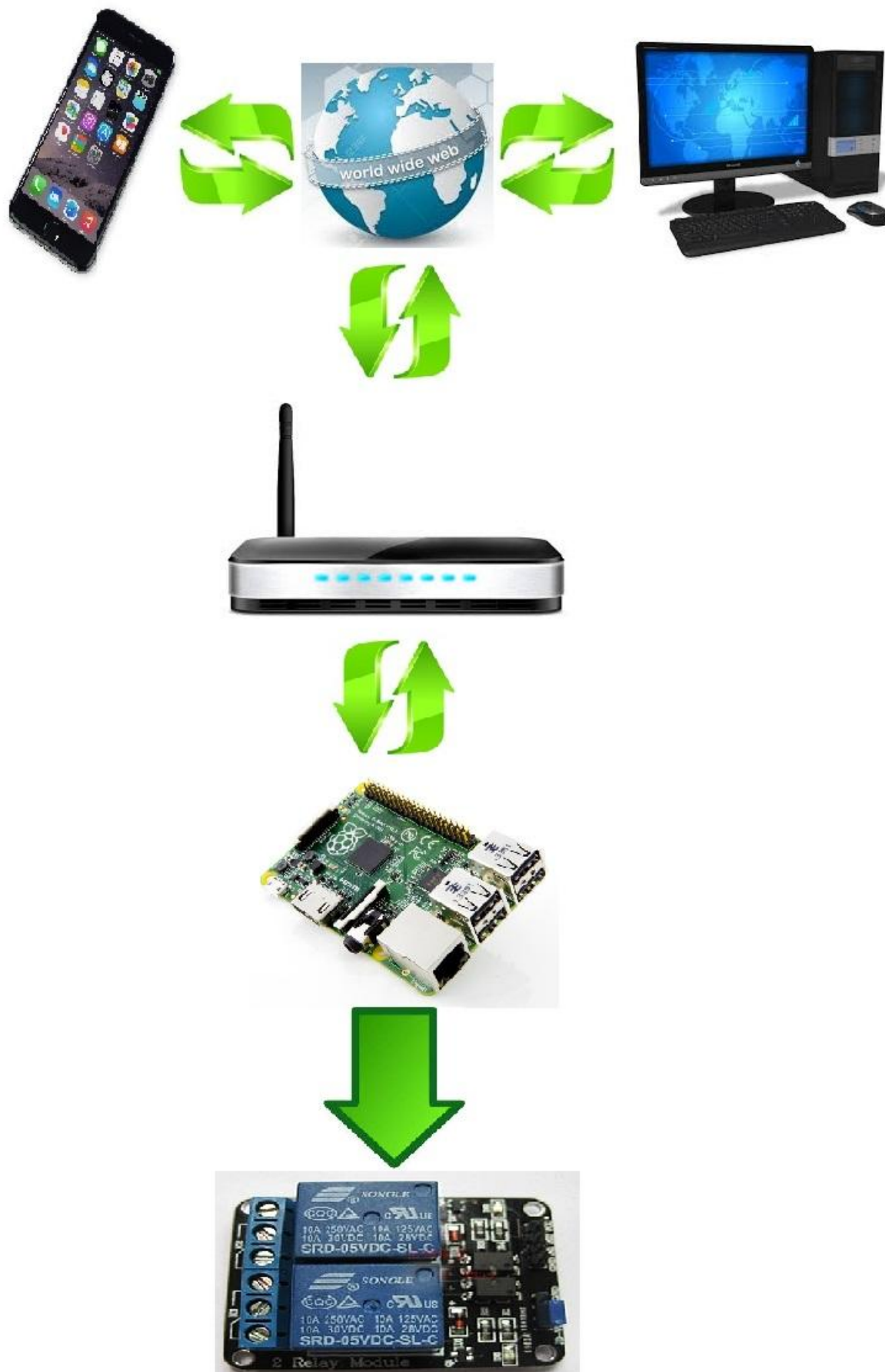
- Μεγαλύτερη ασφάλεια καθώς δημιουργούνται απειροελάχιστοι ιοί για Linux
- Μεγάλη σταθερότητα αφού περιέχουν ελάχιστα σφάλματα στον κώδικα του
- Είναι δωρεάν
- Οι απαιτήσεις σε υλικό είναι σαφώς χαμηλότερες, κάτι που τα κάνει ικανά να τρέξουν χωρίς προβλήματα ακόμα και σε παλαιότερους υπολογιστές
- Όλα τα προγράμματα αλλά και το ίδιο το λειτουργικό σύστημα αναβαθμίζονται αυτόματα με ένα κλικ.

- Όλα τα προγράμματα είναι δωρεάν και όλα είναι συγκεντρωμένα στα repositories (αποθήκες). Υπάρχουν ορισμένα επί πληρωμή προγράμματα αλλά ταυτόχρονα υπάρχουν και αρκετά δωρεάν υποκατάστατα τους που εκτελούν ακριβώς τις ίδιες λειτουργίες.

3 ΠΕΡΙΓΡΑΦΗ ΔΙΑΤΑΞΗΣ

Παρακάτω βλέπουμε το μπλοκ διάγραμμα της διάταξης μας:

Ανάπτυξη Web Interface για απομακρυσμένο έλεγχο με τη χρήση του μικροελεγκτή RaspberryPi



3.1 Χρησιμότητα του Raspberry Pi στην εφαρμογή μας

Το Raspberry Pi στην εφαρμογή μας το χρησιμοποιούμε για να στήσουμε ένα Webserver πάνω στο λειτουργικό σύστημα του (Raspbian/Linux), όπου εκεί θα βάλουμε τα αρχεία τις ιστοσελίδας μας, την οποία θα καλούμε κάθε φορά όταν μπαίνουμε μέσω της IP του Raspberry Pi στο δίκτυο, και μας δίνει την δυνατότητα στη συγκεκριμένη εφαρμογή να ελέγχουμε τη πλακέτα με τα Relay αλλά και γενικά τα pins σαν εξόδους. Επίσης, μας δίνει τη δυνατότητα να χρησιμοποιήσουμε τη κάρτα μνήμης microSD, στην οποία έχουμε εγκατεστημένα το Raspian, τον WebServer και οποιαδήποτε άλλη εφαρμογή, σε οποιοδήποτε Raspberry Pi ίδιας έκδοσης (A, B, B+ ,2) χωρίς καμία απολύτως αλλαγή, ακόμα και σε περίπτωση καταστροφής του μικροελεγκτή.

3.2 Χρησιμότητα του Modem-Router

Χρησιμεύει στο να έχουμε επικοινωνία μεταξύ του Raspberry Pi και τον εξωτερικό κόσμο, δηλαδή το Διαδίκτυο. Μας επιτρέπει ακόμα να έχουμε πρόσβαση στο Raspberry Pi από οποιοδήποτε άλλο τερματικό το οποίο έχει πρόσβαση στο δίκτυο μας χάρης στην σταθερή IP που έχουμε δώσει στον μικροελεγκτή.

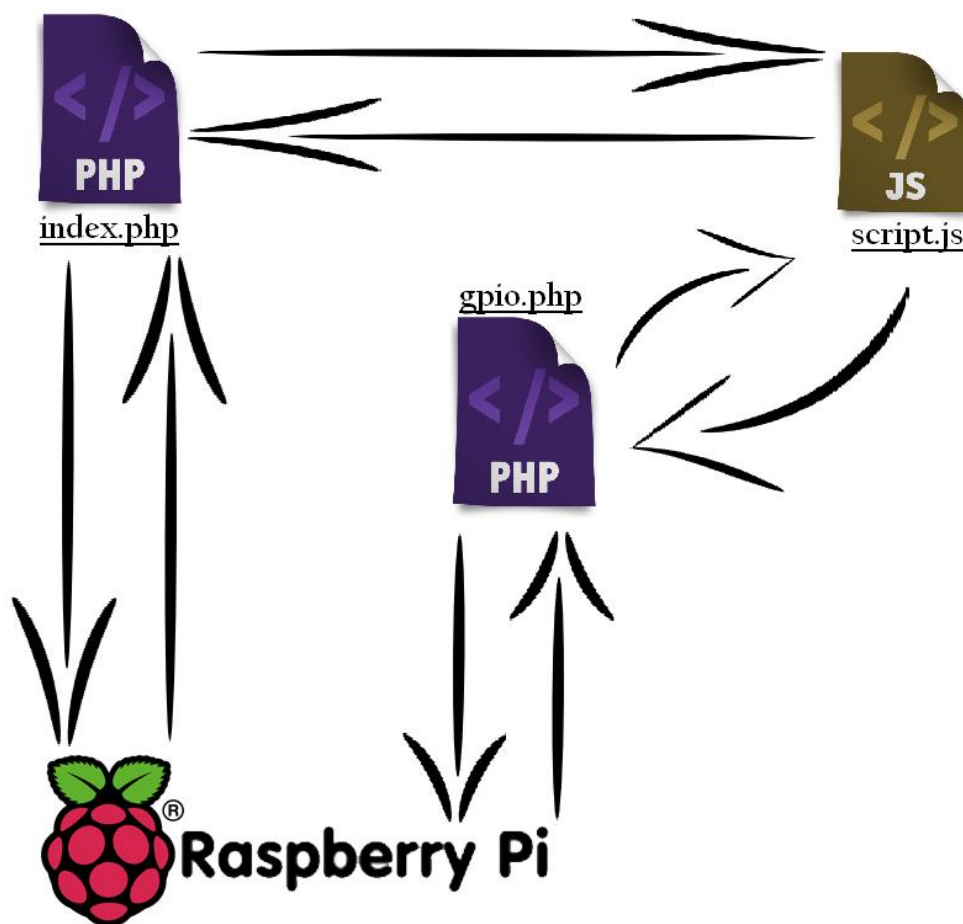
3.3 Χρησιμότητα του υπολογιστή/κινητού τηλεφώνου

Χρησιμεύει στον απομακρυσμένο έλεγχο του Raspberry Pi μέσω ενός Web Browser αφού έχουμε φυσικά πρόσβαση στο δίκτυο και γνωρίζοντας την IP του Raspberry Pi.

3.4 Χρησιμότητα του Relay

Χρησιμεύει στον έλεγχο φορτίων μέσω του μικροελεγκτή Raspberry Pi. Πιο συγκεκριμένα μπορεί να ελέγξουμε διάφορα φορτία των 230V διαμέσου των εξόδων 3,3V του μικροελεγκτή μας.

