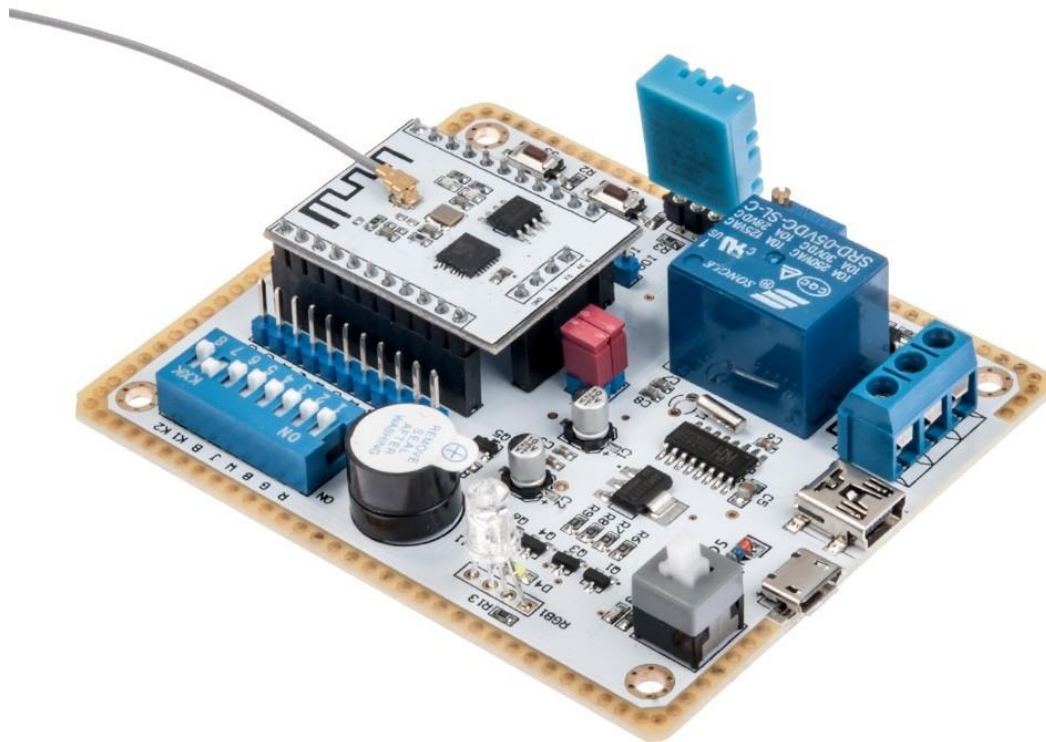


ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ ΔΥΤΙΚΗΣ ΕΛΛΑΔΑΣ
ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ
ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ Τ.Ε.



ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ 1610

ΜΕΛΕΤΗ ΤΟΥ ΜΙΚΡΟΕΛΕΓΚΤΗ ESP8266 ΚΑΙ ΕΞΟΙΚΟΝΟΜΗΣΗ ΕΝΕΡΓΕΙΑΣ ΜΕ ΤΗ ΧΡΗΣΗ ΤΟΥ ESP8266 WIFI MODULE DEVELOP BOARD

ΣΠΟΥΔΑΣΤΗΣ : ΤΕΡΖΗΣ ΑΛΕΞΙΟΣ (6539)

ΕΠΟΠΤΕΥΩΝ ΚΑΘΗΓΗΤΗΣ : ΧΡΙΣΤΟΠΟΥΛΟΣ ΚΩΝΣΤΑΝΤΙΝΟΣ

ΠΑΤΡΑ 2017

ΠΡΟΛΟΓΟΣ / ΠΕΡΙΛΗΨΗ

Σκοπός της πτυχιακής εργασίας είναι η μελέτη και ο προγραμματισμός ενός νέου, χαμηλού κόστους Wi-fi chip με πολλές TCP/IP διευθύνσεις και MCU(Micro Controller Unit), το οποίο δημιουργήθηκε από την Κινεζική Βιομηχανία και ονομάζεται ESP8266.

Στόχος μας είναι να αναλύσουμε τις λειτουργίες του ESP8266 και να τις εφαρμόσουμε σε πραγματικό περιβάλλον, έτσι ώστε να αυτοματοποιήσουμε ορισμένες διαδικασίες που μέχρι σήμερα για τον έλεγχό τους μας δαπανούσαν χρόνο και χρήμα. Επίσης, θα παρουσιάσουμε τον τρόπο με τον οποίο μπορεί να γίνει εξοικονόμηση ενέργειας, έτσι ώστε ο μικροελεγκτής να έχει μεγαλύτερη διάρκεια αυτόνομης λειτουργίας.

Ο λόγος που επιλέχτηκε το ESP8266 Wi-Fi Module Develop Board είναι ότι περιλαμβάνει όλα τα ηλεκτρονικά και ηλεκτρολογικά στοιχεία που χρειάζεται για να λειτουργήσει το ESP8266 , από τις θύρες επικοινωνίας και τροφοδοσίας με Η/Υ έως και τους αισθητήρες και τα LED που μας βοηθούν στην κατανόηση της λειτουργίας του μικροελεγκτή σε πραγματικές συνθήκες.

Τέλος θα ασχοληθούμε με το πρόγραμμα ESPlorer μέσω του οποίου θα γίνει ο προγραμματισμός του ESP8266 με αναλυτικές εικόνες, καθώς και εμφάνιση των αποτελεσμάτων του προγράμματος.

Το ESP είναι ένας νεοσύστατος μικροελεγκτής με αρκετές δυνατότητες, που συνεχώς αναπτύσσεται, διατηρώντας το μικρό κόστος και έχοντας ως βασικό στοιχείο την μικρή κατανάλωση.

ΠΕΡΙΕΧΟΜΕΝΑ

Πρόλογος / Περίληψη	iii
1. Εισαγωγή	Σελ. 1
2. Γενικές πληροφορίες μικροελεγκτών	
2.1 Ιστορία των μικροεπεξεργαστών – μικροελεγκτών.....	Σελ. 2
2.2 Η λειτουργία του μικροελεγκτή.....	Σελ. 3
2.3 Βασικά χαρακτηριστικά που καθιστούν προτιμότερο τον μικροελεγκτή. Σελ. 4	
2.4 Διαφορές μεταξύ μικροελεγκτή και μικροεπεξεργαστή.....	Σελ. 5
2.5 Διαδεδομένες κατηγορίες μικροελεγκτών.....	Σελ. 6
3. Διάφορες Μονάδες ESP8266	
3.1 Πρώτη σειρά μονάδων μικροελεγκτών ESP8266.....	Σελ. 8
3.2 NodeMCU boards.....	Σελ. 9
4. Μικροελεγκτής ESP8266	
4.1 Βασικές συνδεσμολογίες για την ενεργοποίηση του ESP8266.....	Σελ. 10
4.2 Αισθητήρια που χρησιμοποιούνται από τον μικροελεγκτή ESP8266.....	Σελ. 11
5. ESP8266 Wi-Fi Develop Board	
5.1 Παρουσίαση και ανάλυση των στοιχείων που είναι εγκατεστημένα και διασυνδεδεμένα στην πλακέτα.....	Σελ. 15
6. Προγραμματισμός με το ESPlorer	
6.1 Εργαλεία ανάπτυξης και γλώσσες προγραμματισμού.....	Σελ. 18
6.2 Γενικές πληροφορίες και εγκατάσταση του προγράμματος ESPlorer.....	Σελ. 19
6.3 Αναλυτικά βήματα για την επικοινωνία του ESP8266 με H/Y.....	Σελ. 25
6.4 Εισαγωγή και αποθήκευση κώδικα στην μνήμη του ESP8266.....	Σελ. 27
7. Εξοικονόμηση ενέργειας με τη χρήση του ESP8266	
7.1 Αυτονομία και μέθοδος εξοικονόμησης ενέργειας.....	Σελ. 29
7.2 Λειτουργία Deep Sleep και ανάλυση της κατανάλωσης ενέργειας.....	Σελ. 30
Βιβλιογραφία	Σελ. 33

ΚΕΦΑΛΑΙΟ 1

Εισαγωγή

Κάθε συσκευή, μηχανήμα ή ηλεκτρονικό σύστημα, εκτός από τον έλεγχο της σωστής και αποδοτικής λειτουργίας του, χρειάζεται επίσης χειρισμό, ρύθμιση και παρακολούθηση. Ο μικροελεγκτής είναι υπεύθυνος για την είσοδο και έξοδο, την επεξεργασία, αποθήκευση και μετάδοση των αναλογικών και ψηφιακών σημάτων μίας εφαρμογής.

Κάθε μικροελεγκτής αποτελεί ουσιαστικά έναν τύπο μικροεπεξεργαστή, ο οποίος για να καταστεί λειτουργικός χρειάζεται μονάχα ελάχιστα εξωτερικά εξαρτήματα, καθώς διαθέτει πολλά ενσωματωμένα υποσυστήματα. Οι μικροελεγκτές συναντώνται σε ποικίλες εφαρμογές, όπως για παράδειγμα σε ηλεκτρικές και ηλεκτρονικές εφαρμογές, σε αυτοματισμούς, σε κατοικίες και οχήματα.

Είναι φανερό ότι οι μικροελεγκτές και οι μικροεπεξεργαστές αποτελούν απαραίτητο μέρος των σύγχρονων τεχνολογικών εφαρμογών. Ωστόσο, οι δύο αυτές συσκευές διαφέρουν μεταξύ τους σε αρκετά σημεία. Μία βασική τους διαφορά, και η πιο σημαντική, βρίσκεται στη λειτουργικότητά τους. Προκειμένου να λειτουργήσει ένας μικροεπεξεργαστής, θα πρέπει να συνδεθεί και με άλλες συσκευές, όπως η μνήμη ή μία συσκευή αποστολής και λήψης δεδομένων. Αυτό σημαίνει ότι ένας μικροεπεξεργαστής είναι η καρδιά του συστήματος. Αντίθετα, ένας μικροελεγκτής σχεδιάζεται με τέτοιο τρόπο, ώστε να περιέχει όλες τις παραπάνω συσκευές. Συνεπώς, δε χρειάζονται άλλες συσκευές για τη λειτουργία του, εφόσον όλα τα απαραίτητα περιφερειακά είναι ενσωματωμένα στη δομή του. Με τον τρόπο αυτό, εξοικονομείται χώρος και χρόνος, κατά την κατασκευή ενός συστήματος που βασίζεται σε μικροελεγκτή.

Πέρα από τον χώρο και τον χρόνο που εξοικονομείται με τη χρήση μικροελεγκτή στις διάφορες εφαρμογές, μεγάλο ενδιαφέρον δείχνει και η εξοικονόμηση ενέργειας. Φυσικά συγκρίνονται μικροποσότητες καταναλώσεων, οι οποίες όμως σε εξειδικευμένες εφαρμογές έχουν βαρύνουσα σημασία.

ΚΕΦΑΛΑΙΟ 2

Γενικές πληροφορίες μικροελεγκτών

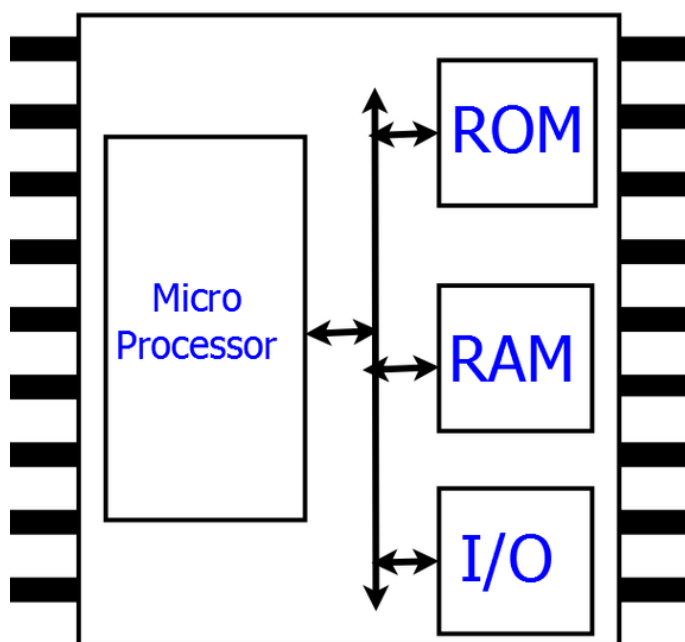
2.1 Ιστορία των μικροεπεξεργαστών – μικροελεγκτών

Η επιθυμία των κατασκευαστικών επιχειρήσεων 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, η οποία χρησιμοποιήθηκε για την κατασκευή των κυκλωμάτων μικροελεγκτή αυτών είχε σαν αποτέλεσμα την ανάπτυξη του μικροεπεξεργαστή.

Οι μικροεπεξεργαστές είναι υπεύθυνοι για την έμπνευση και τη δημιουργία μερικών από τις μεγαλύτερες καινοτομίες στα συστήματα υπολογιστών. Αυτές οι καινοτομίες περιλαμβάνουν τους embedded μικροελεγκτές, τους προσωπικούς υπολογιστές, τους σύγχρονους σταθμούς εργασίας, συσκευές χειρός και κινητές συσκευές (επεξεργαστές κινητών τηλεφώνων), servers (εξυπηρετητές) εφαρμογών και αρχείων, web servers για το internet, υπερυπολογιστές χαμηλού κόστους και ευρείας κλίμακας δίκτυα υπολογιστών. Οι μικροεπεξεργαστές είναι επεξεργαστές συνόλου εντολών (instruction set processors, ISPs). Ένας ISP εκτελεί εντολές ενός προκαθορισμένου συνόλου εντολών. Η λειτουργικότητά του εξαρτάται από το σύνολο των εντολών που είναι ικανός να εκτελέσει ο μικροεπεξεργαστής. Σε αυτό το σύνολο των εντολών κωδικοποιούνται όλα τα προγράμματα που τρέχουν σε έναν μικροεπεξεργαστή. Αυτό το προκαθορισμένο σύνολο εντολών ονομάζεται επίσης αρχιτεκτονική συνόλου εντολών (instruction set architecture, ISA). Το ISA χρησιμεύει ως μία διασύνδεση ανάμεσα στα προγράμματα και τους επεξεργαστές.

Ο όρος «μικροελεγκτής» αναφέρεται σε έναν τύπο επεξεργαστή, ο οποίος μπορεί να λειτουργήσει με ελάχιστα εξωτερικά εξαρτήματα, λόγω των πολλών ενσωματωμένων υποσυστημάτων που διαθέτει. Για τον λόγο αυτό θεωρείται παραλλαγή ενός μικροεπεξεργαστή. Ένας μικροελεγκτής είναι ένα ενσωματωμένο τσιπ (ολοκληρωμένο κύκλωμα) που αποτελεί συχνά μέρος ενός συστήματος. Όπως και ένας απλός τυπικός υπολογιστής, έτσι και ο μικροελεγκτής περιλαμβάνει CPU, RAM, ROM για αποθήκευση δεδομένων και λογισμικού αντίστοιχα, μνήμη flash για μόνιμη αποθήκευση, θύρες εισόδου/εξόδου, μετατροπέα αναλογικού σε ψηφιακό σήμα και το αντίστροφο και τέλος timers. Καθώς είναι σχεδιασμένος να εκτελεί μόνο μία συγκεκριμένη εργασία για τον έλεγχο ενός απλού συστήματος, είναι πολύ μικρότερος και απλούστερα σχεδιασμένος ώστε να μπορεί να περιλαμβάνει όλες τις λειτουργίες, οι οποίες απαιτούνται για ένα μόνο ολοκληρωμένο κύκλωμα. Χρησιμοποιείται ευρέως σε όλα τα ενσωματωμένα συστήματα ελέγχου χαμηλού και μεσαίου κόστους όπως για παράδειγμα αυτά που χρησιμοποιούνται σε αυτοματισμούς, ηλεκτρονικά καταναλωτικά προϊόντα, ηλεκτρικές συσκευές και κάθε είδους αυτοκινούμενα τροχοφόρα οχήματα. Σήμερα, η χρήση μικροελεγκτών είναι καθολική, με την έννοια ότι κάθε προϊόν που αλληλεπιδρά με ένα χρήστη περιλαμβάνει ένα μικροελεγκτή, ο οποίος παίζει το ρόλο του «εγκεφάλου» των ηλεκτρονικών κυκλωμάτων.