3.5 Περιγραφή διάταξης ιστοσελίδας



Στο αρχείο `index.php` περιέχονται όλες οι εντολές για την ενεργοποίηση των εξόδων και την δημιουργία της αρχικής ιστοσελίδας

με τα οχτώ κουμπιά βάση της κατάστασης των εξόδων. Επίσης, υπάρχει διασύνδεση με το αρχείο script.js (Javascript) σε περίπτωση αλλαγής της κατάστασης ενός κουμπιού. Η διαδραστικότητα με τον χρήστη επιτυγχάνεται μέσω του αρχείου αυτού. Το αρχείο gpio.php επιτελεί δύο λειτουργίες. Η πρώτη είναι η ενημέρωση του Raspberry Pi για την οποιαδήποτε αλλαγή στις εξόδους του. Η δεύτερη λειτουργία είναι η χρησιμοποίηση του από το αρχείο της Javascript για την ενημέρωση των εξόδων.

4 Υλοποίηση

4.1 Εγκατάσταση Raspbian στο Raspberry Pi

1. Εισάγουμε τη κάρτα μνήμης microSD (ελάχιστο μέγεθος 4GB) σε ένα υπολογιστή.
2. Κάνουμε Format στη κάρτα μνήμης με την εφαρμογή SD Formatter (https://www.sdcard.org/downloads/formatter_4/) και με την επιλογή size adjustment στο ON έτσι να γίνει προσαρμογή στο μέγεθος της κάρτας.
3. Κατεβάζουμε σε μορφή ZIP το πακέτο NOOBS (<https://www.raspberrypi.org/downloads/>).
4. Αποσυμπιέζουμε τα περιεχόμενα NOOBS στην κάρτα μνήμης.
5. Αφαιρούμε με ασφάλεια τη κάρτα μνήμης από τον υπολογιστή και την εισάγουμε στο Raspberry Pi.

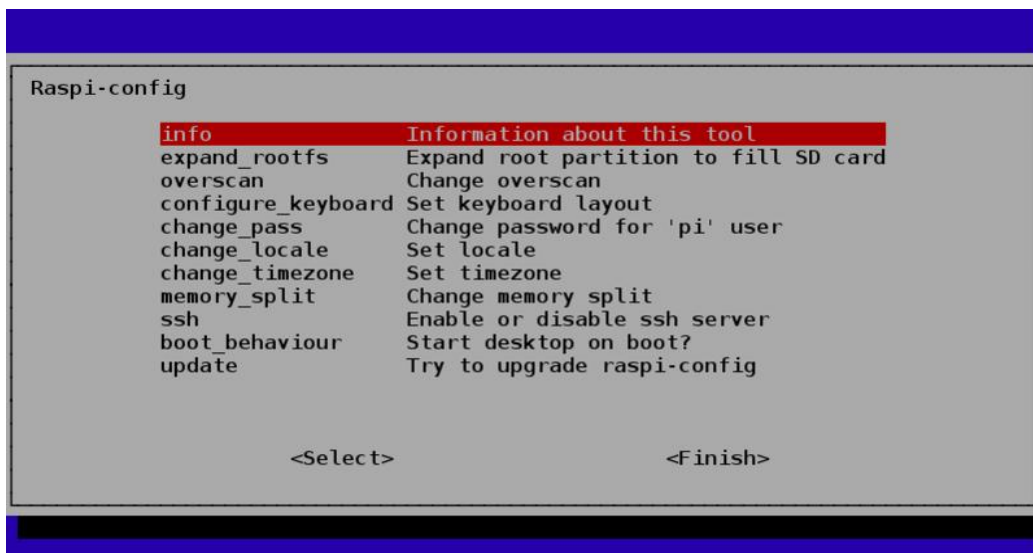
6. Συνδέουμε στο Raspberry Pi το πληκτρολόγιο, το ποντίκι και το συνδέουμε μέσω του HDMI σε κάποια οθόνη/τηλεόραση. Έπειτα συνδέουμε και τη τροφοδοσία.
7. Στο παράθυρο που θα μας εμφανίσει επιλέγουμε Raspbian.



8. Στη συνέχεια θα αρχίσει η εγκατάσταση του λειτουργικού συστήματος.



9. Μόλις τελειώσει η εγκατάσταση θα μας εμφανίσει ένα παράθυρο με επιλογές, γνωστό ως Raspi-config.

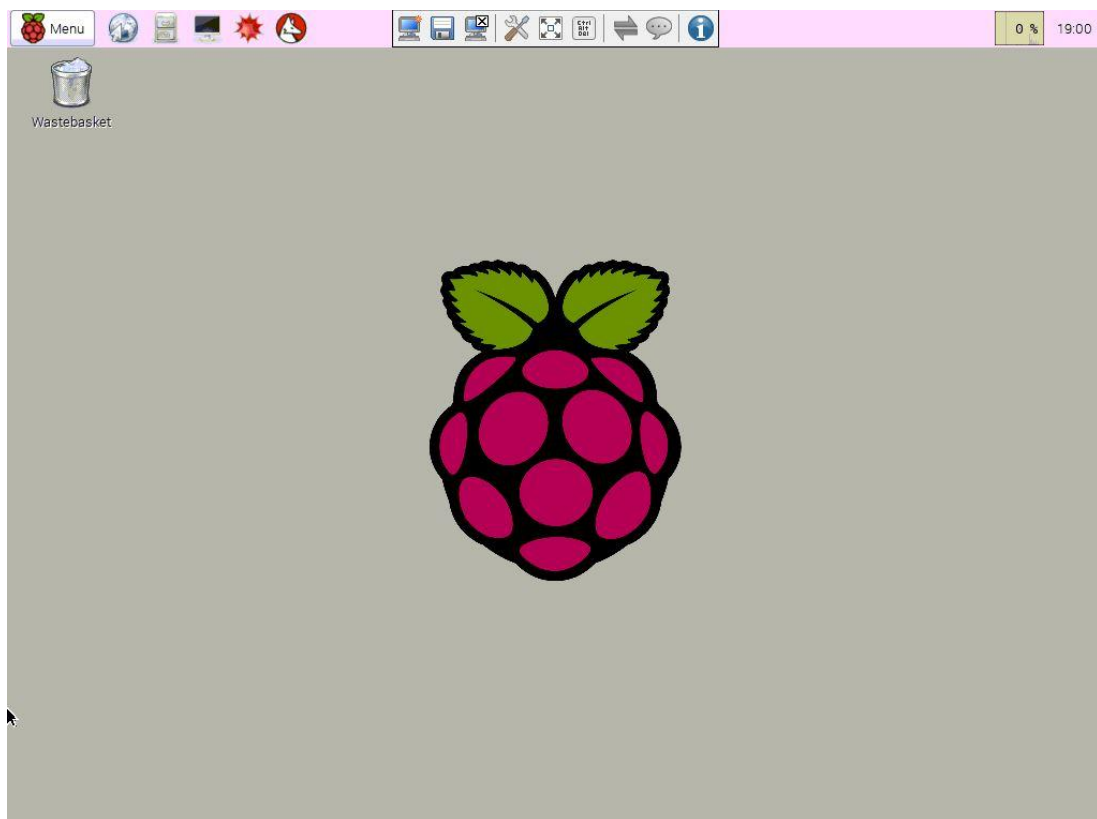


10. Χρησιμοποιώντας τα βελάκια ΠΑΝΩ και ΚΑΤΩ του πληκτρολογίου πατάμε ENTER στην επιλογή “Expand root partition to

fill SD card”. Με αυτή την επιλογή θα μπορούμε να έχουμε πρόσβαση σε όλο το μέγεθος της κάρτας μνήμης.

11. Στη συνέχεια, πατάμε την επιλογή “Start desktop on boot”. Στο επόμενο παράθυρο που θα μας εμφανίσει επιλέγουμε το “Yes”. Έτσι, δε θα μας εμφανίζει κάθε φορά τις επιλογές του Raspri-config αλλά θα κάνει εκκίνηση απευθείας στην επιφάνεια εργασίας του Raspbian. (Για να μπορέσουμε να αλλάξουμε αυτές τις επιλογές στο μέλλον, απλά ανοίγουμε ένα Terminal στο Raspbian και τρέχουμε την εντολή “sudo raspi-config”).

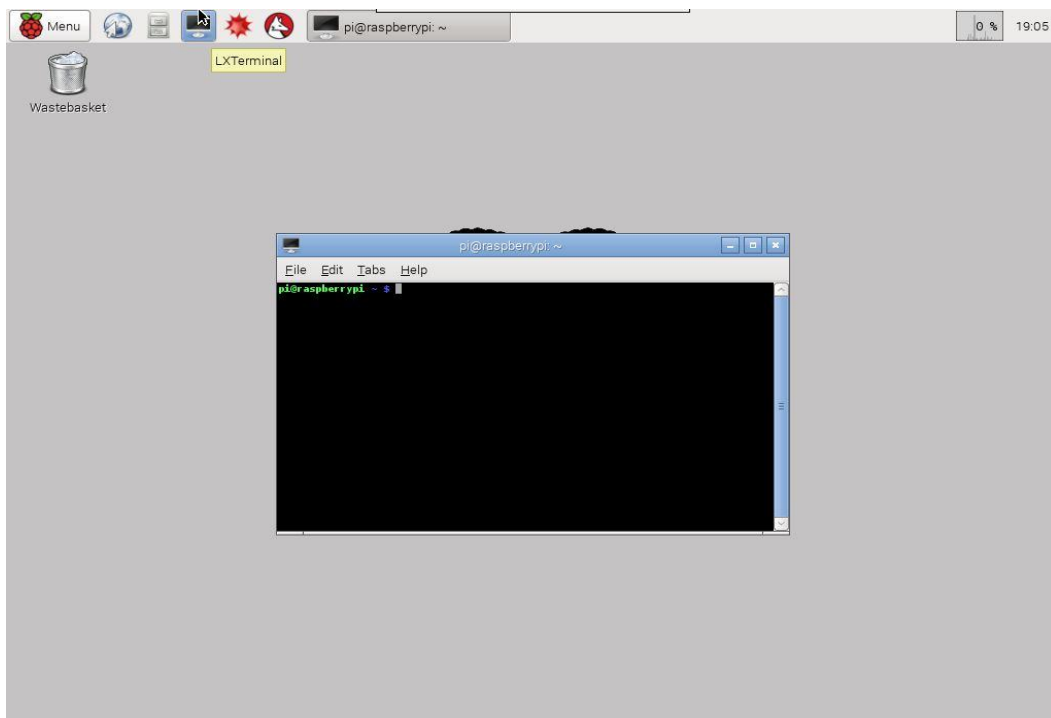
12. Τέλος, κάνοντας μία επανεκκίνηση το Raspberry Pi ξεκινάει απευθείας στην επιφάνεια εργασίας.