Microcontroller



Σχήμα 2.1 : Δομή μικροελεγκτή

2.2 Η λειτουργία του μικροελεγκτή

Υπάρχουν πολλοί μικροελεγκτές και ακόμα περισσότερα προγράμματα που διατίθενται για μικροελεγκτές. Εντούτοις, οι περισσότεροι από αυτούς έχουν πολλές ομοιότητες. Έτσι, αν μπορέσουμε να μάθουμε να χειριζόμαστε έναν από αυτούς τότε θα μπορούμε να χειριστούμε και άλλους επίσης. Ο μικροελεγκτής είναι μια γρήγορη συσκευή (όχι όσο ο υπολογιστής), με αποτέλεσμα κάθε εντολή που εκτελείται σε αυτόν να γίνεται με πολύ υψηλή ταχύτητα. Η λειτουργία του δίνεται παρακάτω:

Όταν ενεργοποιείται η τροφοδοσία, το Μητρώο Λογικού Ελέγχου ενεργοποιεί με τη σειρά του τον ταλαντωτή χαλαζία. Στα πρώτα λίγα χιλιοστά του δευτερολέπτου και ενώ οι πρώτες προετοιμασίες βρίσκονται σε εξέλιξη, οι πυκνωτές παράσιτα φορτίζονται. Όταν η στάθμη της τάσης φτάνει στη μέγιστη τιμή και η συχνότητα του ταλαντωτή χαλαζία γίνεται σταθερή, η διαδικασία της γραφής bits σε ειδικά λειτουργικά μητρώα (SFrs) ξεκινάει. Τα πάντα λαμβάνουν χώρα σύμφωνα με το ρολόι του ταλαντωτή και πάνω από όλα τα ηλεκτρονικά αρχίζουν να δουλεύουν. Όλα αυτά γίνονται σε λίγα νάνο δευτερόλεπτα. Ο υπολογιστής ή ο μετρητής προγράμματος μηδενίζει τη διεύθυνση της μνήμης προγράμματος. Στη συνέχεια, η διεύθυνση αποστέλλει τις οδηγίες στον αποκωδικοποιητή, ο οποίος αποκωδικοποιεί τις εντολές και έτσι τις εκτελεί. Μετά την εκτέλεση μιας εντολής, η διεύθυνση του απαριθμητή προγράμματος αυξάνεται κατά 1 και ως εκ τούτου στέλνει τη διεύθυνση της επόμενης εντολής στον αποκωδικοποιητή εντολών και εκτελεί τις επόμενες οδηγίες.

2.3 Βασικά χαρακτηριστικά που καθιστούν προτιμότερο τον μικροελεγκτή

Το «πακέτο» ενός μικροελεγκτή φέρει κάποια βασικά χαρακτηριστικά που τον καθιστά προτιμότερο για τη χρήση του σε εφαρμογές έναντι της χρήσης των επιμέρους στοιχείων που τον απαρτίζουν ξεχωριστά (επεξεργαστής, μνήμες, συσκευές εισόδου-εξόδου, διεπαφές). Μερικά από αυτά τα χαρακτηριστικά:

- **Χαμηλό κόστος:** είναι ένα από τα βασικότερα χαρακτηριστικά που κάποιος σχεδιαστής λαμβάνει υπόψη. Η συνεχής απελευθέρωση στην αγορά μικροελεγκτών από διάφορες εταιρίες βελτίωσαν την ποιότητα αυτών και μείωσαν τις τιμές λόγω του ανταγωνισμού.
- **Μικρότερο μέγεθος:** η ολοκλήρωση των βασικών στοιχείων από τα οποία απαρτίζεται μείωσε τις διαστάσεις σε σχέση με τη χρήση των επιμέρους στοιχείων ως σύνολο.
- **Χαμηλή κατανάλωση ισχύος:** το γεγονός ότι οι μικροελεγκτές λειτουργούν σε συγκριτικά χαμηλές συχνότητες που φτάνουν τα 32 KHz, οδηγεί στην κατανάλωση μικρών ποσών ισχύος της τάξης των mW ακόμα και μ W. Επιπλέον, έχουν τη δυνατότητα να εισέρχονται σε κατάσταση αναμονής –sleep mode- καταστέλλοντας προσωρινά τη λειτουργία της κεντρικής μονάδας επεξεργασίας (CPU) και των περιφερειακών, με αποτέλεσμα πολύ μεγάλη μείωση της κατανάλωσης ισχύος του μικροελεγκτή. Έτσι μπορούν να χρησιμοποιηθούν σε εφαρμογές με αυστηρές απαιτήσεις ως προς αυτήν την παράμετρο.
- **Αυτονομία:** αυτό επιτυγχάνεται μέσω της ενσωμάτωσης σύνθετων περιφερειακών υποσυστημάτων όπως μνήμες και θύρες επικοινωνίας. Έτσι πολλοί μικροελεγκτές δεν χρειάζονται κανένα άλλο ολοκληρωμένο κύκλωμα για να λειτουργήσουν.
- **Επίτευξη ελέγχου ή μετρήσεων σε πραγματικό χρόνο:** ενώ οι ηλεκτρονικοί υπολογιστές πρέπει να τρέχουν λειτουργικά συστήματα πραγματικού χρόνου (όπως RT-Linux, QNX κ.ά.) για να το επιτύχουν, οι μικροελεγκτές δεν απαιτούν επιπλέον λογισμικό.
- **Μειωμένες εκπομπές ηλεκτρομαγνητικών παρεμβολών:** 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 των χαμηλότερων ταχυτήτων λειτουργίας.
- **Περισσότεροι διαθέσιμοι ακροδέκτες** για ψηφιακές εισόδους-εξόδους (για δεδομένο μέγεθος ολοκληρωμένου κυκλώματος) λόγω της μη δέσμευσής τους για τη σύνδεση εξωτερικών περιφερειακών.
- Η βασική αρχιτεκτονική των μικροελεγκτών δεν διαφέρει από αυτή των κοινών μικροεπεξεργαστών, αν και στους πρώτους συναντάται συχνά η αρχιτεκτονική μνήμης τύπου Harvard, η οποία χρησιμοποιεί διαφορετικές αρτηρίες σύνδεσης της μνήμης προγράμματος και της μνήμης δεδομένων (για παράδειγμα οι σειρές από την Microchip). Στους κοινούς μικροεπεξεργαστές συνηθίζεται η ενιαία διάταξη μνήμης τύπου φον Νόιμαν.

- Η ενσωμάτωση περιφερειακών σημαίνει ευκολότερη υλοποίηση εφαρμογών λόγω των απλούστερων διασυνδέσεων. Επίσης, υπάρχει μεγαλύτερη αξιοπιστία λόγω των λιγότερων διασυνδέσεων και μικρό μέγεθος συνολικού υπολογιστικού συστήματος.

Παρόλο αυτά κάποια από τα μειονεκτήματα του μικροελεγκτή είναι:

- Η μη αλλαγή του προγράμματος για τον λόγο ότι είναι γραμμένο στην ROM
- Έχει μεγάλο χρόνο ανάπτυξης. Για να ολοκληρωθεί ένα προϊόν μπορεί να απαιτηθεί από μία εβδομάδα μέχρι έναν χρόνο
- Η δυσκολία του προγραμματισμού του

2.4 Διαφορά μεταξύ μικροελεγκτή και μικροεπεξεργαστή

Ο μικροεπεξεργαστής είναι ένα ολοκληρωμένο κύκλωμα (IC) που έχει μόνο τη CPU στο εσωτερικό του, δηλαδή μόνο τις εξουσίες επεξεργασίας, όπως για παράδειγμα ο Pentium I,II,III,IV της Intel. Αυτοί οι μικροεπεξεργαστές δεν έχουν μνήμη RAM, ROM και άλλα περιφερειακά στο τσιπ. Για να γίνει λειτουργικός ένας μικροεπεξεργαστής, ο σχεδιαστής του συστήματος πρέπει να τα προσθέσει στο εξωτερικό του. Εφαρμογές του μικροεπεξεργαστή περιλαμβάνουν οι επιτραπέζιοι ηλεκτρονικοί υπολογιστές, οι φορητοί υπολογιστές, τα σημειωματάρια. Όσον αφορά τους μικροελεγκτές, έχουν CPU επιπροσθέτως με ένα σταθερό ποσό της μνήμης RAM, ROM και άλλα περιφερειακά, όλα ενσωματωμένα σε ένα μόνο chip. Σήμερα διαφορετικοί κατασκευαστές παράγουν μικροελεγκτές με ένα ευρύ φάσμα δυνατοτήτων και είναι διαθέσιμοι σε διαφορετικές εκδόσεις. Ορισμένοι κατασκευαστές είναι οι ATMEL, Microchip, Philips, Motorola.

Οι μικροελεγκτές είναι σχεδιασμένοι για να εκτελούν συγκεκριμένες εργασίες. Αυτό σημαίνει εφαρμογές που η σχέση των εισροών και εκροών είναι καθορισμένες. Ανάλογα με την εισροή, πρέπει να γίνει η διεργασία ώστε να υπάρξει εκροή. Για παράδειγμα, πληκτρολόγιο, ποντίκια, πλυντήριο ρούχων, φούρνος μικροκυμάτων, αυτοκίνητα, ποδήλατα, τηλέφωνο, κινητά τηλέφωνα. Δεδομένου ότι οι εφαρμογές είναι πολύ συγκεκριμένες, χρειάζονται μικροί πόροι, όπως μνήμη RAM, ROM, θύρες I/O και ως εκ τούτου να μπορούν να ενσωματωθούν σε ένα μόνο chip. Αυτό με τη σειρά του μειώνει το μέγεθος και το κόστος. Οι μικροεπεξεργαστές βρίσκουν εφαρμογή σε αόριστα καθήκοντα όπως η ανάπτυξη λογισμικού, παιχνίδια, ιστοσελίδες, επεξεργασία φωτογραφιών, δημιουργία εγγράφων. Σε τέτοιες περιπτώσεις, η σχέση μεταξύ εισόδου και εξόδου δεν ορίζεται. Χρειάζονται υψηλό ποσό πόρων, όπως RAM, ROM, θύρες I/O. Η ταχύτητα ρολογιού του μικροεπεξεργαστή είναι αρκετά υψηλή σε σύγκριση με του μικροελεγκτή. Ενώ οι μικροελεγκτές λειτουργούν από μερικά MHz έως 30 με 50 MHz, οι σημερινοί μικροεπεξεργαστές λειτουργούν πάνω από 1 GHz, δεδομένου ότι εκτελούν πολύπλοκα καθήκοντα.

Η σύγκριση του μικροελεγκτή και του μικροεπεξεργαστή όσον αφορά το κόστος δεν είναι δικαιολογημένη. Αναμφίβολα ένας μικροελεγκτής είναι πολύ φθηνότερος από ένα μικροεπεξεργαστή. Ωστόσο, ένας μικροελεγκτής δεν μπορεί να χρησιμοποιηθεί στη θέση ενός μικροεπεξεργαστή. Από την άλλη, ο μικροεπεξεργαστής δεν μπορεί να χρησιμοποιηθεί αυτόνομα. Χρειάζεται άλλα περιφερειακά όπως RAM, ROM, buffer, θύρες I/O κλπ και ως εκ τούτου ένα σύστημα σχεδιασμένο γύρω από έναν μικροεπεξεργαστή, είναι αρκετά δαπανηρό.

2.5 Διαδεδομένες κατηγορίες μικροελεγκτών

Λόγω της τάσης ενσωμάτωσης των μικροελεγκτών σε κάθε ηλεκτρική και ηλεκτρονική συσκευή αλλά και του ισχυρότατου ανταγωνισμού, είναι προτιμότερη η παραγωγή ανταγωνιστικών μοντέλων μαζικής παραγωγής καθώς και η παραγωγή μικροελεγκτών για πιο εξειδικευμένες εφαρμογές. Διακρίνονται οι εξής κατηγορίες:

- Μικροελεγκτές (καμιά φορά 4-bit αλλά συνήθως 8-bit) πολύ χαμηλού κόστους, γενικής χρήσης, με πολύ μικρό αριθμό ακροδεκτών (ακόμη και λιγότερους από 8). Για να μη μπορεί να αντιγραφεί εύκολα το εσωτερικό λογισμικό τους σχεδιάζονται με έμφαση στη χαμηλή κατανάλωση ισχύος και την αυτάρκεια. Απουσιάζει η δυνατότητα επέκτασης της μνήμης τους. Μερικά μοντέλα είναι ευρέως γνωστά στους ερασιτέχνες ηλεκτρονικούς, όπως για παράδειγμα οι περισσότεροι μικροελεγκτές των σειρών PIC (Microchip), AVR (Atmel) και 8051 (Intel, Atmel, Dallas).

- Μικροελεγκτές (συνήθως 8-bit αλλά και 16 ή 32-bit) χαμηλού κόστους, γενικής χρήσης με μέτριο έως σχετικά μεγάλο αριθμό ακροδεκτών. Διαθέτουν μεγάλο αριθμό κοινών περιφερειακών, όπως θύρες UART, I²C, SPI ή CAN, μετατροπείς αναλογικού σε ψηφιακό και ψηφιακού σε αναλογικό. Στους κατασκευαστές της Άπω Ανατολής (Ιαπωνία, Κορέα), συνηθίζεται η ενσωμάτωση ελεγκτών οθόνης υγρών κρυστάλλων και πληκτρολογίου. Μερικές φορές παρέχουν δυνατότητα εξωτερικής επέκτασης μνήμης τους.

- Μικροελεγκτές (κυρίως 32-bit) μέσου κόστους, γενικής χρήσης, με μεγάλο αριθμό ακροδεκτών. Χαρακτηρίζονται από έμφαση στην ταχύτητα εκτέλεσης εντολών, υψηλή αυτάρκεια περιφερειακών και μεγάλες δυνατότητες εσωτερικής ή εξωτερικής μνήμης προγράμματος (FLASH) και RAM. Στο χώρο αυτό έχουν ισχυρή παρουσία οι αρχιτεκτονικές με υψηλή μεταφερσιμότητα λογισμικού (portability) από τον ένα στον άλλο κατασκευαστή. Για παράδειγμα, μεταξύ των μικροελεγκτών τύπου ARM ή MIPS, το σύνολο των βασικών εντολών που αναγνωρίζει η ALU είναι ακριβώς το ίδιο, μειώνοντας έτσι τις μεγάλες αλλαγές στο λογισμικό, όταν στο μέλλον ο πελάτης υιοθετήσει ένα μικροελεγκτή άλλου κατασκευαστή (αρκεί, φυσικά, να υποστηρίζει κι αυτός το σύνολο εντολών ARM ή MIPS, αντίστοιχα).

- Μικροελεγκτές εξειδικευμένων εφαρμογών, οι οποίοι ενσωματώνουν συνήθως κάποιο εξειδικευμένο πρωτόκολλο επικοινωνίας, το οποίο υλοποιείται πάντοτε σε hardware. Τέτοιοι μικροελεγκτές χρησιμοποιούνται σε τηλεπικοινωνιακές συσκευές όπως τα modem.

Η μεγάλη μερίδα πωλήσεων των μικροελεγκτών εξακολουθεί να αφορά αυτούς των 8-bit, καθώς είναι η κατηγορία με το χαμηλότερο κόστος και το μικρότερο μέγεθος λογισμικού για το ίδιο αποτέλεσμα, ιδίως επειδή οι σύγχρονες οικογένειες μικροελεγκτών 8-bit έχουν πολύ βελτιωμένες επιδόσεις σε σχέση με το παρελθόν.

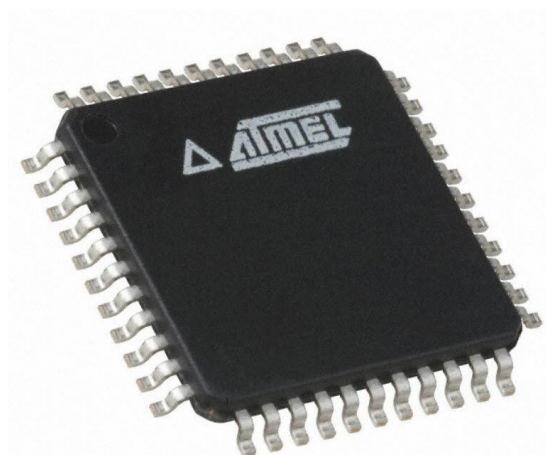
Κατασκευαστές Μικροελεγκτών

Οι περισσότερες εταιρίες παράγουν μεγάλη γκάμα μικροελεγκτών. Από πολύ μικρούς και φθηνούς για απλές εφαρμογές έως ιδιαίτερα προηγμένους για πολύ απαιτητικές εφαρμογές. Μερικοί από τους γνωστότερους κατασκευαστές μικροελεγκτών είναι οι εξής:

- ARM
- Atmel
- Epson
- Hitachi
- Maxim
- Microchip
- NEC
- Toshiba
- Intel
- Analog Devices
- Espressif Systems
- Freescale Semiconductor
- Texas Instruments



Σχήμα 2.2 : Μικροελεγκτής εταιρίας NEC



Σχήμα 2.3 : Μικροελεγκτής εταιρίας Atmel

ΚΕΦΑΛΑΙΟ 3

Διάφορες Μονάδες ESP8266

3.1 Πρώτη σειρά μικροελεγκτών ESP8266

Η πρώτη σειρά από Wi-Fi μικροελεγκτές που δημιουργήθηκε από δυτικούς κατασκευαστές και συγκεκριμένα από την Ai-Thinker, ονομάζεται ESP8266 και παραμένει ως σήμερα η πιο ευρέως διαθέσιμη. Αναφέρονται συλλογικά ως ESP-xx μονάδες (ESP-201, ESP-01, ESP-12F) και για να σχηματίσουν ένα λειτουργικό σύστημα ανάπτυξης, απαιτούν επιπλέον εξαρτήματα. Ειδικότερα χρειάζεται ένας TTL-to-USB προσαρμογέας (συνήθως ονομάζεται μία γέφυρα USB-to-UART) και ένα εξωτερικό τροφοδοτικό που αποδίδει τάση 3,3 Volt.

Ένας αρχάριος προγραμματιστής προτιμεί να εξετάσει ένα ESP8266 Develop Board, όπως το NodeMCU που θα δούμε και στην επόμενη παράγραφο, το οποίο έχει ενσωματωμένα τη γέφυρα USB-to-UART και ένα βύσμα micro-USB σε συνδυασμό με ένα ρυθμιστή ισχύος 3,3 Volt. Σε μία απλή μονάδα ESP8266, όταν ολοκληρωθεί το έργο δεν χρειάζονται αυτά τα επιπλέον στοιχεία με αποτέλεσμα το ESP8266 να είναι οικονομικότερο και ενεργειακά αποδοτικότερο αφού θα καταναλώνει μικρότερη ισχύ.



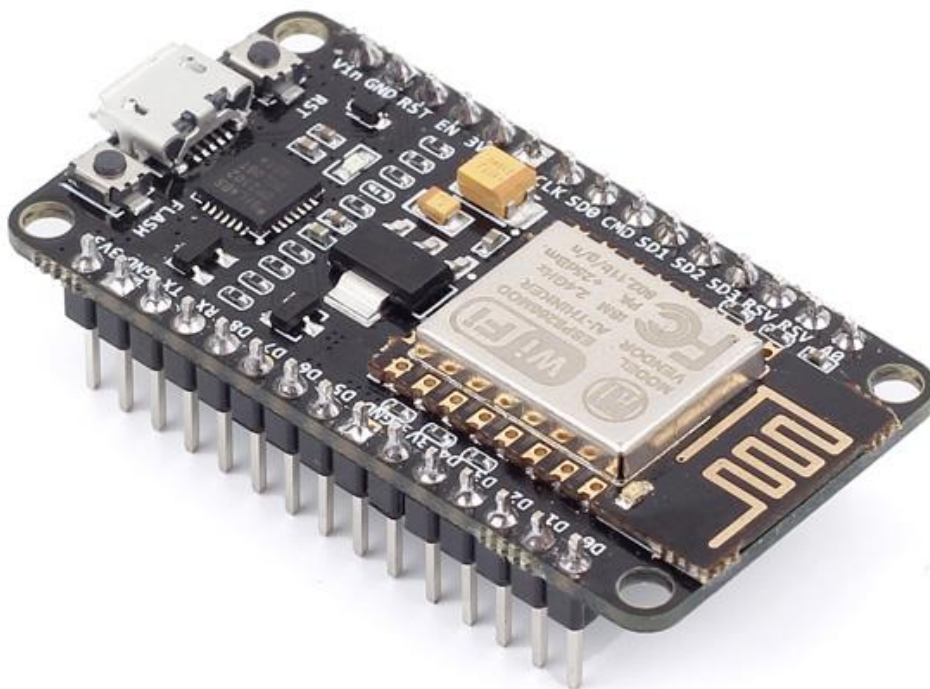
Σχήμα 3.1 : Μονάδα ESP – 12F

3.2 NodeMCU Boards

Οι μονάδες NodeMCU είναι περισσότερο δημοφιλής σε αρχάριους χρήστες διότι έχουν ενσωματωμένα πάνω σε μία πλακέτα όλα τα ηλεκτρονικά στοιχεία που χρειάζεται ο μικροελεγκτής ESP8266 για να ενεργοποιηθεί και να προγραμματιστεί. Αυτά είναι η γέφυρα USB-to-UART και ένα βύσμα Micro-USB σε συνδυασμό με ένα ρυθμιστή τάσης 3,3V τόσο για παροχή ενέργειας στην πλακέτα, όσο και για την δυνατότητα διασύνδεσης του μικροελεγκτή με τον Η/Υ μέσω του κατάλληλου λογισμικού.

Πριν από την δημιουργία αυτών των ολοκληρωμένων μονάδων αυτά τα στοιχεία έπρεπε να αγοραστούν χωριστά και έπειτα να συνδεθούν με το ESP8266. Πλέον οι συγχρονες μονάδες ESP8266 όπως είναι οι NodeMCU Boards, είναι λιγότερο επώδυνες για την κατασκευή τους και προσφέρουν περισσότερες ακίδες GPIO για μεγαλύτερο εύρος λειτουργιών.

Οι NodeMCU Boards στο μεγαλύτερο κομμάτι τους βασίζονται στο ESP-12E όμως σταδιακά αναπτύσσονται και νέες μονάδες με διαφορετικά μοντέλα ESP8266.



Σχήμα 3.2 : Μονάδα NodeMCU

ΚΕΦΑΛΑΙΟ 4

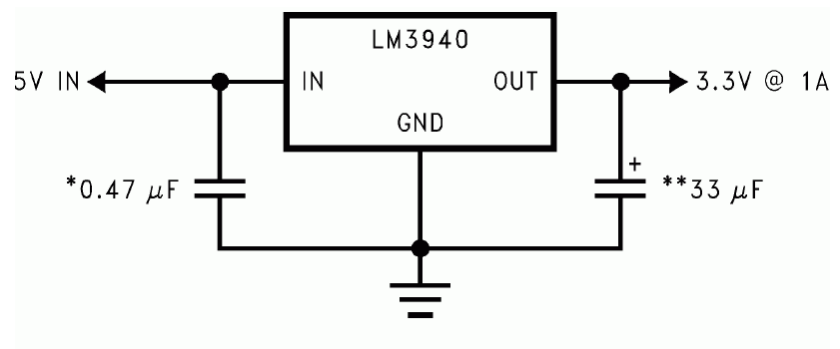
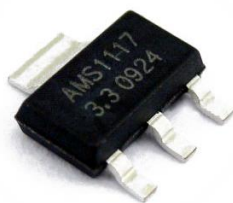
Μικροελεγκτής ESP8266

4.1 Βασικές συνδεσμολογίες για την ενεργοποίηση του ESP8266

Όπως κάθε ηλεκτρονικό εξάρτημα, έτσι και ο μικροελεγκτής ESP8266 προαπαιτεί για την έναρξη της λειτουργίας του μια σειρά διαδικασιών, ώστε να ενεργοποιηθεί σωστά και να λειτουργήσει ομαλά. Με αυτόν τον τρόπο αποτρέπεται και το ενδεχόμενο παρουσίασης βλάβης ή ακόμα και απώλειας ενός ηλεκτρονικού στοιχείου μέσα στον μικροελεγκτή, η οποία θα οδηγούσε στην καταστροφή όλης της μονάδας.

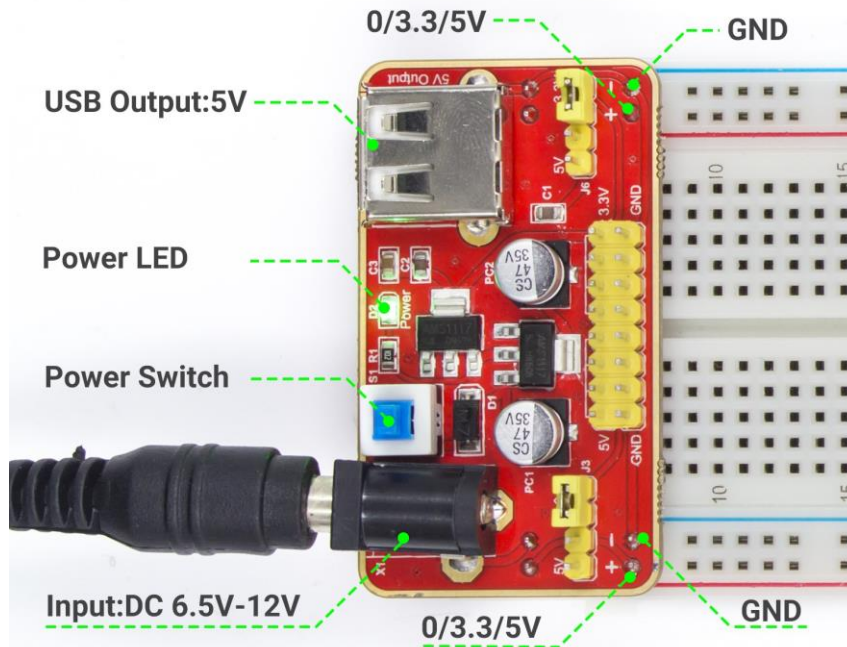
Οι δύο βασικές συνδεσμολογίες για την ενεργοποίηση του ESP8266 είναι η τροφοδοσία της μονάδας και η επικοινωνία του με τον ηλεκτρονικό υπολογιστή για την αποστολή και παραλαβή δεδομένων προγραμματισμού ή παρακολούθησης αποτελεσμάτων.

Όλες οι μονάδες ESP8266 εργάζονται με τάση 3.3 Volt. Αυτό το μέγεθος τάσης δεν χρησιμοποιείται συχνά από «φιλικές» συσκευές διασύνδεσης του ESP8266 όπως ένας Η/Υ μέσω μιας θύρας USB. Για το λόγο αυτό χρησιμοποιείται ένας ρυθμιστής τάσης ο οποίος δέχεται τάση εισόδου 5 Volt και ως έξοδο έχει την επιθυμητή τάση 3.3 Volt.



Σχήμα 4.1 : Ρυθμιστής Τάσης 5V to 3.3V
Σχήμα 4.2 : Voltage Regulator 5V to 3.3 V

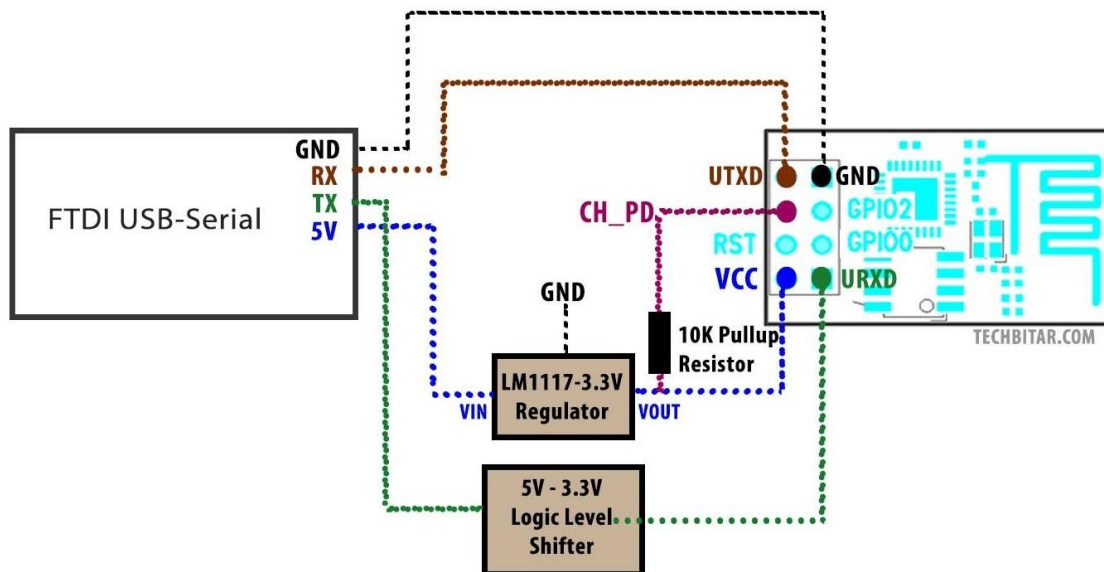
Ένας ακόμη τρόπος για να τροφοδοτηθεί το ESP8266 είναι τα τροφοδοτικά για breadboard τα οποία έχουν ως είσοδο από 6.5 έως 12 Volts και εξόδους 3.3 και 5 Volt.



**Σχήμα 4.3 : Τροφοδοτικό Breadboard
Breadboard power supply**

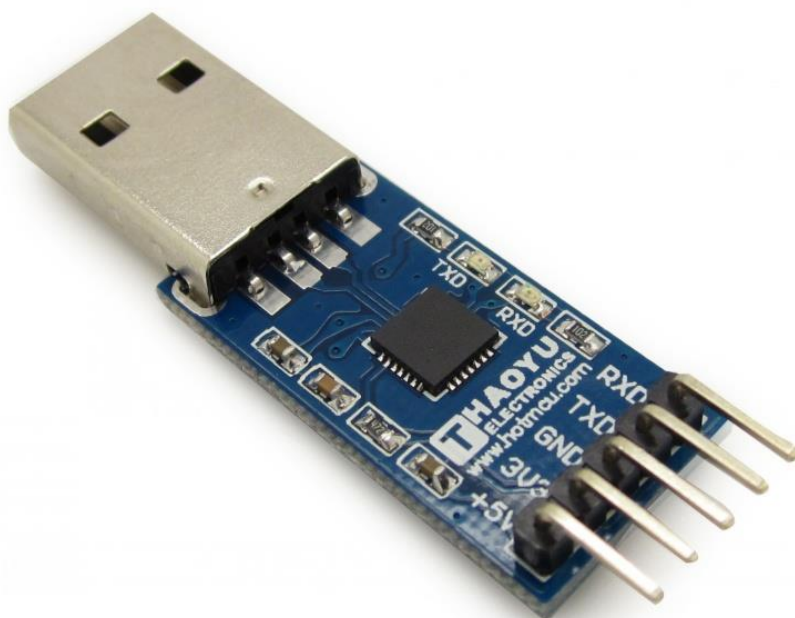
Η δεύτερη βασική συνδεσμολογία που πρέπει να ληφθεί υπόψη κατά την ενεργοποίηση του ESP8266 είναι η επικοινωνία του με τον Η/Υ. Αυτή η διαδικασία είναι βασική διότι οποιαδήποτε ενέργεια πραγματοποιήσει ο μικροελεγκτής είναι αποτέλεσμα των δεδομένων που έχουν μεταφερθεί από τον χρήστη μέσω του κατάλληλου λογισμικού, στην μνήμη του μικροελεγκτή.