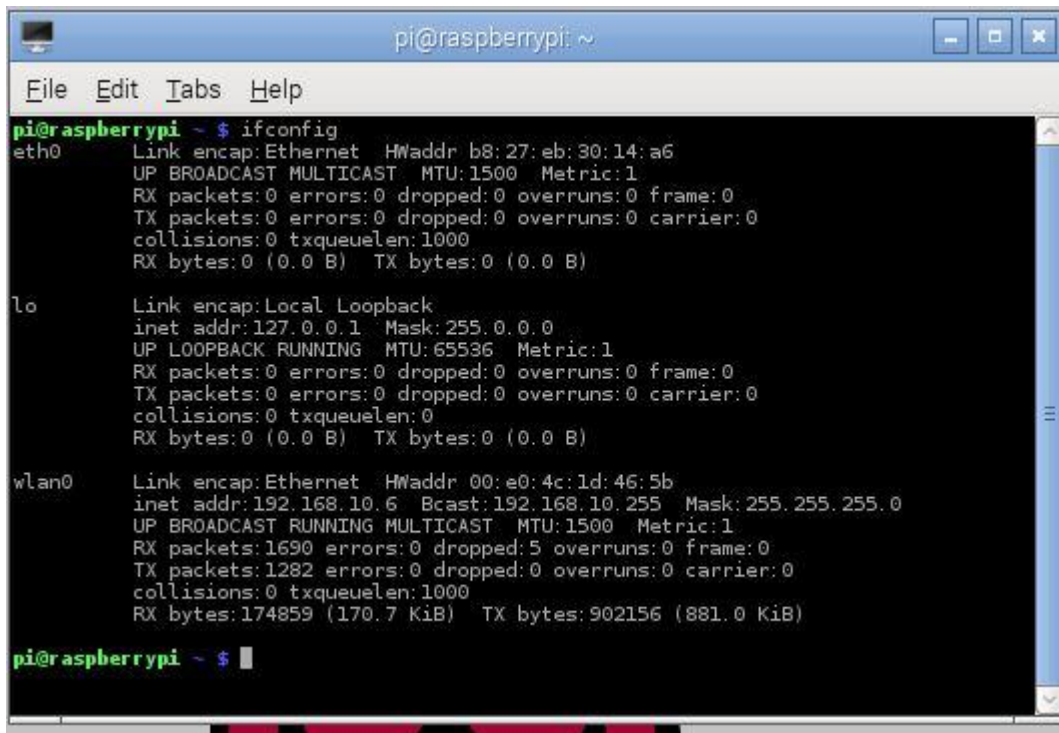
4.2 Απόδοση στατικής διεύθυνσης IP στο Raspberry Pi

Για να έχουμε πρόσβαση στην ψηφιακή πλατφόρμα για τον έλεγχο των εξόδων του Raspberry Pi, θα πρέπει να πληκτρολογήσουμε την διεύθυνση IP του μικροελεγκτή μας στη γραμμή διεύθυνσεως του εκάστοτε Web Browser. Όμως, σε κάθε εκκίνηση του Raspberry Pi, του αποδίδετε και μία πιθανώς διαφορετική διεύθυνση IP, ανάλογα με τις συσκευές που έχουν συνδεθεί στο Router (δυναμική IP-dynamic IP). Επιπροσθέτως, κάθε φορά που θα θέλαμε να έχουμε πρόσβαση στην ψηφιακή πλατφόρμα, θα έπρεπε να ελέγξουμε το Raspberry Pi και να βρούμε ποια διεύθυνση IP του έχει αποδοθεί, η οποία ως διαδικασία πιθανώς να χρειάζεστε ορισμένα περιφερειακά, όπως οθόνη, πληκτρολόγιο κ.α. Τελικώς, αντιλαμβανόμαστε ότι θα πρέπει να αποδώσουμε μία στατική διεύθυνση IP (static IP), η οποία δεν θα αλλάζει κάθε φορά

1. Κλικάρουμε το LXTerminal και ανοίγει ένα παράθυρο για εισαγωγή εντολών.



2. Πληκτρολογούμε 'ifconfig' και πατάμε Enter. Έτσι θα εμφανιστούν οι βασικές πληροφορίες για τους προσαρμογείς δικτύου.



```
pi@raspberrypi ~ $ ifconfig
eth0      Link encap:Ethernet  HWaddr b8:27:eb:30:14:a6
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

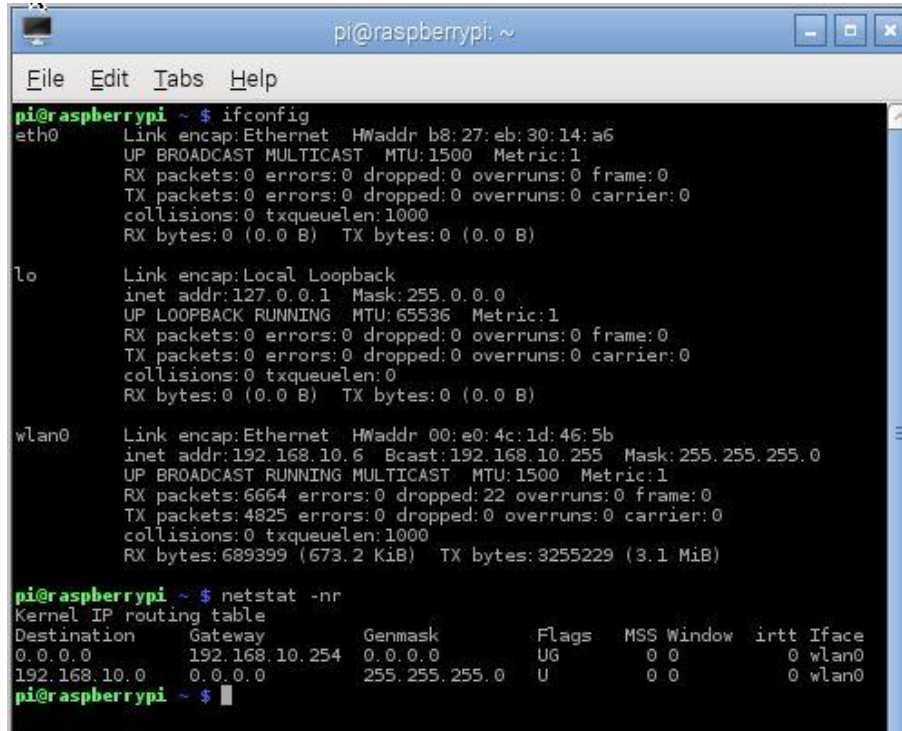
lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

wlan0     Link encap:Ethernet  HWaddr 00:a0:4c:1d:46:5b
          inet addr:192.168.10.6  Bcast:192.168.10.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:1690 errors:0 dropped:5 overruns:0 frame:0
          TX packets:1282 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:174859 (170.7 KiB)  TX bytes:902156 (881.0 KiB)

pi@raspberrypi ~ $
```

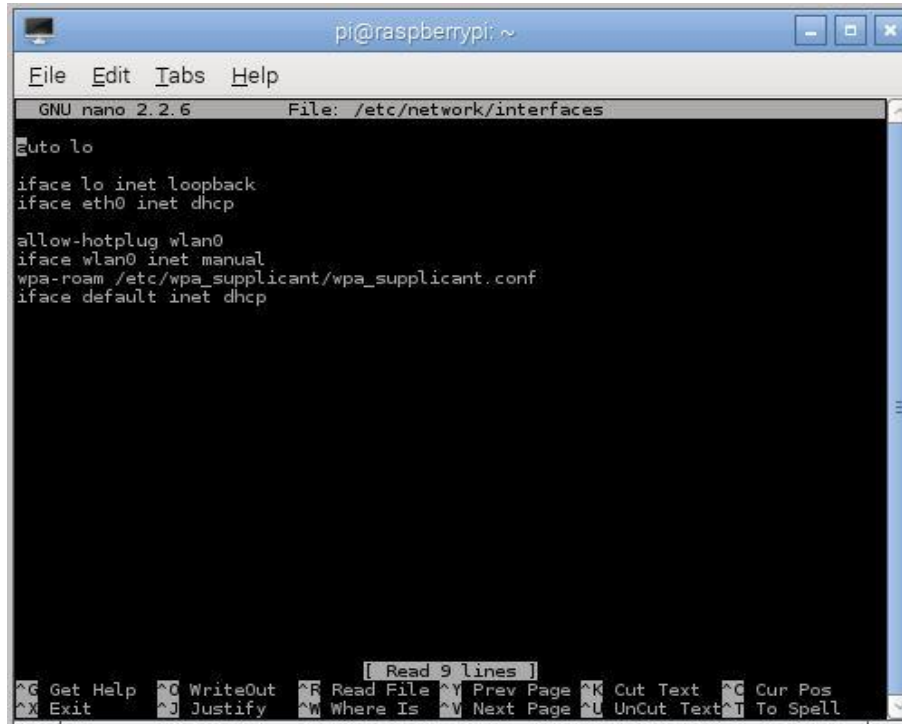
Αν έχουμε συνδεθεί στο Router μέσω καλωδίου RJ-45 (Ethernet), τότε μας ενδιαφέρουν οι πληροφορίες στον τομέα του eth0, ενώ αν έχουμε συνδεθεί ασύρματα (με χρήση κάποιου Wi-Fi dongle) οι πληροφορίες στον τομέα του wlan0. Κρατάμε τις διευθύνσεις inet addr, Bcast και Mask.

3. Στη συνέχεια πληκτρολογούμε 'netstat -nr' και πατάμε Enter. Οι πληροφορίες που μας ενδιαφέρουν είναι οι μη μηδενικές διευθύνσεις κάτω από τα Destination και Gateway.