Ο δημοφιλέστερος τρόπος για να γίνει μεταφορά δεδομένων από και προς το ESP8266 είναι μία FTDI συσκευή. Σε αυτή την περίπτωση χρειάζεται προσοχή στην σύνδεση μεταξύ εισόδου (Receiver) και εξόδου (Transmitter) του μικροελεγκτή και του Η/Υ αντίστοιχα.



Σχήμα 4.4 : FTDI USB Serial

Ένας ακόμη τρόπος διασύνδεσης για μεταφορά δεδομένων είναι ο μετατροπέας USB to UART. Σε αυτή την περίπτωση η μεταφορά δεδομένων τροποποιείται μέσω του chip CH340G το οποίο επικοινωνεί με την θύρα USB του Η/Υ.



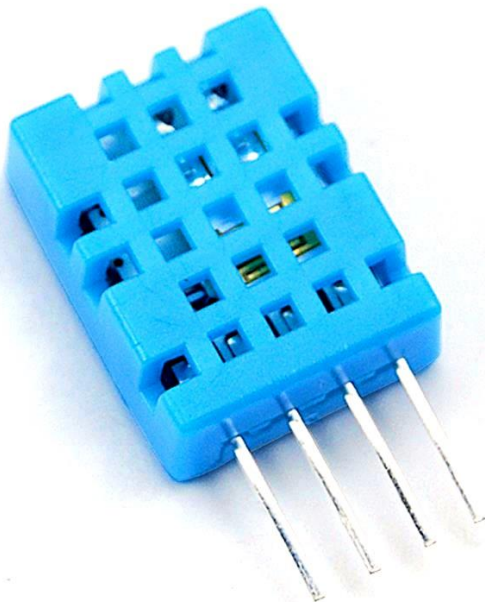
Σχήμα 4.5 : Μετατροπέας USB to UART

4.2 Αισθητήρια που χρησιμοποιούνται από τον μικροελεγκτή ESP8266

Όπως σε όλους τους μικροελεγκτές έτσι και ο ESP8266 δέχεται στις εισόδους του διάφορα αισθητήρια που άλλοτε στόχο έχουν να ενημερώνουν απλά το χρήστη και άλλοτε ανάλογα με το μέγεθος της μέτρησης, ο μικροελεγκτής να λαμβάνει εντολή για την εκτέλεση μίας ενέργειας. Υπάρχουν διάφοροι τύποι αισθητήρων για την κάλυψη των απαιτήσεων που δημιουργούνται στην συνεχή εξελισσόμενη αγορά. Οι πιο δημοφιλείς αισθητήρες που χρησιμοποιούνται από τους χρήστες του ESP8266 είναι οι παρακάτω:

- Αισθητήρας Θερμοκρασίας – Υγρασίας DHT11

DHT-11



Ο DHT-11 διαθέτει ένα συγκρότημα αισθητήρων θερμοκρασίας και υγρασίας με ένα βαθμονομημένο ψηφιακό σήμα εξόδου. Με την αποκλειστική χρήση της ψηφιακής τεχνικής για την απόκτηση του σήματος, ο αισθητήρας εξασφαλίζει υψηλή αξιοπιστία και εξαιρετική μακροχρόνια σταθερότητα. Ο DHT-11 περιλαμβάνει ένα στοιχείο τύπου αντίστασης για την μέτρηση της υγρασίας και ένα NTC στοιχείο για την μέτρηση της θερμοκρασίας, όπου συνδέονται με έναν υψηλής απόδοσης μικροελεγκτή 8-bit προσφέροντας άριστη ποιότητα, γρήγορη απόκριση, ικανότητα καταπολέμησης παρεμβολών και καλή σχέση κόστους-αποτελεσματικότητας.

- Αισθητήρας κίνησης - PIR Sensor



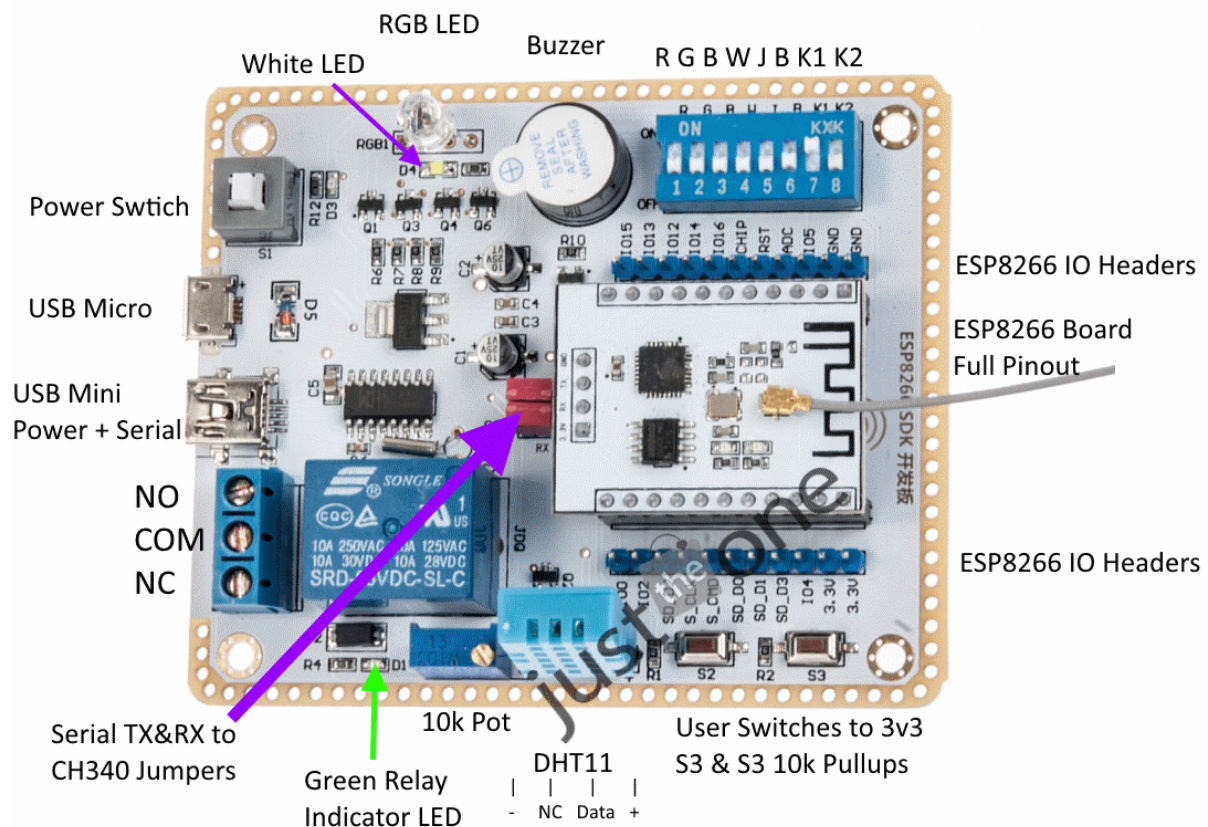
Οι αισθητήρες PIR επιτρέπουν την «αίσθηση» της κίνησης. Σχεδόν πάντα χρησιμοποιείται για να ανιχνεύει αν ένας άνθρωπος έχει κινηθεί μέσα ή έξω από την περιοχή των αισθητήρων. Έχουν αρκετά πλεονεκτήματα όπως ότι είναι μικροί σε μέγεθος, φθηνοί, χαμηλής κατανάλωσης ισχύος, εύκολοι στην χρήση και δεν φθείρονται. Για το λόγο αυτό συνήθως βρίσκονται σε συσκευές και gadgets που χρησιμοποιούνται σε σπίτια ή επιχειρήσεις. Συχνά αναφέρονται ως PIR (Passive Infrared) ή αισθητήρες κίνησης IR.

ΚΕΦΑΛΑΙΟ 5

ESP8266 Wi-Fi Develop Board

5.1 Παρουσίαση και ανάλυση των στοιχείων που είναι εγκατεστημένα και διασυνδεδεμένα στην πλακέτα

Στις παρακάτω εικόνες απεικονίζονται η πλακέτα και όλα τα στοιχεία που την περιβάλλουν καθώς και ένα διάγραμμα το οποίο είναι απαραίτητο για να κατανοήσουμε την επικοινωνία των στοιχείων με τον μικροεπεξεργαστή.



DIP Switch Function

<http://scottsnowden.co.uk>

- R - RGB LED Red to pin 'IO15' (must be on to boot to application mode)
- G - RGB LED Green to pin 'IO13'
- B - RGB LED Blue to pin 'IO12'
- W - White LED to pin 'IO14'
- J - Relay coil to pin 'IO16'
- B - Buzzer to pin 'IO5'
- K1 - GND to pin 'IO15' (same as Red RGB LED)
- K2 - GND to pin 'IO0' (Same as S2 - ON to reprogram)

Σχήμα 5.1 : Μονάδα ESP8266 Wi-Fi Develop Board

Αφετηρία της παραπάνω μονάδας είναι ο κεντρικός διακόπτης S1 ο οποίος τροφοδοτεί όλο το σύστημα. Εάν δεν είναι ενεργοποιημένος δεν λειτουργεί κανένα στοιχείο πάνω στην πλακέτα. Αφού λοιπόν ο διακόπτης ενεργοποιηθεί τότε το σύστημα τροφοδοτείται είτε από την θύρα USB Micro είτε από την θύρα USB Mini, με την δεύτερη να έχει και την ευθύνη για την μεταφορά δεδομένων από και προς τον μικροελεγκτή.

Άλλα στοιχεία που είναι τοποθετημένα στην πλακέτα είναι ένα RGB Led, ένα λευκό Led και ένα Buzzer. Αυτά βοηθούν το χρήστη να δοκιμάσει και να κατανοήσει το αποτέλεσμα μιας σειράς εντολών και να το επαληθεύσει οπτικά (Led's) και ακουστικά (buzzer) πριν το εφαρμόσει σε πραγματικές συνθήκες.

Επίσης στην πλακέτα υπάρχει μία σειρά από διακόπτες και μπουτόν οι οποίοι είναι συνδεδεμένοι με θύρες του ESP8266, με Led και με το GND. Η χρησιμότητά τους είναι μεγάλη διότι βοηθούν κυρίως τους αρχάριους χρήστες αφενός για το αρχικό στάδιο του προγραμματισμού και αφετέρου στο να μην συνωστίζονται όλες οι ενδείξεις σε ενδεχόμενο λάθος. Με την χρήση των διακοπών ο χρήστης ενεργοποιεί και απενεργοποιεί χειροκίνητα τα στοιχεία που θέλει, έως ότου εξοικειωθεί με το προγραμματιστικό περιβάλλον.

Άλλο ένα σημαντικό εξάρτημα που βρίσκεται στην πλακέτα είναι το Single Relay το οποίο είναι ένας μικρός ηλεκτρονόμος. Με αυτό ο χρήστης μπορεί να ελέγξει φορτίο με τάση έως 250 V και φορτίο 10 A. Αυτό σημαίνει ότι μπορεί εύκολα να ελέγξει ένα φορτίο στο σπίτι χωρίς προσθήκη άλλων στοιχείων. Διαθέτει μία NO και μια NC επαφή ενώ για τον έλεγχο του απαιτείται τάση 5V την οποία λαμβάνει απευθείας από την πλακέτα και κατ' επέκταση από την θύρα USB του H/Y.

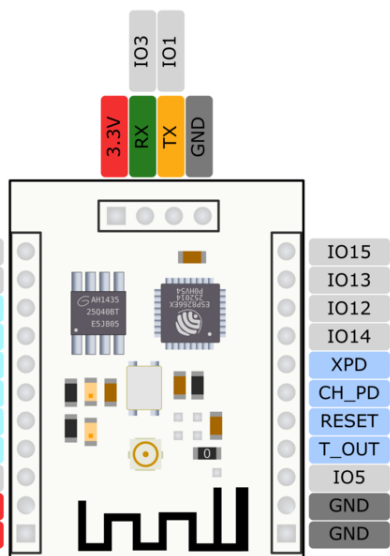
Τέλος υπάρχουν διάφορα μικρά ηλεκτρονικά στοιχεία όπως αντιστάσεις, πυκνωτές, ρυθμιστές τάσης για την εύρυθμη λειτουργία του συστήματος και μία βάση στην οποία εγκαθίσταται το ESP-201. Στα πλαϊνά της βάσης υπάρχουν pins για όλες τις εισόδους και εξόδους του ESP8266, για ενδεχόμενη σύνδεσή τους με το κατάλληλο μέσο.

ESP8266 MODULE ESP-201 CHEAT SHEET

- ESP8266 FEATURES:**
- 32 BIT CPU @ 80MHZ
 - 64 KB COMMAND RAM
 - 96 KB DATA RAM
 - EXTERNAL QPI FLASH USUALLY 512KB UP TO 4MBIT
 - IEEE 802.11 B/G/N WIFI 2.4GHZ, WEP/WPA/WPA2
 - UP TO 16 GPIO PINS
 - SPI, I²C, I²S, UART
 - 10-BIT ADC

- ESP-201 FEATURES:**
- 2x11 PINS @ 2.54MM
 - BREADBOARD-FRIENDLY
 - INTEGRATED ANTENNA (MANUALLY ACTIVATED)
 - U.FL EXTERNAL ANTENNA CONNECTOR

IO0	IO2
IO9	D2
IO6	CLK
IO11	CMD
IO7	D0
IO8	D1
IO10	D3
IO4	IO4
3.3V	3.3V

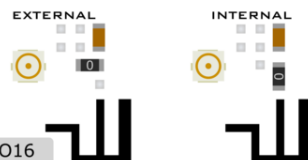


OPERATING

VCC: 3.3V (UP TO 200MA)
IO AND UART ARE NOT 5V TOLERANT
CH_PD (AKA ENABLE) MUST BE PULLED HIGH TO OPERATE

ANTENNA SELECTION

ESP-201 ARE PRECONFIGURED TO USE AN EXTERNAL ANTENNA. TO USE THE PCB-ANTENNA ROTATE THE 0-ΩM RESISTOR BY 90 DEGREES



ADDITIONAL GPIO

CYAN PINS ARE USED TO COMMUNICATE WITH THE SPI FLASH - IF YOU RELOCATE PINS 3 AND 7 OF THE SPI FLASH TO VCC AND USE D10 INSTEAD OF Q10 PINS D2 AND D3 CAN BE USED AS ADDITIONAL GPIO