Ανάπτυξη Web Interface για απομακρυσμένο έλεγχο
με τη χρήση του μικροελεγκτή RaspberryPi

```
pi@raspberrypi: ~  
File Edit Tabs Help  
pi@raspberrypi ~ $ ifconfig  
eth0      Link encap:Ethernet  HWaddr b8:27:eb:30:14:a6  
UP BROADCAST MULTICAST  MTU:1500  Metric:1  
RX packets:0 errors:0 dropped:0 overruns:0 frame:0  
TX packets:0 errors:0 dropped:0 overruns:0 carrier:0  
collisions:0 txqueuelen:1000  
RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)  
  
lo        Link encap:Local Loopback  
inet addr:127.0.0.1  Mask:255.0.0.0  
UP LOOPBACK RUNNING  MTU:65536  Metric:1  
RX packets:0 errors:0 dropped:0 overruns:0 frame:0  
TX packets:0 errors:0 dropped:0 overruns:0 carrier:0  
collisions:0 txqueuelen:0  
RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)  
  
wlan0     Link encap:Ethernet  HWaddr 00:e0:4c:1d:46:5b  
inet addr:192.168.10.6  Bcast:192.168.10.255  Mask:255.255.255.0  
UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1  
RX packets:6664 errors:0 dropped:22 overruns:0 frame:0  
TX packets:4825 errors:0 dropped:0 overruns:0 carrier:0  
collisions:0 txqueuelen:1000  
RX bytes:689399 (673.2 KiB)  TX bytes:3255229 (3.1 MiB)  
  
pi@raspberrypi ~ $ netstat -nr  
Kernel IP routing table  
Destination      Gateway          Genmask         Flags   MSS Window  irtt Iface  
0.0.0.0          192.168.10.254  0.0.0.0        UG      0 0        0 wlan0  
192.168.10.0     0.0.0.0         255.255.255.0  U       0 0        0 wlan0  
pi@raspberrypi ~ $
```

4. Για να αποδώσουμε την στατική διεύθυνση IP που θέλουμε πρέπει να αλλάξουμε τα δεδομένα του αρχείου interfaces στο φάκελο /etc/network. Οπότε, πληκτρολογούμε ‘sudo nano /etc/network/interfaces’.



```
pi@raspberrypi: ~
File Edit Tabs Help
GNU nano 2.2.6 File: /etc/network/interfaces
auto lo
iface lo inet loopback
iface eth0 inet dhcp

allow-hotplug wlan0
iface wlan0 inet manual
wpa-roam /etc/wpa_supplicant/wpa_supplicant.conf
iface default inet dhcp

[ Read 9 lines ]
^G Get Help ^C WriteOut ^F Read File ^Y Prev Page ^K Cut Text ^C Cur Pos
^X Exit ^J Justify ^W Where Is ^N Next Page ^L UnCut Text ^T To Spell
```

5. Αν είμαστε συνδεδεμένοι διαμέσου του θύρας Ethernet του Raspberry Pi τότε αλλάζουμε το ‘iface eth0 inet dhcp’ σε ‘iface eth0 inet static’ και ακριβώς από κάτω γράφουμε:

address 192.168.10.10 (inet addr-το 10 μπορεί να είναι από 1 έως 253)

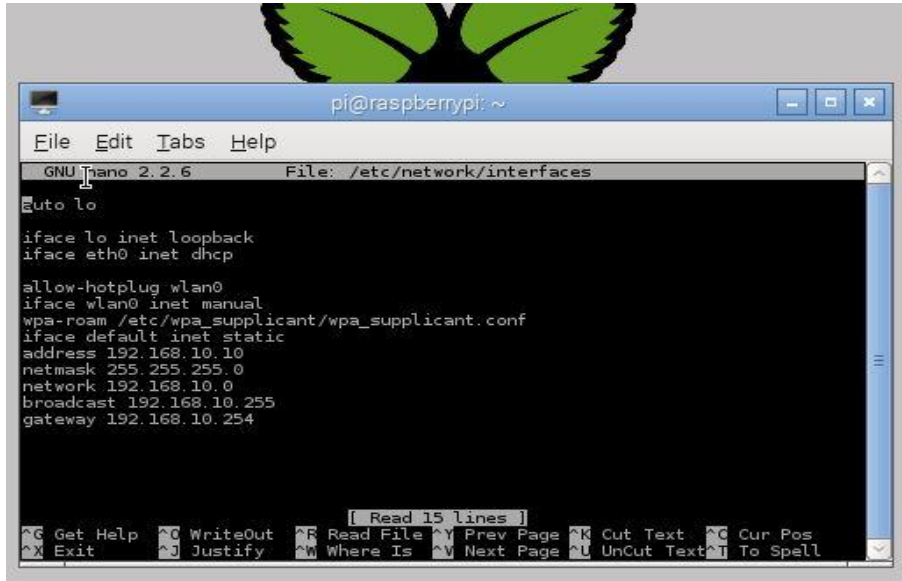
netmask 255.255.255.0 (Mask)

network 192.168.10.0 (Destination)

broadcast 192.168.10.255 (Bcast)

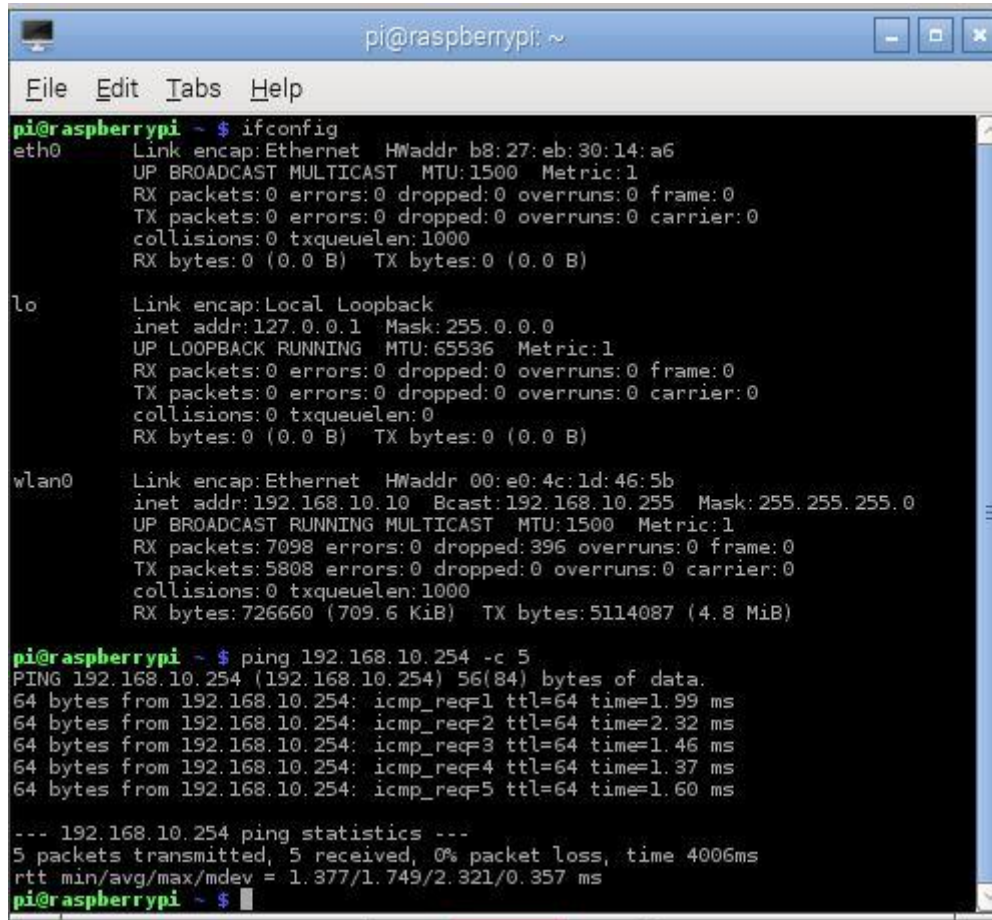
gateway 192.168.10.254 (Gateway)

Αν είμαστε συνδεδεμένοι ασύρματα τότε αλλάζουμε το ‘iface default inet dhcp’ σε ‘iface default inet static’ και ακριβώς από κάτω γράφουμε τις ίδιες πληροφορίες με πριν.

Ανάπτυξη Web Interface για απομακρυσμένο έλεγχο
με τη χρήση του μικροελεγκτή RaspberryPi

```
pi@raspberrypi: ~  
File Edit Tabs Help  
GNU nano 2.2.6 File: /etc/network/interfaces  
#auto lo  
iface lo inet loopback  
iface eth0 inet dhcp  
  
allow-hotplug wlan0  
iface wlan0 inet manual  
wpa-roam /etc/wpa_supplicant/wpa_supplicant.conf  
iface default inet static  
address 192.168.10.10  
netmask 255.255.255.0  
network 192.168.10.0  
broadcast 192.168.10.255  
gateway 192.168.10.254  
  
[ Read 15 lines ]  
Get Help WriteOut Read File Prev Page Cut Text Cur Pos  
Exit Justify Where Is Next Page UnCut Text To Spell
```

6. Πατάμε Ctrl+X, y και Enter έτσι ώστε να αποθηκευτούν οι αλλαγές και να κλείσουμε τον επεξεργαστή κειμένου.
7. Κάνουμε επανεκκίνηση πληκτρολογώντας ‘sudo reboot’.
8. Τέλος, για να επαληθεύσουμε ότι εφαρμόστηκαν οι αλλαγές μας πληκτρολογούμε αρχικά ‘ifconfig’ για να ελέγξουμε ότι η inet addr είναι αυτή που δώσαμε στο βήμα 5 και έπειτα ‘ping 192.168.10.254 -c 5’ για να ελέγξουμε αν έχουμε επιτυχή ανταλλαγή πακέτων μεταξύ του Router και του Raspberry Pi.



```
pi@raspberrypi: ~  
File Edit Tabs Help  
pi@raspberrypi ~ $ ifconfig  
eth0      Link encap:Ethernet  HWaddr b8:27:eb:30:14:a6  
          UP BROADCAST MULTICAST  MTU:1500  Metric:1  
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0  
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0  
          collisions:0 txqueuelen:1000  
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)  
  
lo        Link encap:Local Loopback  
          inet addr:127.0.0.1  Mask:255.0.0.0  
          UP LOOPBACK RUNNING  MTU:65536  Metric:1  
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0  
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0  
          collisions:0 txqueuelen:0  
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)  
  
wlan0     Link encap:Ethernet  HWaddr 00:e0:4c:1d:46:5b  
          inet addr:192.168.10.10  Bcast:192.168.10.255  Mask:255.255.255.0  
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1  
          RX packets:7098 errors:0 dropped:396 overruns:0 frame:0  
          TX packets:5808 errors:0 dropped:0 overruns:0 carrier:0  
          collisions:0 txqueuelen:1000  
          RX bytes:726660 (709.6 KiB)  TX bytes:5114087 (4.8 MiB)  
  
pi@raspberrypi ~ $ ping 192.168.10.254 -c 5  
PING 192.168.10.254 (192.168.10.254) 56(84) bytes of data:  
64 bytes from 192.168.10.254: icmp_req=1 ttl=64 time=1.99 ms  
64 bytes from 192.168.10.254: icmp_req=2 ttl=64 time=2.32 ms  
64 bytes from 192.168.10.254: icmp_req=3 ttl=64 time=1.46 ms  
64 bytes from 192.168.10.254: icmp_req=4 ttl=64 time=1.37 ms  
64 bytes from 192.168.10.254: icmp_req=5 ttl=64 time=1.60 ms  
  
--- 192.168.10.254 ping statistics ---  
5 packets transmitted, 5 received, 0% packet loss, time 4006ms  
rtt min/avg/max/mdev = 1.377/1.749/2.321/0.357 ms  
pi@raspberrypi ~ $
```

4.3 Εγκατάσταση απαιτούμενων πακέτων και εφαρμογών

4.3.1 Εγκατάσταση βιβλιοθήκης WiringPi

Η βιβλιοθήκη WiringPi μας προσφέρει διάφορα πλεονεκτήματα. Το βασικότερο από αυτά και το οποίο θα μας χρησιμεύσει είναι ότι μπορούμε να χρησιμοποιήσουμε τις εντολές της απευθείας στο παράθυρο εντολών (LXTerminal). Τη βιβλιοθήκη WiringPi μπορούμε να τη προμηθευτούμε εύκολα με το εργαλείο GIT.