BOOT MODES

PINS MUST BE PULLED TO THE APPROPRIATE LEVEL DURING POWERUP

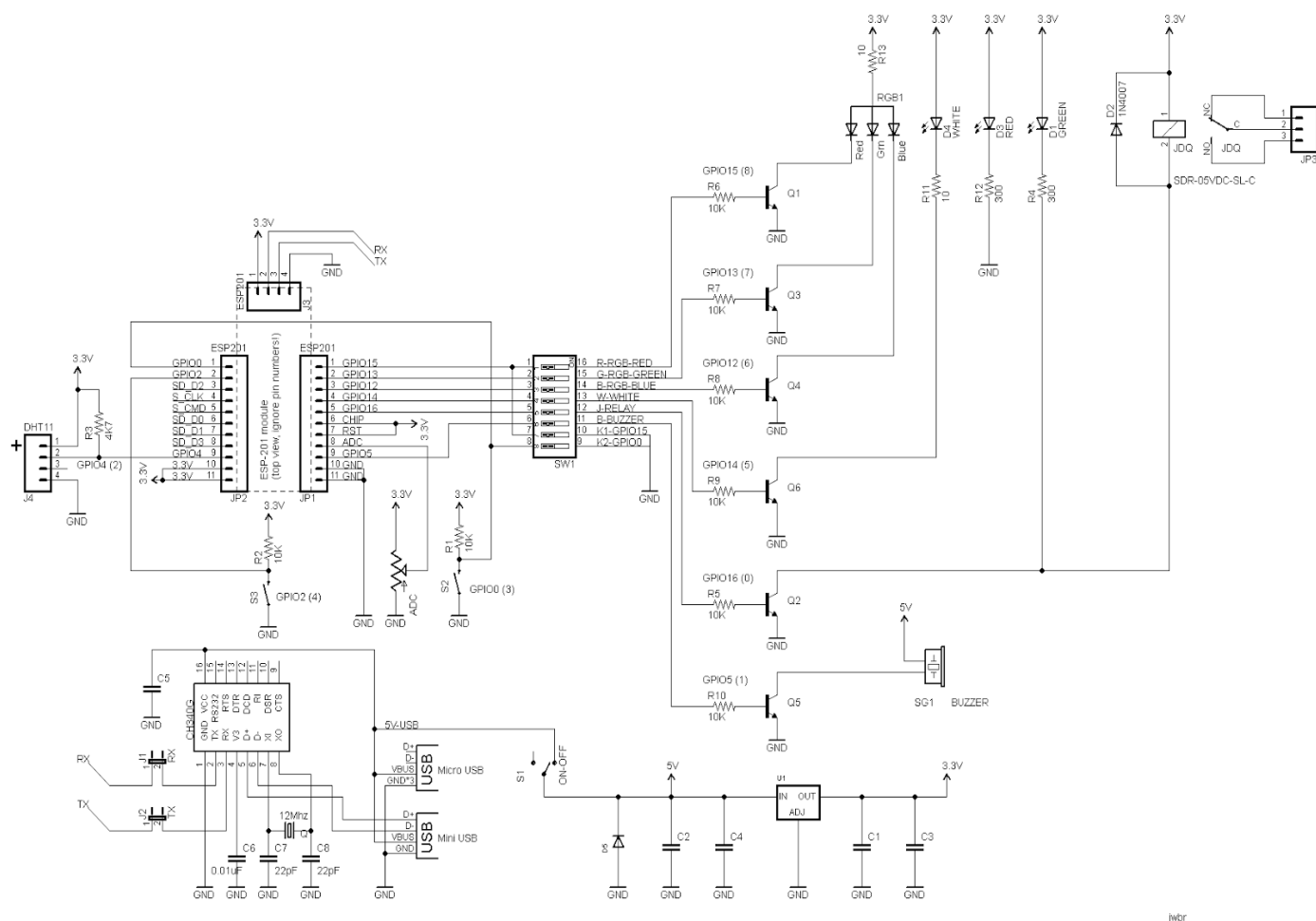
GP1015	GP109	GP102	Mode
LOW	LOW	HIGH	Serial Programming
LOW	HIGH	HIGH	Boot from Flash
HIGH	ANY	ANY	Boot from SD-Card

Σχήμα 5.2 : Τεχνικά χαρακτηριστικά ESP-201

Ο μικροελεγκτής που χρησιμοποιείται στην παραπάνω πλακέτα είναι ο ESP-201. Στο πάνω μέρος υπάρχουν 4 ακροδέκτες από τους οποίους οι δύο χρησιμοποιούνται για την τροφοδοσία του μικροελεγκτή και οι άλλοι δύο για την αποστολή (TX) και παραλαβή (RX) δεδομένων. Στα πλαϊνά υπάρχουν εισοδοί, έξοδοι, αναλογική είσοδος, Reset και pin που μας δίνουν 3,3 Volts και GND για να τροφοδοτήσουμε κάποιο επιπλέον στοιχείο.

Επιπλέον, η κεραία για την ασύρματη επικοινωνία με το router μπορεί να διαφέρει ανάλογα με την έκδοση του μικροελεγκτή και να είναι εσωτερική ή εξωτερική.

Το ESP-201 αποτελείται από 32-Bit CPU με ταχύτητα 80 MHz , 64 KB Command RAM και 96 KB Data RAM



Σχήμα 5.3 : Διάγραμμα διασύνδεσης στοιχείων στην πλακέτα ESP8266 WiFi Develop Board

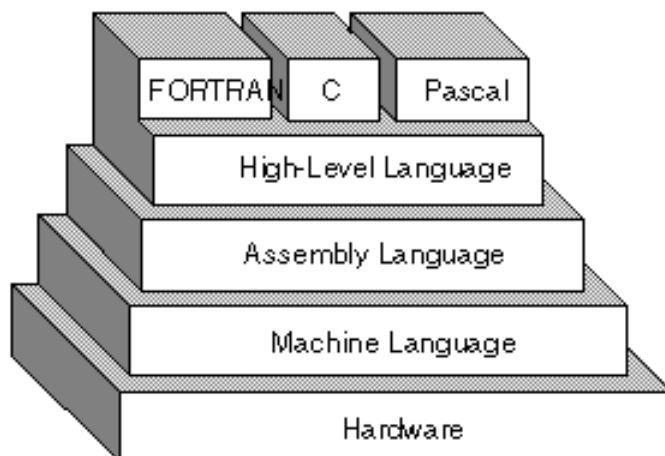
ΚΕΦΑΛΑΙΟ 6

Προγραμματισμός με το ESPlorer

6.1 Εργαλεία ανάπτυξης και γλώσσες προγραμματισμού

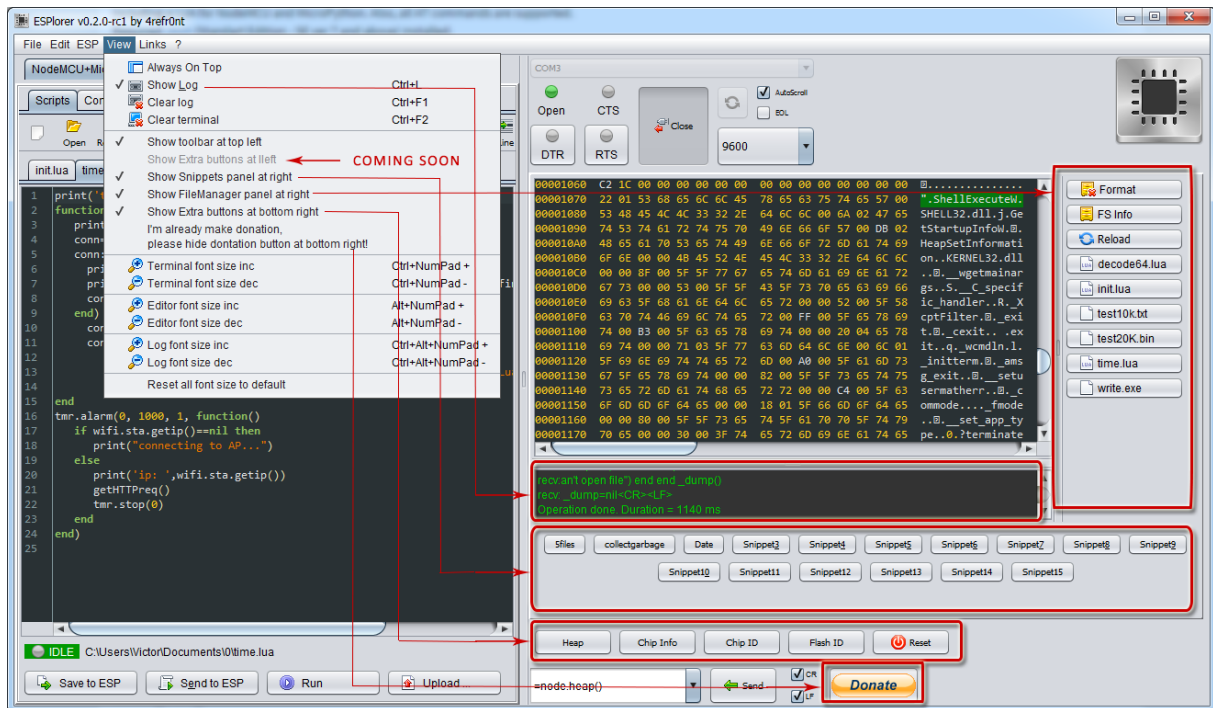
Η επιτυχία καθορίζεται σε μεγάλο βαθμό από τη διαθεσιμότητα και την ευχρηστία των σχετικών εργαλείων ανάπτυξης, όπως μεταφραστές από γλώσσες υψηλού επιπέδου σε γλώσσα κατανοητή από τον μικροελεγκτή (Assembly), προγραμματιστές της εσωτερικής μνήμης και εργαλεία εκσφαλμάτωσης (Debugger). Στους μικροελεγκτές, τα εργαλεία αυτά δεν αποτελούνται ποτέ μόνο από το λογισμικό, καθώς δεν υπάρχει τυποποιημένος τρόπος επικοινωνίας με αυτούς. Στον τομέα των εργαλείων ανάπτυξης, δραστηριοποιούνται όχι μόνο οι ίδιοι οι κατασκευαστές μικροελεγκτών αλλά και εξειδικευμένες εταιρίες.

Η C, η C++ και οι παραλλαγές τους είναι από τις πιο διαδεδομένες γλώσσες προγραμματισμού των μικροελεγκτών. Η Assembly μπορεί να χρησιμοποιείται σε τμήματα του λογισμικού όπου απαιτείται ταχύτητα ή μικρό μέγεθος χρησιμοποιούμενης μνήμης. Όμως οι μεγαλύτερες απαιτήσεις σε λειτουργικότητα και η ευκολία προγραμματισμού της C έναντι της Assembly, σε συνδυασμό με την επάρκεια μνήμης των σύγχρονων μικροελεγκτών, έχουν εκτοπίσει την Assembly από τις περισσότερες εφαρμογές.



Σχήμα 6.1 : Γλώσσες ανάπτυξης προγραμματισμού

6.2 Γενικές πληροφορίες και εγκατάσταση του προγράμματος ESPlorer



Σχήμα 6.1 : Πλήρης εικόνα του προγράμματος ESPlorer

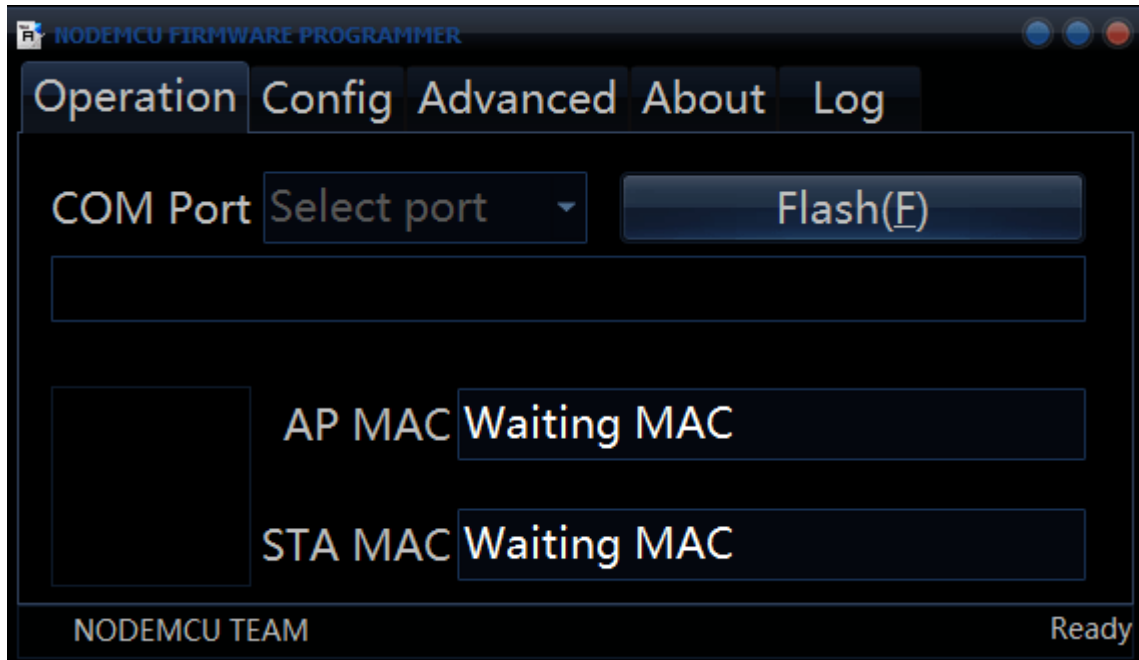
Το ESPlorer είναι ένα IDE (Ολοκληρωμένο περιβάλλον ανάπτυξης) για ESP προγραμματιστές. Είναι multiplatform, που σημαίνει ότι είναι συμβατό με διαφορετικά λειτουργικά συστήματα όπως Windows , Mac OS X ή Linux. Το ESPlorer δημιουργήθηκε σε Java και απαιτείται η εγκατάσταση της έκδοσης SE 7 και άνω για την σωστή λειτουργία του. Το ESPlorer είναι βασικό εργαλείο για οποιονδήποτε προγραμματιστή ESP8266. Το πρόγραμμα περιλαμβάνει την γλώσσα LUA για NodeMCU και MicroPython. Ο προγραμματιστής μπορεί να αναπτύξει πρόγραμμα στο LUA Editor που είναι ένα λογισμικό ανοιχτού κώδικα και είναι εύκολο στη χρήση. Επίσης υποστηρίζει όλες τις εντολές AT (AT commands).

Για την εγκατάσταση του ESPlorer στον Η/Υ απαιτείται:

- Η λήψη και εγκατάσταση του λογισμικού JAVA 7 version και άνω
- Με σκοπό την ολοκλήρωση της διαδικασίας του προγραμματισμού, πρέπει να εκτελεστεί μια ενέργεια που ονομάζεται Flash ESP82266 με το NodeMCU firmware.

Το NodeMCU είναι ένα firmware που επιτρέπει στον χρήστη να προγραμματίσει τις μονάδες ESP8266 με LUA script. Αυτός ο τρόπος είναι παρόμοιος με τον τρόπο που προγραμματίζεται ένα arduino. Με λίγες μόνο γραμμές κώδικα ο χρήστης μπορεί να δημιουργήσει μία σύνδεση Wi-Fi, να ελέγξει τις θύρες GPIO's του ESP8266, να μετατρέψει τη μονάδα σε ένα web server και πολλά άλλα.

Μετά την καλωδίωση του κυκλώματος και κατεβάζοντας το αρχείο NodeMCU Flasher μπορεί να ξεκινήσει η διαδικασία του Flashing. Ανοίγοντας την συντόμευση θα εμφανιστεί το παρακάτω παράθυρο.



Σχήμα 6.2 : Περιβάλλον για το Flashing του ESP8266

Τροποποιώντας μερικές ρυθμίσεις όπως την θύρα Port και παραμέτρους στην καρτέλα για προχωρημένους (Advanced) και πατώντας το κουμπί Flash ξεκινά η διαδικασία. Μετά την ολοκλήρωση αυτής της διαδικασίας θα πρέπει να εμφανίζεται ένας πράσινος κύκλος με ένα εικονίδιο ελέγχου.