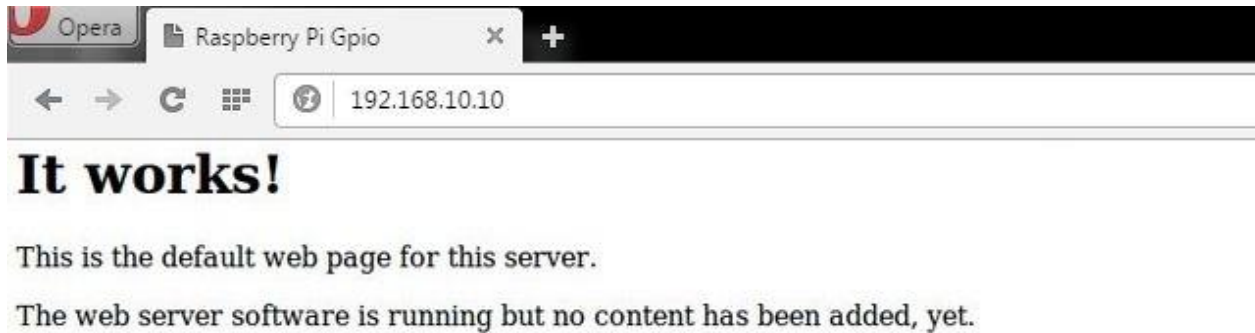
1. Ανοίγουμε το LXTerminal και εκτελούμε την εντολή ‘`sudo apt-get install git-core`’. Μετά το πέρας της εντολής θα έχουμε εγκαταστήσει το εργαλείο GIT.
2. Στη συνέχεια, εκτελούμε την εντολή ‘`git clone git://git.drogon.net/wiringPi`’, με την οποία θα γίνει λήψη της εγκατάστασης της βιβλιοθήκης WiringPi.
3. Έπειτα μεταβαίνουμε στον φάκελο του WiringPi εκτελώντας ‘`cd wiringPi`’ και τέλος, εκτελούμε την εντολή ‘`./build`’.

Για να επαληθεύσουμε ότι όλα έγιναν σωστά αρκεί να εκτελέσουμε την εντολή ‘`gpio mode 0 out`’. Αν δεν μας εμφανίσει κάποιο μήνυμα του τύπου η εντολή δε βρέθηκε, τότε η βιβλιοθήκη WiringPi εγκαταστάθηκε και δουλεύει σωστά.

4.3.2 Εγκατάσταση Apache Web Server και της γλώσσας PHP5

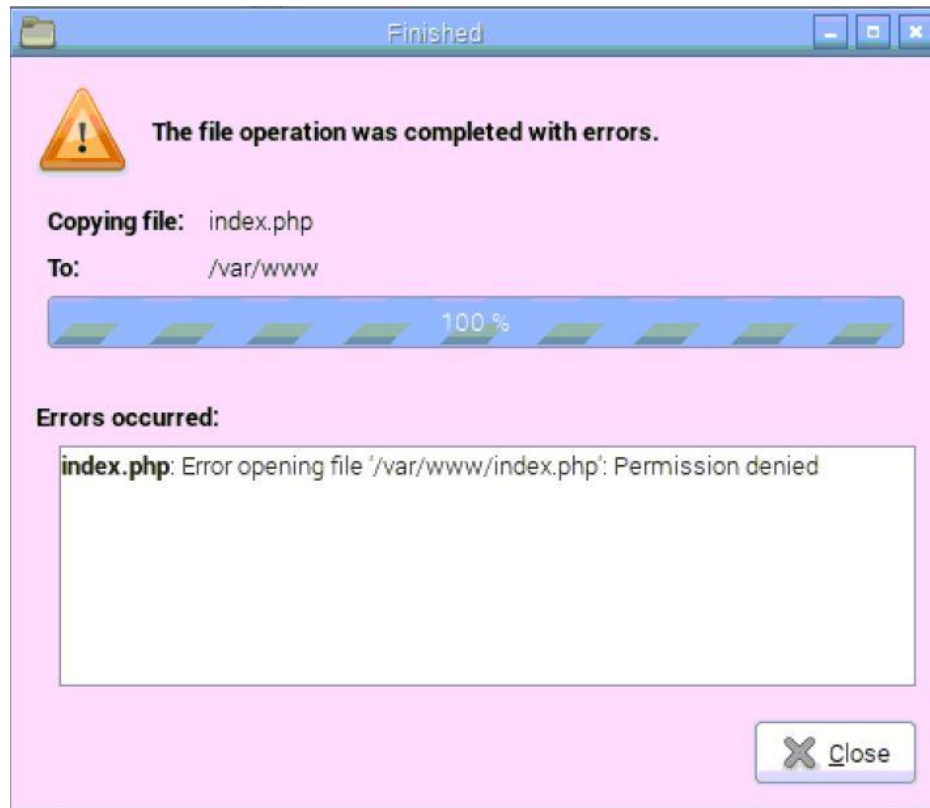
1. Κλικάρουμε το LXTerminal και ανοίγει το παράθυρο για εισαγωγή εντολών.
2. Πληκτρολογούμε “`sudo apt-get install apache2 php5 libapache2-mod-php5`” και πατάμε Enter. Με αυτή την εντολή θα εγκατασταθεί ο Apache Web Server, η PHP καθώς και η διασύνδεση τους. Γράφοντας τη λέξη ‘`sudo`’ μπροστά από κάθε εντολή μας επιτρέπει να εκτελέσουμε την εκάστοτε ενέργεια με δικαιώματα διαχειριστή.
3. Αφού εγκαταστάθηκε ο Apache Web Server μπορούμε να δοκιμάσουμε αν έχουν γίνει όλα σωστά μέχρι εδώ ανοίγοντας έναν οποιονδήποτε Web Browser σε οποιοδήποτε άλλο υπολογιστή ή κινητό τηλέφωνο και απλά να πληκτρολογήσουμε στο πεδίο της διεύθυνσεως την στατική IP που έχουμε δώσει στο Raspberry Pi (192.168.10.10). Αν

όλα έχουν αποπερατωθεί με επιτυχία θα μας εμφανιστεί μία ιστοσελίδα της παρακάτω μορφής.



4.4 Αντιγραφή των αρχείων της ιστοσελίδας στο φάκελο του Web Server

Για να εμφανίζεται η ιστοσελίδα που θέλουμε θα πρέπει να τοποθετήσουμε τα αρχεία στον φάκελο `/var/www`. Με την εγκατάσταση του Apache Web Server αντιγράφεται σε αυτόν τον φάκελο ένα αρχείο `index.html` με το οποίο μας εμφανίζει το μήνυμα “It works”, όπως είδαμε στο προηγούμενο κεφάλαιο. Θα πρέπει να διαγράψουμε αυτό το αρχείο και να αντιγράψουμε τα αρχεία της ιστοσελίδας μας (φάκελος `data`, `index.php`, `gpio.php`, `script.js`). Όμως, ο φάκελος `/var/www` έχει δικαιώματα διαχειριστή και δεν μπορούμε να προβούμε σε καμία αλλαγή σε αυτόν, καθώς δεν μας καλύπτουν τα δικαιώματα απλού χρήστη που έχουμε. Επομένως μία απλή αντιγραφή-επικόλληση δεν θα έχει επιτυχία.

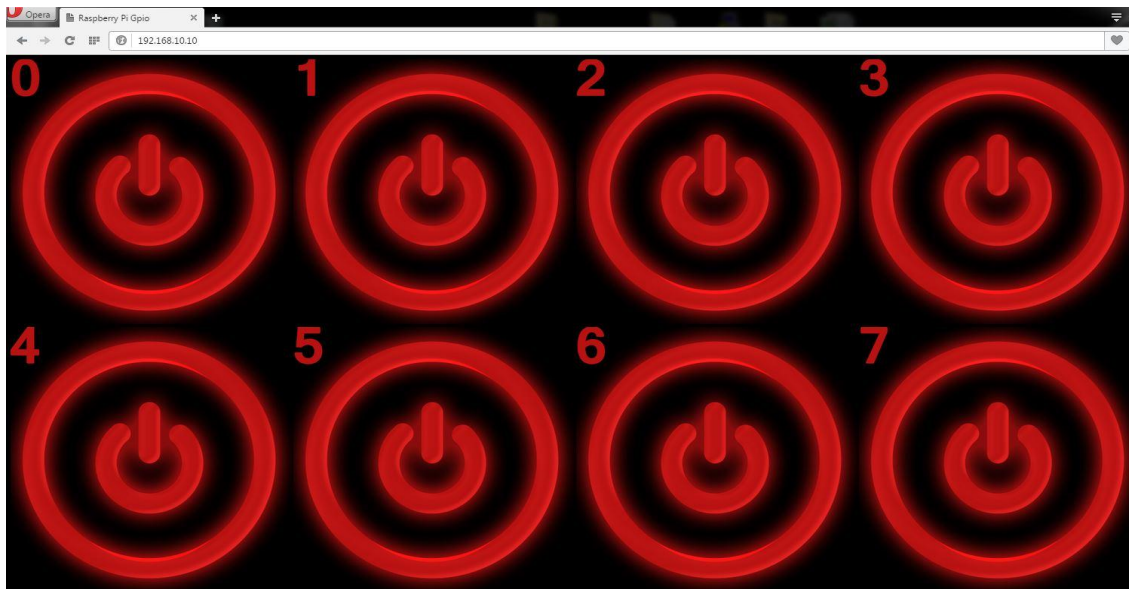


1. Ανοίγουμε το LXTerminal και εκτελούμε 'sudo rm /var/www/index.html'. Η εντολή rm θα διαγράψει το αρχείο index.html στο φάκελο /var/www και θα εκτελεστεί με δικαιώματα διαχειριστή καθώς έχουμε εισάγει την εντολή sudo από μπροστά. Επομένως, ο φάκελος θα είναι πλέον κενός.
2. Αντιγράφουμε τα αρχεία της ιστοσελίδας στην επιφάνεια εργασίας του Raspbian (αντιγραφή-επικόλληση).
3. Εκτελούμε την εντολή 'cp -r Desktop/{data,index.php,gpio.php,script.js} /var/www'. Με την εντολή cp θα αντιγραφούνε και τα τέσσερα στοιχεία μαζί στον φάκελο /var/www. Το πρόθεμα -r μας χρησιμεύει στο να αντιγραφεί ο φάκελος data σταδιακά, επειδή αν δεν το χρησιμοποιούσαμε θα έβγαζε σφάλμα.

4.5 Έλεγχος λειτουργίας

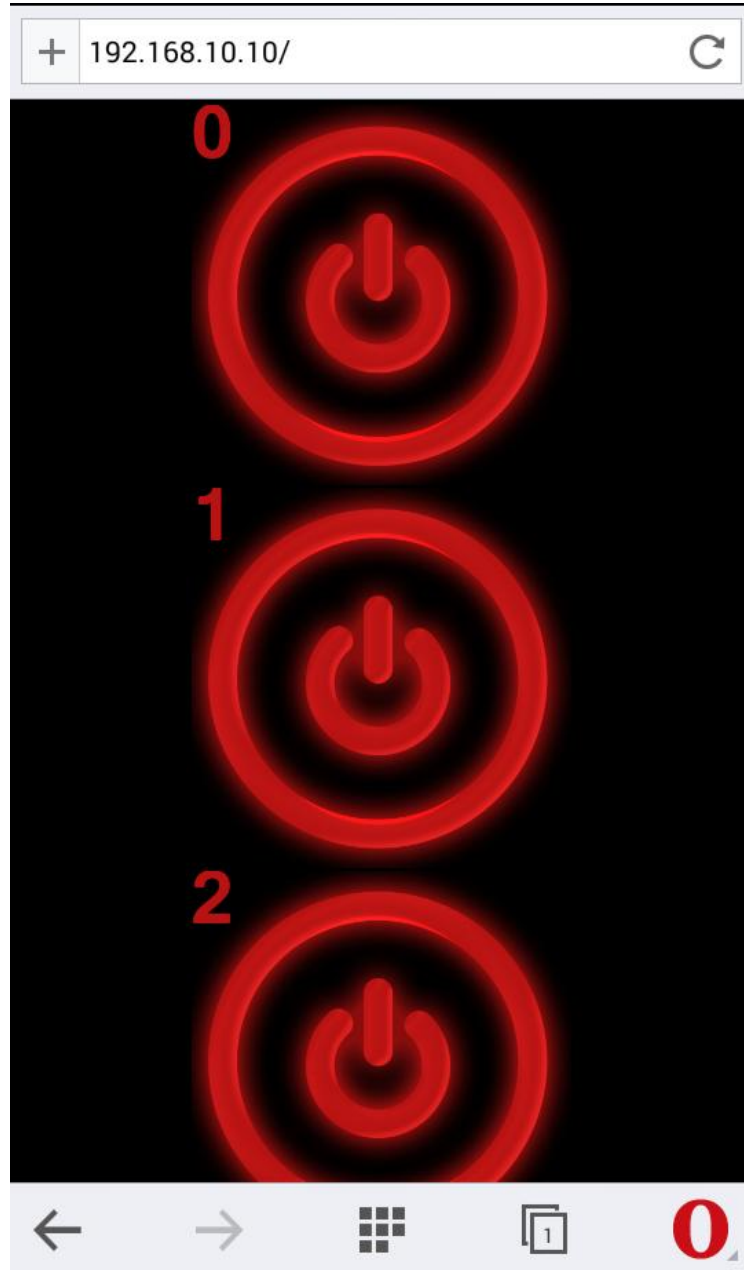
Όλα πλέον είναι έτοιμα για να ελέγξουμε αν λειτουργούν σωστά. Το μόνο που έχουμε να κάνουμε είναι να έχουμε συνδεδεμένο το Raspberry Pi στο ρεύμα και απλά να το ελέγξουμε μέσω ενός Web Browser ενός υπολογιστή ή κινητού τηλεφώνου συνδεδεμένο στο δίκτυο μας.

1. Έχοντας το Raspberry Pi συνδεδεμένο στο ρεύμα, ανοίγουμε ένα Web Browser σε κάποιο υπολογιστή ή κινητό τηλέφωνο, που είναι συνδεδεμένο στο δίκτυο μας είτε ενσύρματα είτε ασύρματα.
2. Πληκτρολογούμε στο πεδίο διεύθυνσης του Web Browser την διεύθυνση IP του Raspberry Pi (στη περίπτωση μας 192.168.10.10).
3. Αν όλα έχουν διεκπεραιωθεί σωστά τότε θα δούμε την παρακάτω ιστοσελίδα με τα οχτώ κουμπιά-διακόπτες για τις εξόδους του Raspberry Pi.



Στη περίπτωση που ανοίξουμε την ιστοσελίδα από smartphone ή tablet τότε τα κουμπιά-διακόπτες θα είναι στοιχισμένα στο κέντρο.

Ανάπτυξη Web Interface για απομακρυσμένο έλεγχο
με τη χρήση του μικροελεγκτή RaspberryPi



5 Παράρτημα 1

5.1 Περιγραφή βασικών εντολών της βιβλιοθήκης WiringPi

Με τη βιβλιοθήκη WiringPi μπορούμε να χρησιμοποιήσουμε μία εντολή για τον έλεγχο του Raspberry Pi έναντι πολλών. Οι βασικές εντολές που θα χρησιμοποιήσουμε στην εργασίας μας είναι οι εξής:

- gpio mode 0 out - ορίζει το pin 0 του Raspberry Pi βάση της βιβλιοθήκης WiringPi ως έξοδο
- gpio mode 6 in - ορίζει το pin 6 του Raspberry Pi βάση της βιβλιοθήκης WiringPi ως είσοδο
- gpio read 3 – διαβάζει την κατάσταση (0 ή 1, high ή low) του pin 3 του Raspberry Pi βάση της βιβλιοθήκης WiringPi
- gpio write 2 1 – ενεργοποιεί (high) το pin 2 του Raspberry Pi βάση της βιβλιοθήκης WiringPi (προϋποθέτει το pin 2 να έχει οριστεί ως έξοδος)
- gpio write 1 0 – απενεργοποιεί (low) το pin 1 του Raspberry Pi βάση της βιβλιοθήκης WiringPi (προϋποθέτει το pin 1 να έχει οριστεί ως έξοδος)

5.2 Τρόποι αρίθμησης των pins του Raspberry Pi

Για να αναφερθούμε σε κάποιο συγκεκριμένο pin του Raspberry Pi έχουμε κυρίως δυο τρόπους. Ο πρώτος είναι ανάλογα τον αριθμό του pin που θέλουμε να χρησιμοποιήσουμε (1, 2, 3...). Στα pin των 0, 3.3 και 5 Volt ακόμα και αν έχουν αριθμηθεί, δεν μπορούμε να αλλάξουμε την κατάσταση τους. Ο δεύτερος τρόπος είναι βάση της αρίθμησης του

μικροελεγκτή BCM2835 της Broadcom που περιέχει το Raspberry Pi B+ (0, 1, 2...).

Στη βιβλιοθήκη WiringPi, που θα χρησιμοποιήσουμε, υπάρχει διαφορετική αρίθμηση. Για παράδειγμα, το pin 15 (GPIO22) στον μικροελεγκτή του Raspberry Pi B+ είναι το 22, ενώ για τη βιβλιοθήκη WiringPi είναι το 3. Επίσης, θα πρέπει να είμαστε προσεκτικοί στην αναφορά κάθε pin, καθώς αναφέροντας απλώς τον αριθμό 8, υπάρχουν πολλές εκδοχές για το ποιο εννοούμε. Μπορεί να είναι το τέταρτο pin της δεξιάς στήλης (GPIO14), μπορεί να είναι pin 24 (GPIO8), όπως επίσης και το pin 3 (GPIO2) για την βιβλιοθήκη WiringPi.

WiringPi	BCM	Όνομα	Pin		Όνομα	BCM	WiringPi
3.3V			1	2	5V		
8	2	GPIO2	3	4	5V		
9	3	GPIO3	5	6	0V-Ground		
7	4	GPIO4	7	8	GPIO14	14	15
0V-Ground			9	10	GPIO15	15	16
0	17	GPIO17	11	12	GPIO18	18	1
2	27	GPIO27	13	14	0V-Ground		
3	22	GPIO22	15	16	GPIO23	23	4
3.3V			17	18	GPIO24	24	5
12	10	GPIO10	19	20	0V-Ground		
13	9	GPIO9	21	22	GPIO25	25	6
14	11	GPIO11	23	24	GPIO8	8	10
0V-Ground			25	26	GPIO7	7	11
	0	ID_SD	27	28	ID_SC	1	
21	5	GPIO5	29	30	0V-Ground		
22	6	GPIO6	31	32	GPIO12	12	26
23	13	GPIO13	33	34	0V-Ground		
24	19	GPIO19	35	36	GPIO16	16	27
25	26	GPIO26	37	38	GPIO20	20	28
0V-Ground			39	40	GPIO21	21	29

5.3 Τεχνικά χαρακτηριστικά των εκδόσεων του Raspberry Pi

Raspberry Pi	A	A+	B	B+	2
Μικροελεγκτής	Broadcom BCM2835	Broadcom BCM2835	Broadcom BCM2835	Broadcom BCM2835	Broadcom BCM2836
CPU	ARM11	ARM11	ARM11	ARM11	Cortex A7
Πυρήνες	1	1	1	1	4
Clock	700 MHz	700 MHz	700 MHz	700 MHz	900 MHz
GPU	VideoCore IV	VideoCore IV	VideoCore IV	VideoCore IV	VideoCore IV
RAM	256 MB	256 MB	512 MB	512 MB	1 GB
USB θύρες	1	1	2	4	4
Τύπος κάρτας	SD	microSD	SD	microSD	microSD
Pins	26	40	26	40	40

5.4 Δευτερεύουσες λειτουργίες των ακροδεκτών του Raspberry Pi

Ορισμένοι ακροδέκτες του Raspberry Pi μπορούν να επιτελέσουν και άλλες λειτουργίες εκτός από απλές εισοδοι και έξοδοι. Μερικές φορές ίσως να χρειάζονται και κάποια συγκεκριμένη παραμετροποίηση.

Τα pins 3 και 5 (SDA και SCL) χρησιμοποιούνται για διασύνδεση I2C (I2C). Μέσω αυτών των δύο ακροδεκτών μπορούμε να συνδέσουμε στο Raspberry Pi διάφορες άλλα απλά τσιπ όπως το MCP23017 (για επέκταση των ακροδεκτών Raspberry Pi).