Έπειτα, κατεβάζοντας το πρόγραμμα ESPlorer ο χρήστης βρίσκει το συμπιεσμένο φάκελο με το αρχείο ESPlorer.zip όπου εκεί περιέχονται και τα υπόλοιπα αρχεία που χρειάζονται για την εκκίνηση. Στην παρακάτω εικόνα φαίνεται το αρχείο ESPlorer.jar που επιλέγεται.

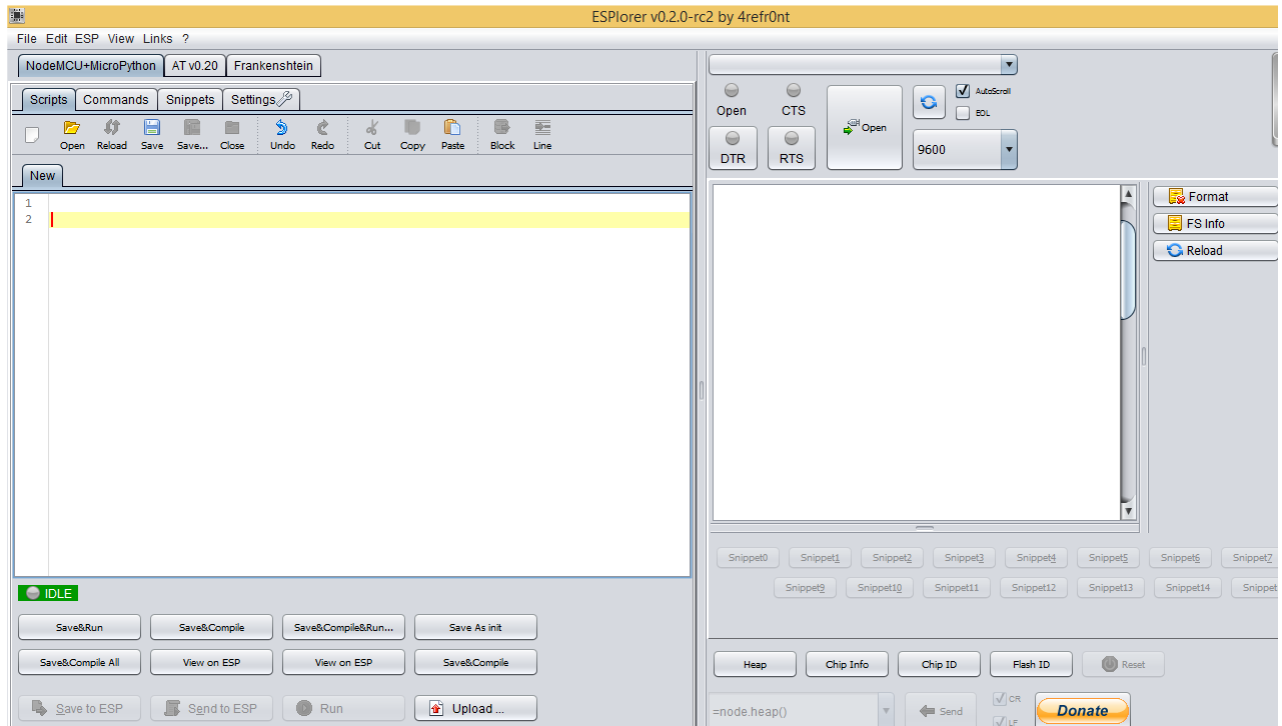
Name	Date modified	Type	Size
_lua	23/03/2015 14:02	File folder	
lib	23/03/2015 13:57	File folder	
ESPlorer	15/12/2014 23:49	Windows Batch File	1 KB
ESPlorer	26/04/2015 19:47	Executable Jar File	2 097 KB
version	26/04/2015 20:11	Text Document	1 KB

Σχήμα 6.3 : Αρχείο εκκίνησης του ESPlorer

Εκτελώντας το αρχείο αυτό, μετά από μερικά δευτερόλεπτα ανοίγει το περιβάλλον του ESPlorer IDE το οποίο είναι αυτό που χρειάζεται κάθε φορά για να προγραμματίσεις το ESP8266.

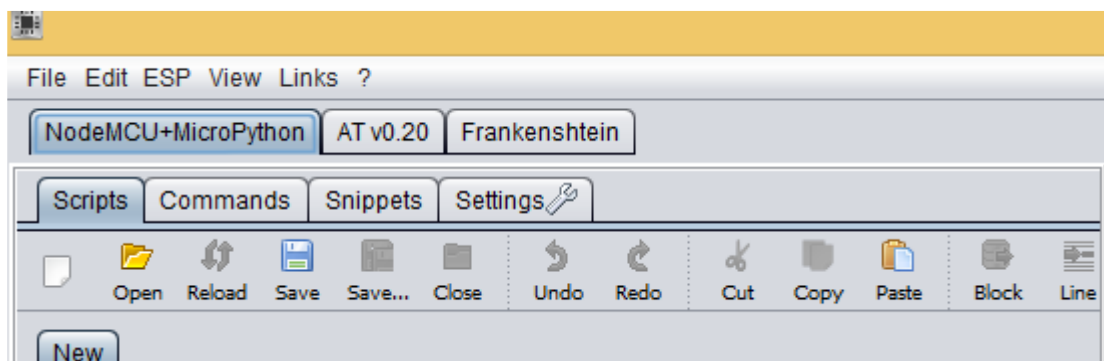
Σημείωση : Αν το λειτουργικό σύστημα είναι Mac OS X ή Linux μπορεί απλά να χρησιμοποιηθεί η γραμμή εντολών `sudo- java - jar ESPlorer.jar` στο τερματικό, ώστε να τρέξει το πρόγραμμα.

Ανοίγοντας την εφαρμογή παρουσιάζεται στην οθόνη η παρακάτω εικόνα.



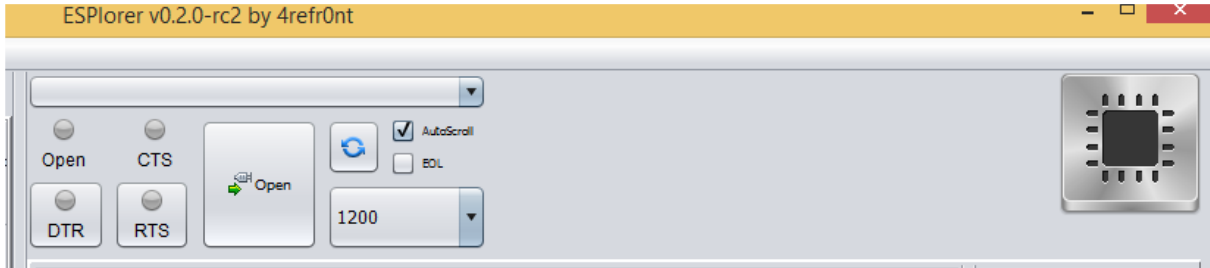
Σχήμα 6.5 : Αρχική οθόνη μετά την εκκίνηση ESPlorer

Το ESPlorer IDE έχει ένα ζευγάρι δύο κύριων τμημάτων. Στην πάνω αριστερά γωνία υπάρχουν όλες οι επιλογές που χρησιμοποιούνται τακτικά σε πολλά τα λογισμικά. Κάποιες από αυτές είναι η δημιουργία νέου αρχείου, η αποθήκευση του αρχείου, αποθήκευση ως, επανάληψη, αναίρεση και άλλα.



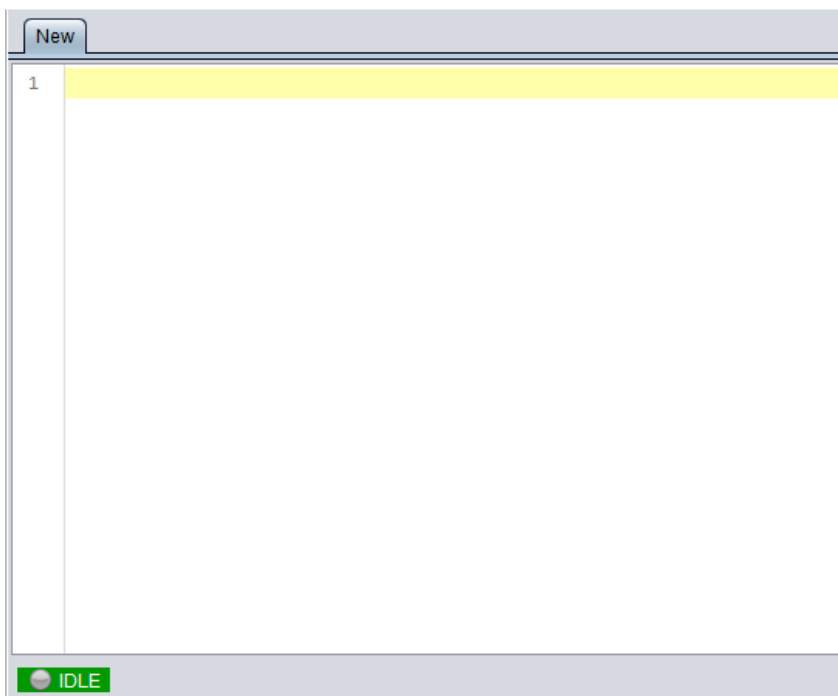
Σχήμα 6.6 : Κύρια τμήματα προγράμματος ESPlorer

Στην επάνω αριστερά γωνία υπάρχουν όλες οι επιλογές που χρειάζεται ο χρήστης για να δημιουργήσει μία σειριακή επικοινωνία. Το θέμα αυτό θα αναλυθεί πιο κάτω.



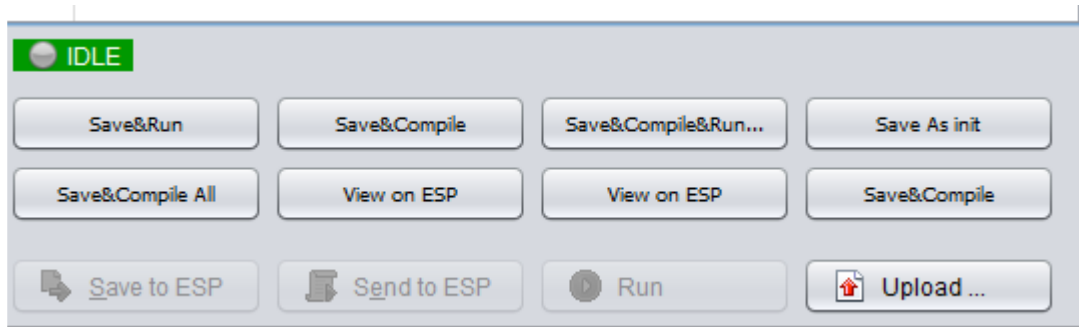
Σχήμα 6.7 : Επιλογές για την σειριακή επικοινωνία με το ESP8266

Στην παρακάτω εικόνα φαίνεται το πλαίσιο που γράφονται τα σενάρια (scripts) δηλαδή ο κώδικας με τις κατάλληλες εντολές.



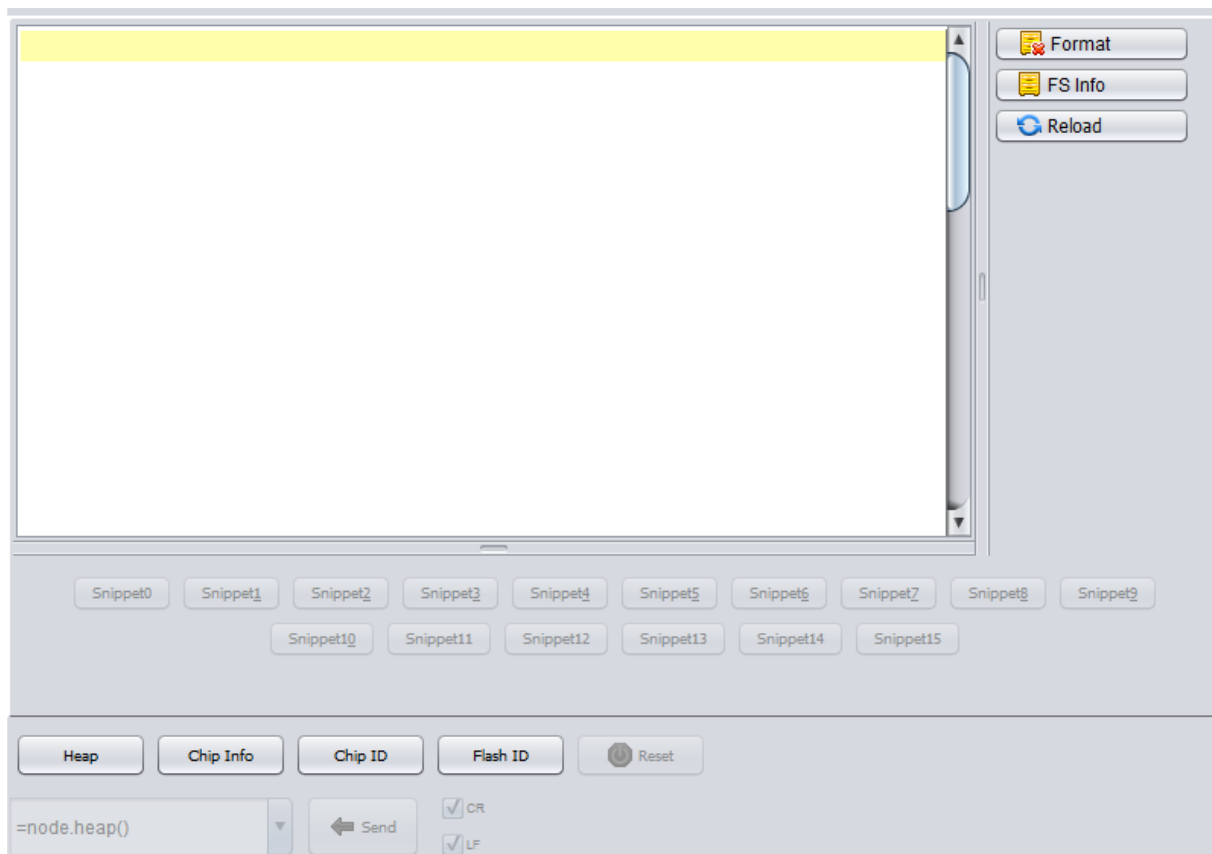
Σχήμα 6.8 : Οθόνη γραφής κώδικα

Κάτω από το παράθυρο του κώδικα υπάρχουν δώδεκα κουμπιά που προσφέρουν όλες τις αναγκαίες λειτουργίες που θα χρειαστεί να αλληλεπιδράσουν με το ESP8266. Οι δύο εντολές που χρησιμοποιούνται περισσότερο είναι η αποθήκευση στο ESP (Save to ESP) και η αποστολή στο ESP (Send to ESP).



Σχήμα 6.9 : Λειτουργίες και εντολές που αλληλεπιδρούν με το ESP8266

Τέλος, στην παρακάτω εικόνα φαίνεται το παράθυρο εξόδου που παρουσιάζει τι ακριβώς συμβαίνει στο ESP8266. Εκεί ο χρήστης μπορεί να δει τυχόν λάθη που υπάρχουν στον κώδικα και να εκτυπώσει το σενάριο ώστε να διορθώσει το project.