Τα pins 27 και 28 χρησιμεύουν για την επικοινωνία του εκάστοτε συνδεδεμένου HAT (Hardware Attached on Top). Τα HATs είναι διάφορες πλακέτες οι συνδέονται πάνω από το Raspberry Pi και στηρίζονται στους ακροδέκτες του. Μέσω των pins 27 και 28 το Raspberry Pi επικοινωνεί με το Hat που έχει συνδεθεί και αναγνωρίζει ποιο HAT έχει συνδεθεί και πως πρέπει να παραμετροποιήσει τα υπόλοιπα pins κατά τη διάρκεια της εκκίνησης του Raspberry Pi. Υπάρχουν μόνο στο Raspberry Pi B+ και Raspberry Pi 2.

Με τα pins 8 και 10 μπορούμε να επικοινωνήσουμε σειριακά με κάποιο υπολογιστή ή κάποια άλλη συσκευή που υποστηρίζει σειριακή επικοινωνία (UART-universal asynchronous receiver/transmitter). Με

το pin 8 (GPIO 14) γίνεται η μετάδοση (TXD) και με το pin 10 (GPIO 15) η λήψη (RXD).

Τα pins 19, 21, 23, 24, 26, 35, 38 και 40 μπορούν να χρησιμοποιηθούν για σύνδεση και επικοινωνία διαμέσου του πρωτοκόλλου SPI (Serial Peripheral Interface) με διάφορες άλλες μικροσυσκευές, όπως αισθητήρες ή και μετατροπείς (A2D και D2A).

Τέλος, το pin 12 μπορεί να χρησιμοποιηθεί ως έξοδος PWM (Pulse width modulation-Διαμόρφωση του Πλάτους του Παλμού), έτσι ώστε να έχουμε τη δυνατότητα να ελέγχουμε κινητήρες για παράδειγμα.

5.5 GIT

Το Git είναι ένα σύστημα ελέγχου διανεμόμενης έκδοσης και διαχείρισης πηγαίου κώδικα (SCM) με έμφαση στην ταχύτητα, στην ακεραιότητα των δεδομένων και στην υποστήριξη για κατανεμημένες μη γραμμικές ροές εργασίας. Το Git σχεδιάστηκε και αναπτύχθηκε αρχικά από τον Λίνους Τόρβαλντς για τη ανάπτυξη του πυρήνα Linux το 2005 και έχει γίνει από τότε το πιο πλατιά διαδεδομένο σύστημα ελέγχου εκδόσεων για ανάπτυξη λογισμικού. Πολλοί προγραμματιστές μπορούν να συμμετέχουν και να συνεισφέρουν στο ίδιο project. Το git, λειτουργώντας στο σύστημα μας, εν γένει δημιουργεί μια δικιά του βάση δεδομένων (σαν ένα filesystem) το οποίο αντί να έχει ως βασική λειτουργία την αποθήκευση, τροποποίηση και διαγραφή δεδομένων, μόνο αποθηκεύει καινούργια δεδομένα. Στη ρίζα του συστήματος αυτού βρίσκεται το repository, που είναι το working directory του κάθε project. Με οποιαδήποτε αλλαγή κάποιου αρχείου στο repository μας, το git ανιχνεύει τις αλλαγές και προσθέτει στη βάση δεδομένου του ό,τι είναι διαφορετικό. Τα δεδομένα που έμειναν ανέπαφα απλώς συνδέονται με τις προηγούμενες υπάρξεις αυτών ώστε να μην υφίστανται πολλαπλές υπάρξεις του ίδιου αρχείου.

6 Παράρτημα 2

6.1 Επεξήγηση κώδικα

6.1.1 Κώδικας αρχείου gpio.php

```
<?php
//Χρήση της εντολής GET για την ανάγνωση των τιμών
if (isset($_GET["pin"]) && isset($_GET["status"])) {
    $pin = strip_tags($_GET["pin"]);
    $status = strip_tags($_GET["status"]);
    //Έλεγχος για το αν οι τιμές είναι αριθμοί και μέσα στα
    //προβλεπόμενα όρια.
    if ( (is_numeric($pin)) && (is_numeric($status)) && ($pin <= 7)
        && ($pin >= 0) && ($status == "0") || ($status == "1") ) {
        //αρχικοποίηση των ακροδεκτών του Raspberry Pi ως έξοδοι
        system("gpio mode ".$pin." out");
        //αρχικοποίηση των ακροδεκτών σε κατάσταση 0 ή 1
        //αναλόγως με τις τιμές που έχουμε πάρει από πριν
        if ($status == "0" ) { $status = "1"; }
        else if ($status == "1" ) { $status = "0"; }
        //Η εντολή system μας επιτρέπει να εκτελέσουμε τις εντολές
        //της βιβλιοθήκης WiringPi απευθείας από τον κώδικα της
        //PHP
        system("gpio write ".$pin." ".$status );
        //Η εντολή exec μας επιτρέπει να εκτελέσουμε τις εντολές
        //της βιβλιοθήκης WiringPi απευθείας από τον κώδικα της
        //PHP και να αποθηκεύσουμε την τιμή που επιστρέφεται σε
        //κάποια μεταβλητή
        exec ("gpio read ".$pin, $status, $return );
```

```

//επιστρέφει τη κατάσταση του pin
echo ( $status[0] );
}
//Αν οι τιμές είναι λάθος τότε επιστρέφει τη τιμή fail
else { echo ("fail"); }
}
//αν αποτύχει η ανάγνωση των τιμών τότε επιστρέφει τη τιμή fail
else { echo ("fail"); }
?>

```

6.1.2 Κώδικας αρχείου index.php

```

<html>
//Ορισμός τίτλου ιστοσελίδας
<head>
    <meta charset="utf-8" />
    <title>Raspberry Pi Gpio</title>
</head>
//Ορισμός βασικού σώματος της ιστοσελίδας
<body style="background-color: black;">
    <?php
//Κώδικας PHP για τον ορισμό της ιστοσελίδας βάση των
//καταστάσεων των ακροδεκτών
//συνάρτηση για τον έλεγχο πρόσβασης μέσω smartphone ή
//tablet
function isMobile() {
    return
preg_match("/(android|avantgo|blackberry|bolt|boost|cricket|d
ocomo|fone|hiptop|mini|mobi|palm|phone|pie|tablet|up\.brows
er|up\.link|webos|wos)/i",
$_SERVER["HTTP_USER_AGENT"]);
}
//χρήση της συνάρτησης isMobile για τον έλεγχο της
//πρόσβασης μέσω smartphone ή tablet. Αν η πρόσβαση

```

```
//γίνεται μέσω υπολογιστή τότε τα κουμπιά της ιστοσελίδας  
//θα στοιχηθούν κανονικά ξεκινώντας από αριστερά λόγω  
//του μεγαλύτερου μεγέθους της οθόνης αλλά και ευκολίας.  
//Αν η πρόσβαση γίνεται μέσω smartphone ή tablet τότε τα  
//κουμπιά της ιστοσελίδας θα στοιχηθούν ένα ανά σειρά και  
//στο κέντρο της ιστοσελίδας.
```

```
if(isMobile()){  
    $var="center";  
} else {  
    $var="left";  
}
```

```
//Δημιουργία του πίνακα των καταστάσεων
```

```
$status = array (0, 0, 0, 0, 0, 0, 0, 0);
```

```
//βρόγχος για την δημιουργία του εκάστοτε κουμπιού-  
//εικονιδίου της ιστοσελίδας με βάση τη κατάσταση του  
//αντίστοιχου ακροδέκτη
```

```
for ($i = 0; $i < count($status); $i++) {
```

```
    //αρχικοποίηση του ακροδέκτη σε κατάσταση εξόδου  
    system("gpio mode ".$i." out");
```

```
    //ανάγνωση της τιμής του εκάστοτε ακροδέκτη και  
    //αποθήκευση της επιστρεφόμενης τιμής στην  
    //αντίστοιχη θέση του πίνακα καταστάσεων που  
    //δημιουργήθηκε παραπάνω
```

```
    exec ("gpio read ".$i, $status[$i], $return );
```

```
    //αν ο ακροδέκτης είναι σε κατάσταση off τότε  
    //επιστρέφει την αντίστοιχη εικόνα του ακροδέκτη σε  
    //κατάσταση off
```

```
    if ($status[$i][0] == 0 ) {
```

```
        echo ("<body align=$var><img id='button_".$i."' src='data/img/red/red_".$i.".jpg' alt='off'></body>");
```

```
    }
```

```
    //αν ο ακροδέκτης είναι σε κατάσταση on τότε  
    //επιστρέφει την αντίστοιχη εικόνα του ακροδέκτη σε  
    //κατάσταση on
```

```
    if ($status[$i][0] == 1 ) {
```

```
        echo ("<body align=$var><img id='button_".$i."'
        src='data/img/green/green_".$i.".jpg'
        alt='on'></body>");
    }
}
?>
//ορισμός θέσης αρχείου της javascript
<script src="script.js"></script>
</body>
</html>
```