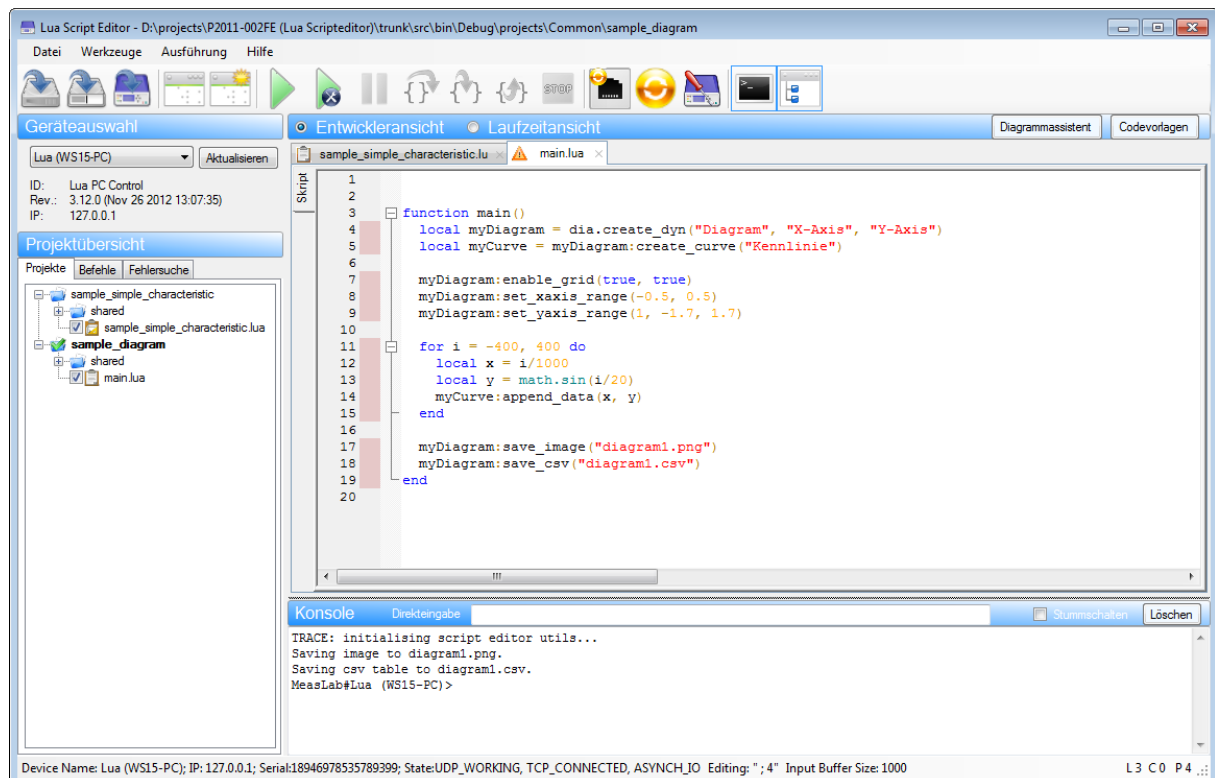


Σχήμα 6.10 : Παράθυρο εξόδου ESPlorer

Στις παρακάτω εικόνες φαίνεται ένα ενδεικτικό LUA script, που σκοπό έχει να αναβοσβήσει ένα LED. Έτσι γίνεται κατανοητή η μορφή ενός σεναρίου και ο κώδικας που περιέχεται. Έτοιμα scripts βρίσκονται εύκολα σε ιστοσελίδες όπως το Git Hub.

```
init.lua
1 lighton=0
2 pin=4
3 gpio.mode(pin,gpio.OUTPUT)
4 tmr.alarm(1,2000,1,function()
5     if lighton==0 then
6         lighton=1
7         gpio.write(pin,gpio.HIGH)
8     else
9         lighton=0
10        gpio.write(pin,gpio.LOW)
11    end
12 end)
```

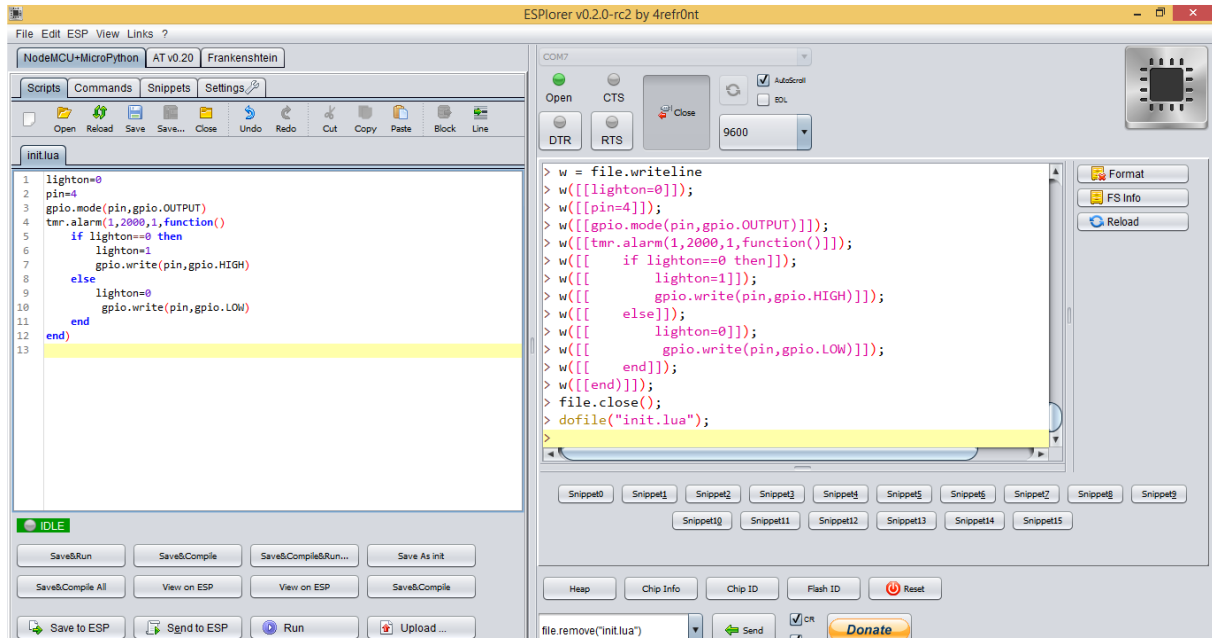
Σχήμα 6.11 : Παράδειγμα κώδικα που αναβοσβήνει ένα LED



Σχήμα 6.10 : Editor για την δημιουργία αρχείων script

6.3 Αναλυτικά βήματα για την επικοινωνία του ESP8266 με Η/Υ

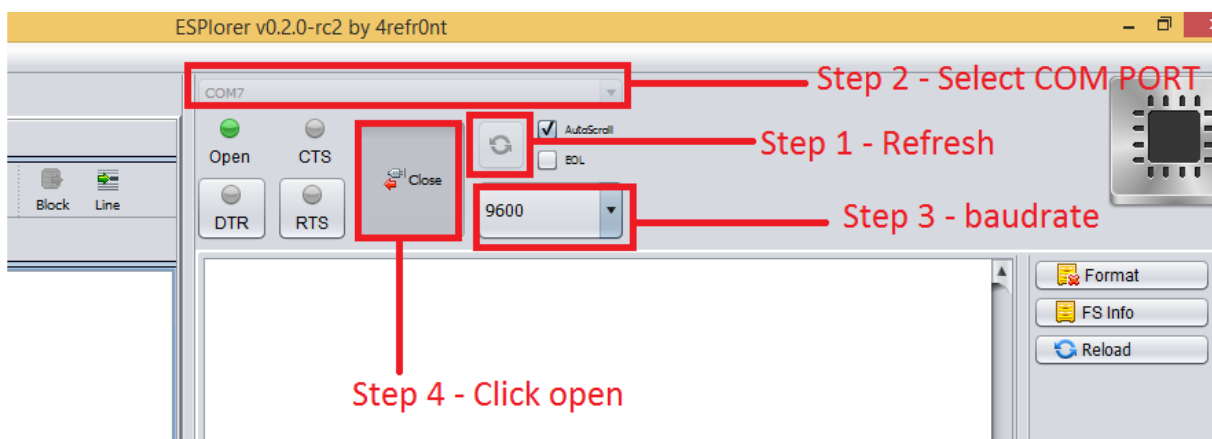
Έχοντας το ESP8266 και τον FTDI προγραμματιστή συνδεδεμένο στον υπολογιστή και το ESPlorer IDE σε λειτουργία, ο μικροελεγκτής μπορεί να επικοινωνήσει με τον υπολογιστή.



Σχήμα 6.11 : Αρχική οθόνη που περιέχει κώδικα

Κοιτάζοντας στην πάνω δεξιά γωνία του προγράμματος ο χρήστης πρέπει να ακολουθήσει τα παρακάτω βήματα.

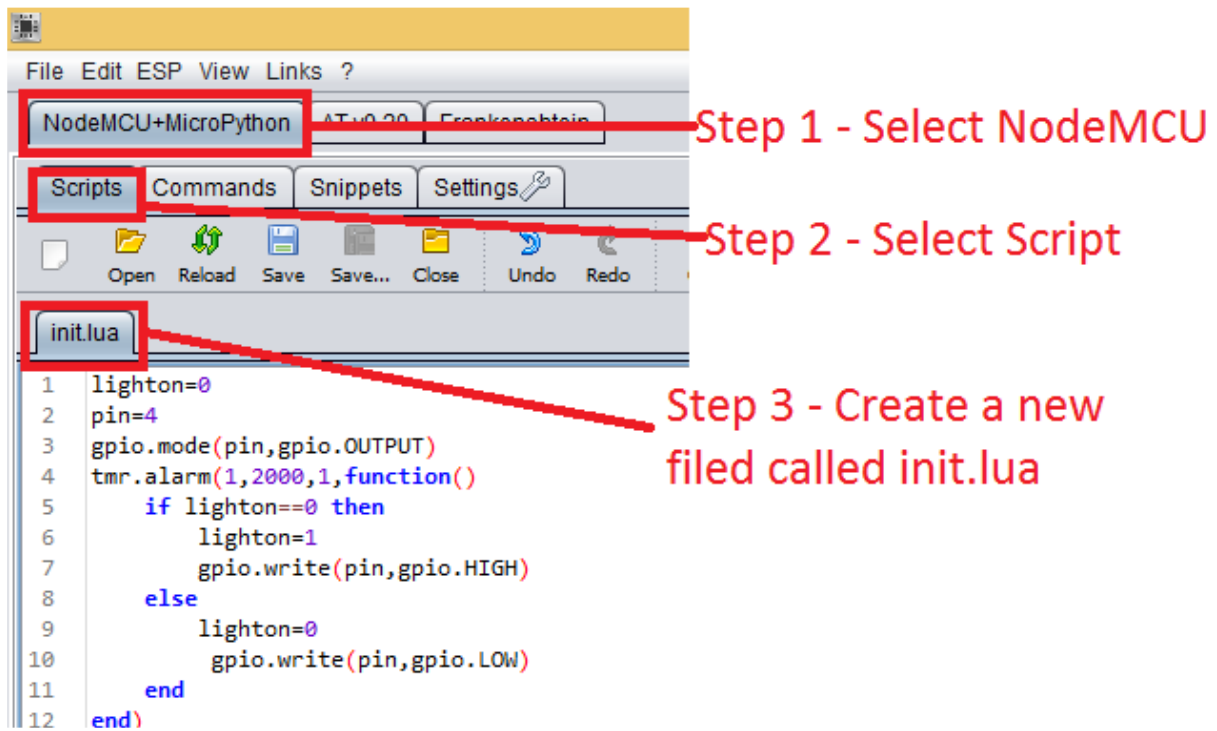
1. Να πατήσει το κουμπί ανανέωση
2. Να επιλέξει την κατάλληλη θύρα COM στην οποία έχει συνδέσει τον FTDI προγραμματιστή
3. Να επιλέξει το κατάλληλο baudrate, ανάλογα με την μονάδα ESP που εργάζεται
4. Να πατήσει το κουμπί Open, για να αρχίσει η επικοινωνία



Σχήμα 6.12 : Επικοινωνία με το ESP8266

Στη συνέχεια κοιτάζοντας την πάνω αριστερή γωνία του προγράμματος ο χρήστης πρέπει να ακολουθήσει τα εξής βήματα

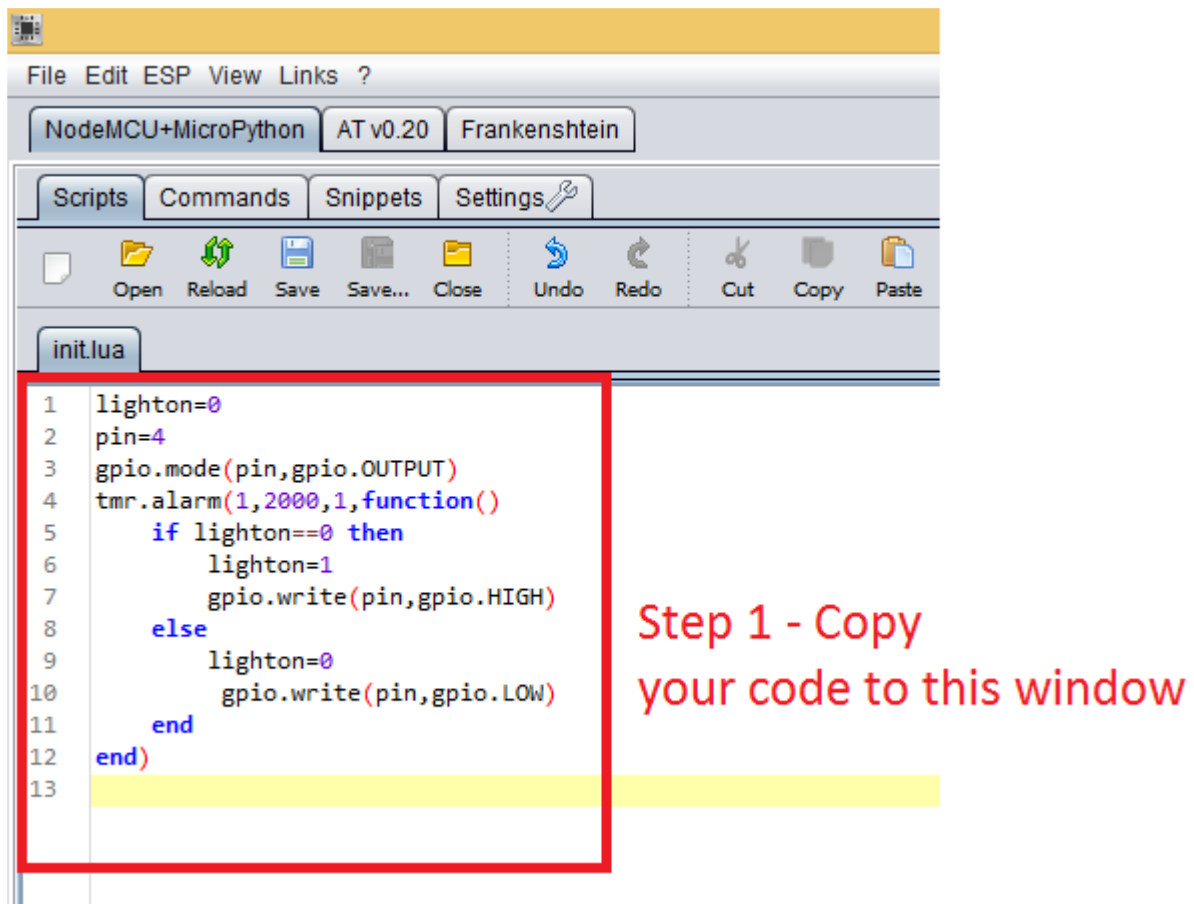
1. Να επιλέξει την καρτέλα NodeMCU
2. Να επιλέξει την καρτέλα scripts
3. Να δημιουργήσει ένα νέο αρχείο και να το ονομάσει «init.lua»



Σχήμα 6.13 : Εισαγωγή αρχείου στο ESPlorer

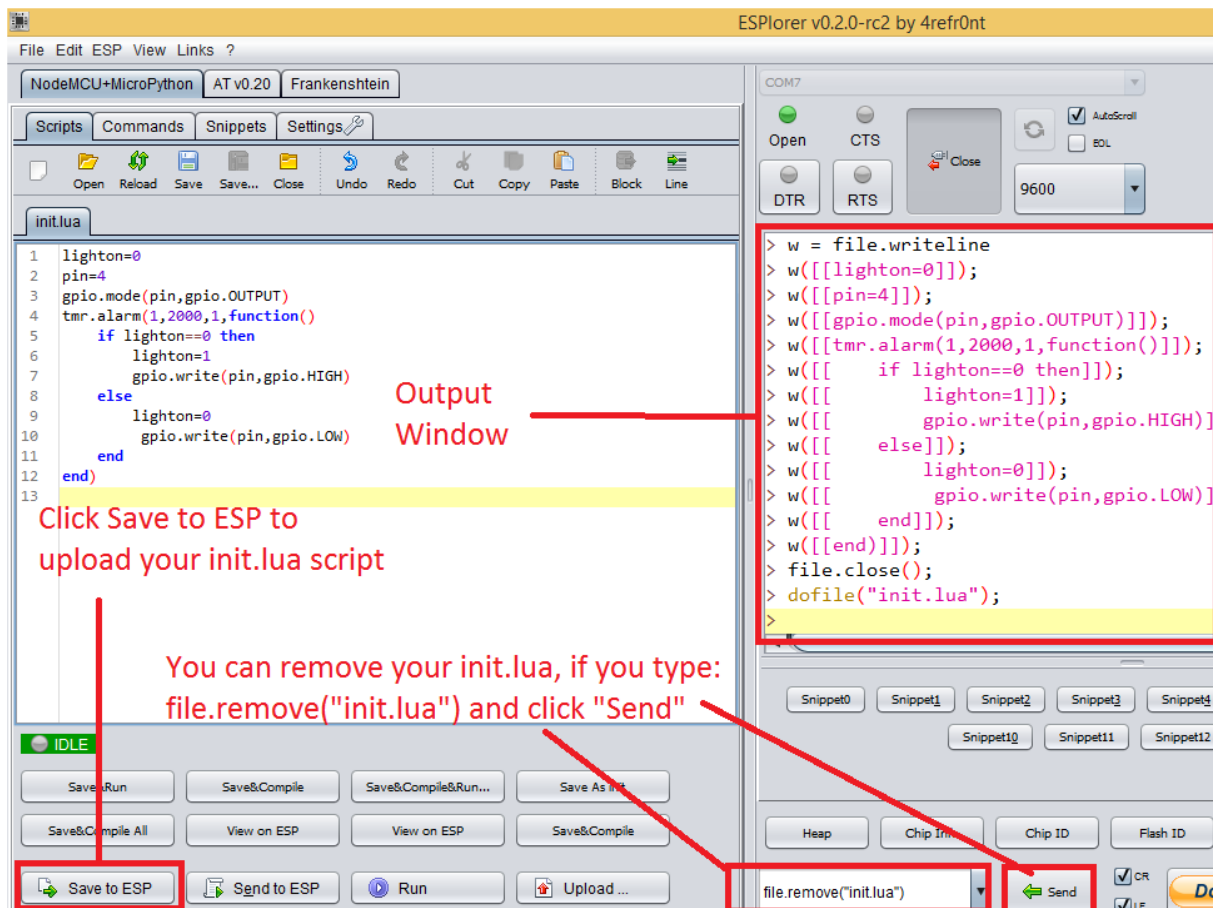
6.4 Εισαγωγή και αποθήκευση κώδικα στη μνήμη του ESP8266

Για να γράψει ο προγραμματιστής κώδικα θα πρέπει να γράψει εντολές μέσα στο νέο αρχείο ή να εισάγει ένα έτοιμο LUA script όπως αυτό που αναφέρθηκε στην προηγούμενη ενότητα. Αντιγράφοντας το προηγούμενο παράδειγμα ο προγραμματιστής βρίσκεται στο σημείο που φαίνεται στην παρακάτω εικόνα.



Σχήμα 6.14 : Αντιγραφή κώδικα μέσα στο αρχείο

Το επόμενο βήμα είναι να αποθηκευτεί ο κώδικας στο ESP8266. Αυτό γίνεται εύκολα πατώντας το κουμπί «Save to ESP» που βρίσκεται στο κάτω αριστερά μέρος του προγράμματος. Τότε στο παράθυρο εξόδου θα πρέπει να εμφανίζονται ακριβώς ποιες εντολές αποστέλλονται στην μνήμη του ESP8266, καθώς και τυχόν συντακτικά λάθη που πρέπει να διορθωθούν. Η οθόνη εξόδου θα μοιάζει όπως την εικόνα που φαίνεται παρακάτω.



Σχήμα 6.15 : Αποθήκευση κώδικα και εμφάνιση αποτελεσμάτων στην οθόνη εξόδου

Σημείωση : Εάν ο χρήστης επιθυμεί να διαγράψει το αρχείο init.lua μπορεί να το κάνει πολύ εύκολα. Απλά πληκτρολογεί την εντολή «file.remove("init.lua")» στο πλαίσιο που φαίνεται στην παραπάνω εικόνα και πατά το κουμπί «Send». Διαφορετικά μπορεί να πληκτρολογήσει την εντολή «file.format()» και να καταργήσει όλα τα αρχεία που βρίσκονται αποθηκευμένα στο ESP8266.

Επίσης μπορεί να πληκτρολογήσει οποιαδήποτε εντολή και να την στείλει μέσα από αυτό το παράθυρο στο ESP8266.

ΚΕΦΑΛΑΙΟ 7

Εξοικονόμηση ενέργειας με τη χρήση του ESP8266

7.1 Αυτονομία και μέθοδος εξοικονόμησης ενέργειας

Στα περισσότερα project που περιέχουν μικροελεγκτές ESP8266, δεν δίνεται ιδιαίτερη σημασία σχετικά με την κατανάλωση ενέργειας. Αυτό συμβαίνει διότι, αν για παράδειγμα κατασκευαστεί ένα έργο με καταγραφέα δεδομένων σε ένα κτίριο, το ESP είναι συνδεδεμένο με το δίκτυο του κτιρίου, καθώς και οι συσκευές ελέγχου τροφοδοτούνται από εκεί αφού είναι εύκολη η πρόσβαση σε αυτό.

Ωστόσο σε ορισμένες περιπτώσεις δεν υπάρχει η δυνατότητα αδιάλειπτης τροφοδοσίας και χρειάζεται να έχουμε αυτονομία ενέργειας στο σύστημα μας. Αυτό επιτυγχάνεται με την χρήση μπαταριών. Ένα παράδειγμα είναι η εγκατάσταση ενός αισθητήρα κίνησης σε μία απομακρυσμένη τοποθεσία όπως το αγρόκτημα.

Σε αυτή την περίπτωση όμως έχουμε συγκεκριμένο ποσό ενέργειας που μπορούμε να καταναλώσουμε. Για παράδειγμα ένα ESP chip με μια πρότυπη μπαταρία 2500 mAh θα διαρκέσει για περίπου 30 μέρες. Αυτό δεν είναι καλό γιατί ο χρήστης θα πρέπει να αλλάζει συνεχώς μπαταρίες.

Σε αυτό το άρθρο θα παρουσιαστεί ο τρόπος με τον οποίο θα μειωθεί σημαντικά η κατανάλωση ενέργειας χρησιμοποιώντας την λειτουργία deep sleep του chip, με την οποία θα δημιουργήσουμε εφαρμογές που η αυτονομία τους θα διαρκέσει για χρόνια.

Το πρώτο πράγμα που χρειάζεται είναι μία πλακέτα ESP8266, χωρίς πολλά χαρακτηριστικά που θα μειώσουν τη διάρκεια ζωής της μπαταρίας. Επίσης, θα χρειαστεί ένα πολύμετρο που θα φαίνεται η κατανάλωση ενέργειας της πλακέτας και μία μπαταρία τύπου Li Po (Lithium Polymer).



Σχήμα 7.1 : Μπαταρία τύπου Li-Po

Ένα πολύμετρο που θα μπορούσε να χρησιμοποιηθεί για την παρατήρηση της μείωσης κατανάλωσης ενέργειας είναι το USB Detector της Keweisi. Η είσοδος του συνδέεται με μία θύρα USB και στην έξοδο συνδέεται το ESP8266. Το εξάρτημα αυτό, αποτελείται από δύο οθόνες στην όψη του που αναγράφουν την τάση στην έξοδο της θύρας USB και το ρεύμα που καταναλώνει οτιδήποτε είναι συνδεδεμένο στην έξοδο. Με τον τρόπο αυτό παρατηρώντας το ρεύμα που καταναλώνει όταν εργάζεται ή όταν είναι σε λειτουργία sleep mode μπορούμε να βγάλουμε τα απαραίτητα συμπεράσματα.



Σχήμα 7.2 : USB Detector – Power capacity tester

Για να μειωθεί η κατανάλωση ενέργειας, η διαμόρφωση που πρέπει να γίνει στο κύκλωμα είναι πολύ απλή. Αρχικά θα πρέπει να συνδεθεί το pin DTR της πλακέτας με το pin XPD ώστε να είναι βέβαιο ότι το chip θα «ξυπνήσει» από την κατάσταση deep sleep mode. Στο σημείο αυτό παρακολουθώντας το USB detector, ο χρήστης μπορεί να μετρήσει την κατανάλωση ισχύος και το ρεύμα που ρέει μέσα στη μονάδα.

7.2 Λειτουργίες Node Sleep και ανάλυση της κατανάλωσης ενέργειας

Για να μειωθεί η κατανάλωση ενέργειας στο ESP8266 γίνεται χρήση των λειτουργιών deep sleep που θα ενεργοποιούνται όταν δεν απαιτείται η εκτέλεση κάποιας εντολής από την μονάδα.

Ένα απλό παράδειγμα είναι όταν ένα τυποποιημένο μήνυμα συνδεθεί στην πλατφόρμα του Dropbox ή οποία είναι μια υπηρεσία cloud και χρησιμοποιείται για την διασύνδεση δεδομένων στο διαδίκτυο. Με τον παρακάτω κώδικα δημιουργείται ένας καταγραφέας δεδομένων που θα κάνει μετρήσεις κάθε δέκα λεπτά, ενώ τον υπόλοιπο χρόνο θα είναι σε κατάσταση sleep και θα καταναλώνεται ελάχιστη ενέργεια.

```

// Library
#include <ESP8266WiFi.h>
// WiFi settings
const char* ssid = "wifi-name";
const char* password = "wifi-password";
// Time to sleep (in seconds):
const int sleepTimeS = 10;
// Host
const char* host = "Dropbox";
void setup()
{
  // Serial
  Serial.begin(115200);
  Serial.println("ESP8266 in normal mode");
  // Connect to WiFi
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("");
  Serial.println("WiFi connected");
  // Print the IP address
  Serial.println(WiFi.localIP());
  // Logging data to cloud
  Serial.print("Connecting to ");
  Serial.println(host);
  // Use WiFiClient class to create TCP connections
  WiFiClient client;
  const int httpPort = 80;
  if (!client.connect(host, httpPort)) {
    Serial.println("connection failed");
    return;
  }
  // This will send the request to the server
  client.print(String("GET /dweet/for/myesp8266?message=lowpower") + " HTTP/1.1\r\n" +
    "Host: " + host + "\r\n" +
    "Connection: close\r\n\r\n");
  delay(10);
  // Read all the lines of the reply from server and print them to Serial
  while(client.available()){
    String line = client.readStringUntil('\r');
    Serial.print(line);
  }
  Serial.println();
  Serial.println("closing connection");
  // Sleep
  Serial.println("ESP8266 in sleep mode");
  ESP.deepSleep(sleepTimeS * 1000000);
}

```

```
}  
void loop()  
{  
}  
}
```

Ο παραπάνω κώδικας είναι αρκετά μεγάλος γιατί περιέχει όλες τις εντολές για να λειτουργήσει το παραπάνω παράδειγμα.

Για να γίνει ο προγραμματισμός του ESP8266 θα πρέπει να αφαιρεθεί το καλώδιο μεταξύ DTR και XPD. Επίσης θα πρέπει να τροποποιηθούν και οι παράμετροι του Wi-Fi ανάλογα με το εκάστοτε router που συνδέεται. Αφού ανέβει ο κώδικας ξανά συνδέουμε την γέφυρα μεταξύ DTR και XPD.

Παρακολουθώντας την κατανάλωση του ρεύματος φαίνεται το ρεύμα που καταναλώνει η μονάδα όταν φορτώνει δεδομένα στο Dropbox ή αν εκτελούσε κάποια άλλη διαδικασία πάνω στην πλακέτα. Σε αυτή την περίπτωση καταναλώνει περίπου 80 mA και μία μπαταρία 2500 mAh θα διαρκέσει περίπου 28,5 ώρες

Μετά από λίγο αφού εισέλθει σε κατάσταση deep sleep mode παρατηρώντας το ρεύμα, φαίνεται η μείωση αφού η κατανάλωση πέφτει στα 8 mA.

Όπως αποδεικνύεται η κατανάλωση μειώνεται κατά 10 φορές. Με αυτό το ρυθμό, στην περίπτωση του καταγραφέα δεδομένων που μένει τις περισσότερες φορές σε κατάσταση sleep mode η μπαταρία θα διαρκέσει 300 ώρες ή 12,5 ημέρες. Αυτό είναι μία σημαντική βελτίωση.

Στην πραγματικότητα σε πολλές εκδόσεις ESP8266 το μεγαλύτερο μέρος της ενέργειας καταναλώνεται από το Led της ένδειξης ισχύος. Είναι βοηθητικό όταν το σύστημά μας είναι συνδεδεμένο στο δίκτυο, όχι όμως στην περίπτωση που τροφοδοτείται από μπαταρία. Ως εκ τούτου πρέπει να αποσυνδεθεί. Εάν συμβεί αυτό η κατανάλωση μειώνεται δραματικά και το πολύμετρο δείχνει 77μΑ ή 0,077 mA. Με μαθηματική προσέγγιση η διάρκεια της μπαταρίας θα διαρκέσει 3,7 χρόνια. Στην πραγματικότητα αυτό δεν συμβαίνει γιατί δεν έχουν ληφθεί υπόψη τα χαρακτηριστικά της μπαταρίας που έχει διάρκεια ζωής 1-2 χρόνια.

Τελικά με τον τρόπο αυτό η κατανάλωση ενέργειας είναι πολύ χαμηλή σε σχέση με την αρχική και έτσι εκτός από εξοικονόμηση ενέργειας, μπορεί το σύστημα να γίνει αυτόνομο σε συνδυασμό με μία ανανεώσιμη πηγή ενέργειας χωρίς να χρειάζεται η αλλαγή μπαταριών ακόμα και μετά από μεγάλο χρονικό διάστημα.

ΒΙΒΛΙΟΓΡΑΦΙΑ

- <https://openhomeautomation.net/>
- <http://randomnerdtutorials.com/esp8266-web-server/>
- <http://esp8266.ru/esplorer/>
- <http://randomnerdtutorials.com/home-automation-using-esp8266/>
- <https://myesp8266.blogspot.gr/2015/03/other-interesting-board-is-this-bo>
- <http://www.newagepublishers.com/samplechapter/001599.pdf>
- <http://www.engineersgarage.com/tutorials/difference-between-microprocessor-and-microcontroller>
- <http://www.engineersgarage.com/microcontroller>
- <http://www.mikroe.com/chapters/view/65/chapter-2-8051-microcontroller-architecture/>
- <http://www.microplanet.gr/>
- <http://cgi.di.uoa.gr/~std04013/>
- <http://www.areresearch.net/2015/11/using-nodedsleep-in-nodemcu-to-extend.html>
- Βιβλίο : Kolban's book on ESP8266
- Βιβλίο : Home Automation With the ESP8266 - Marco Schwartz
- Βιβλίο : Getting Started with the ESPlorer IDE - Rui Santos
- Βιβλίο : Home Automation Using ESP8266 - Rui Santos