6.1.3 Κώδικας αρχείου script.js

```
//Αντιστοίχιση των κουμπιών σε μεταβλητές
```

```
var button_0 = document.getElementById("button_0");
var button_1 = document.getElementById("button_1");
var button_2 = document.getElementById("button_2");
var button_3 = document.getElementById("button_3");
var button_4 = document.getElementById("button_4");
var button_5 = document.getElementById("button_5");
var button_6 = document.getElementById("button_6");
var button_7 = document.getElementById("button_7");
```

```
//με τη συνάρτηση change_pin στέλνουμε και λαμβάνουμε τον αριθμό
και τη κατάσταση των ακροδεκτών του Raspberry Pi
```

```
function change_pin (pin, status) {
```

```
//το αίτημα http
```

```
var request = new XMLHttpRequest();  
request.open( "GET" , "gpio.php?pin=" + pin + "&status=" +  
status );  
request.send(null);
```

```
//λήψη των πληροφοριών
```

```
request.onreadystatechange = function () {  
    if (request.readyState == 4 && request.status == 200) {  
        return (parseInt(request.responseText));  
    }  
}
```

```
//έλεγχος για το αν υπάρχει κάποιο σφάλμα στον server και  
//επιστροφή της τιμής fail αν όντως υπάρχει
```

```
else if (request.readyState == 4 && request.status == 500) {  
    alert ("server error");  
    return ("fail");  
}
```

```
// επιστροφή της τιμής fail αν υπάρχει κάποιο οποιοδήποτε  
//σφάλμα
```

```
else { return ("fail"); }
```

```
}  
}
```

```
//κάλυψη όλων των γεγονότων για τα κουμπιά σε περίπτωση που γίνει  
//κλικ σε κάποιο από αυτά. Αν γίνει κάποιο κλικ τότε καλεί τη
```

```
//συνάρτηση change_pin και στη συνέχεια ανανεώνει την ιστοσελίδα  
//ανάλογα με τα αποτελέσματα που θα επιστραφούν
```

```
button_0.addEventListener("click", function () {
```

```
    //αν είναι σε κατάσταση off
```

```
    if ( button_0.alt === "off" ) {
```

```
        // χρήση της συνάρτησης change_pin
```

```
        var new_status = change_pin ( 0, 0);
```

```
        //έλεγχος ύπαρξης σφάλματος
```

```
        if (new_status !== "fail") {
```

```
            //αλλαγή της κατάστασης και της εικόνας στην ιστοσελίδα
```

```
                button_0.alt = "on"
```

```
                button_0.src = "data/img/green/green_0.jpg";
```

```
                return 0;
```

```
            }
```

```
        }
```

```
    //αν είναι σε κατάσταση on
```

```
    if ( button_0.alt === "on" ) {
```

```
        // χρήση της συνάρτησης change_pin
```

```
        var new_status = change_pin ( 0, 1);
```

```
        //έλεγχος ύπαρξης σφάλματος
```

```
        if (new_status !== "fail") {
```

```
            //αλλαγή της κατάστασης και της εικόνας στην ιστοσελίδα
```

```
                button_0.alt = "off"
```



```
        button_0.src = "data/img/red/red_0.jpg";
        return 0;
    }
}

});

button_1.addEventListener("click", function () {
    //αν είναι σε κατάσταση off
    if ( button_1.alt === "off" ) {
        // χρήση της συνάρτησης change_pin
        var new_status = change_pin ( 1, 0);
        //έλεγχος ύπαρξης σφάλματος
        if (new_status !== "fail") {
            //αλλαγή της κατάστασης και της εικόνας στην ιστοσελίδα
            button_1.alt = "on"
            button_1.src = "data/img/green/green_1.jpg";
            return 0;
        }
    }

    //αν είναι σε κατάσταση on
    if ( button_1.alt === "on" ) {
        // χρήση της συνάρτησης change_pin
        var new_status = change_pin ( 1, 1);
        //έλεγχος ύπαρξης σφάλματος
```

```
if (new_status !== "fail") {  
    //αλλαγή της κατάστασης και της εικόνας στην ιστοσελίδα  
    button_1.alt = "off"  
    button_1.src = "data/img/red/red_1.jpg";  
    return 0;  
}  
}  
});  
button_2.addEventListener("click", function () {  
    //αν είναι σε κατάσταση off  
    if ( button_2.alt === "off" ) {  
        // χρήση της συνάρτησης change_pin  
        var new_status = change_pin ( 2, 0);  
        //έλεγχος ύπαρξης σφάλματος  
        if (new_status !== "fail") {  
            //αλλαγή της κατάστασης και της εικόνας στην ιστοσελίδα  
            button_2.alt = "on"  
            button_2.src = "data/img/green/green_2.jpg";  
            return 0;  
        }  
    }  
}  
//αν είναι σε κατάσταση on  
if ( button_2.alt === "on" ) {
```

```
// χρήση της συνάρτησης change_pin
var new_status = change_pin ( 2, 1);
//έλεγχος ύπαρξης σφάλματος
if (new_status !== "fail") {
//αλλαγή της κατάστασης και της εικόνας στην ιστοσελίδα
    button_2.alt = "off"
    button_2.src = "data/img/red/red_2.jpg";
    return 0;
}
}
});
button_3.addEventListener("click", function () {
//αν είναι σε κατάσταση off
if ( button_3.alt === "off" ) {
// χρήση της συνάρτησης change_pin
var new_status = change_pin ( 3, 0);
//έλεγχος ύπαρξης σφάλματος
if (new_status !== "fail") {
//αλλαγή της κατάστασης και της εικόνας στην ιστοσελίδα
    button_3.alt = "on"
    button_3.src = "data/img/green/green_3.jpg";
    return 0;
}
}
```

```
    }  
  
    //αν είναι σε κατάσταση on  
    if ( button_3.alt === "on" ) {  
        // χρήση της συνάρτησης change_pin  
        var new_status = change_pin ( 3, 1);  
        //έλεγχος ύπαρξης σφάλματος  
        if (new_status !== "fail") {  
            //αλλαγή της κατάστασης και της εικόνας στην ιστοσελίδα  
            button_3.alt = "off"  
            button_3.src = "data/img/red/red_3.jpg";  
            return 0;  
        }  
    }  
  
});  
  
button_4.addEventListener("click", function () {  
    //αν είναι σε κατάσταση off  
    if ( button_4.alt === "off" ) {  
        // χρήση της συνάρτησης change_pin  
        var new_status = change_pin ( 4, 0);  
        //έλεγχος ύπαρξης σφάλματος  
        if (new_status !== "fail") {  
            //αλλαγή της κατάστασης και της εικόνας στην ιστοσελίδα  
            button_4.alt = "on"
```

```
        button_4.src = "data/img/green/green_4.jpg";
        return 0;
    }
}

//αν είναι σε κατάσταση on
if ( button_4.alt === "on" ) {
    // χρήση της συνάρτησης change_pin
    var new_status = change_pin ( 4, 1);
    //έλεγχος ύπαρξης σφάλματος
    if (new_status !== "fail") {
        //αλλαγή της κατάστασης και της εικόνας στην ιστοσελίδα
        button_4.alt = "off"
        button_4.src = "data/img/red/red_4.jpg";
        return 0;
    }
}

});

button_5.addEventListener("click", function () {
    //αν είναι σε κατάσταση off
    if ( button_5.alt === "off" ) {
        // χρήση της συνάρτησης change_pin
        var new_status = change_pin ( 5, 0);
        //έλεγχος ύπαρξης σφάλματος
```

```
if (new_status !== "fail") {  
    //αλλαγή της κατάστασης και της εικόνας στην ιστοσελίδα  
    button_5.alt = "on"  
    button_5.src = "data/img/green/green_5.jpg";  
    return 0;  
}  
}  
  
//αν είναι σε κατάσταση on  
if ( button_5.alt === "on" ) {  
    // χρήση της συνάρτησης change_pin  
    var new_status = change_pin ( 5, 1);  
    //έλεγχος ύπαρξης σφάλματος  
    if (new_status !== "fail") {  
        //αλλαγή της κατάστασης και της εικόνας στην ιστοσελίδα  
        button_5.alt = "off"  
        button_5.src = "data/img/red/red_5.jpg";  
        return 0;  
    }  
}  
});  
button_6.addEventListener("click", function () {  
    //αν είναι σε κατάσταση off  
    if ( button_6.alt === "off" ) {
```

```
// χρήση της συνάρτησης change_pin
var new_status = change_pin ( 6, 0);

//έλεγχος ύπαρξης σφάλματος
if (new_status !== "fail") {

//αλλαγή της κατάστασης και της εικόνας στην ιστοσελίδα
    button_6.alt = "on"
    button_6.src = "data/img/green/green_6.jpg";
    return 0;
}
}

//αν είναι σε κατάσταση on
if ( button_6.alt === "on" ) {

// χρήση της συνάρτησης change_pin
var new_status = change_pin ( 6, 1);

//έλεγχος ύπαρξης σφάλματος
if (new_status !== "fail") {

//αλλαγή της κατάστασης και της εικόνας στην ιστοσελίδα
    button_6.alt = "off"
    button_6.src = "data/img/red/red_6.jpg";
    return 0;
}
}

});
```

```
button_7.addEventListener("click", function () {  
    //αν είναι σε κατάσταση off  
    if ( button_7.alt === "off" ) {  
        // χρήση της συνάρτησης change_pin  
        var new_status = change_pin ( 7, 0);  
        //έλεγχος ύπαρξης σφάλματος  
        if (new_status !== "fail") {  
            //αλλαγή της κατάστασης και της εικόνας στην ιστοσελίδα  
            button_7.alt = "on"  
            button_7.src = "data/img/green/green_7.jpg";  
            return 0;  
        }  
    }  
    //αν είναι σε κατάσταση on  
    if ( button_7.alt === "on" ) {  
        // χρήση της συνάρτησης change_pin  
        var new_status = change_pin ( 7, 1);  
        //έλεγχος ύπαρξης σφάλματος  
        if (new_status !== "fail") {  
            //αλλαγή της κατάστασης και της εικόνας στην ιστοσελίδα  
            button_7.alt = "off"  
            button_7.src = "data/img/red/red_7.jpg";  
            return 0;  
        }  
    }  
}
```


**Ανάπτυξη Web Interface για απομακρυσμένο έλεγχο
με τη χρήση του μικροελεγκτή RaspberryPi**

```
    }  
  }  
});
```

7 Βιβλιογραφία

<http://el.wikipedia.org>

<http://en.wikipedia.org>

<http://www.engineersgarage.com>

<http://ti-einai.gr/>

<http://www.pcsteps.gr>

<http://www.adafruit.com>

<https://github.com>

<http://mechwork.physics.upatras.gr/conV2-17.html>

<http://www.surecontrols.com/what-is-industrial-automation/>

http://pdplab.it.uom.gr/teaching/embedded/material/HRY_401_Embedded_Notes_CH1-4.pdf

<http://electronics.howstuffworks.com/microcontroller.htm>

<http://el.wingwit.com/Hardware/computer-drives-storage/47393.html#.VYAcjKOPSSg>

<http://www.newagepublishers.com/samplechapter/001599.pdf>

<http://www.cnet.com/uk/products/raspberry-pi-model-b-plus/>

<http://www.raspberrypi-spy.co.uk/2014/07/raspberry-pi-model-b-first-impressions/>

<http://www.raspberry-projects.com/pi/pi-hardware/raspberry-pi-model-b-plus/model-b-plus-io-pins>

<https://www.philoxenos.com/ti-ine-o-server-web-server-diakomistis-expiretitis/>

http://www.breylancommunications.com/productsupport/5/what_are_some_of_the_different_kinds_of_servers.php

http://www.webopedia.com/quick_ref/servers.asp

https://semfe.gr/files/users/376/tecnologies_diadiktyou-kefalaio10.pdf

<http://slideplayer.gr/slide/1933533/>

<http://studentguru.gr/w/tutorials/a0-javascript-javascript-html>

<http://www.w3schools.com>

<http://www.wlearn.gr>

<http://www.php.net>

http://avlab.ee.auth.gr:8080/index.php?option=com_glossary&id=70&Itemid=72

<http://dide.flo.sch.gr/Plinet/Tutorials/Tutorials-HTML.html>

<https://www.raspbian.org/FrontPage>

<http://wiringpi.com/download-and-install/>