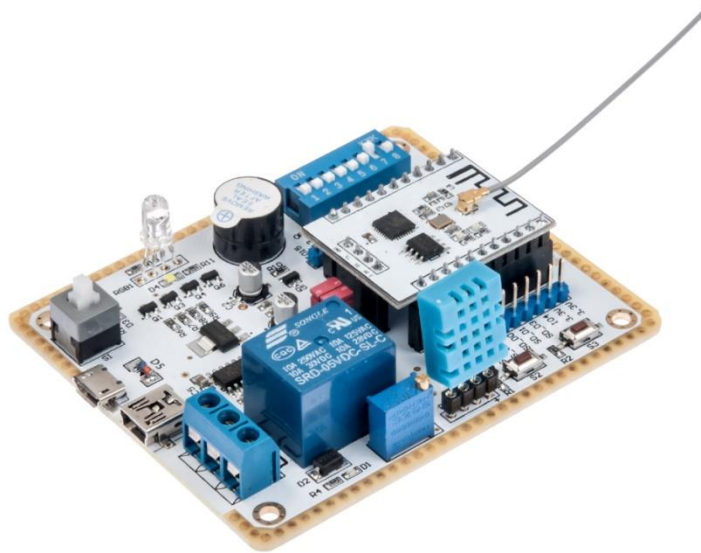


ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ ΔΥΤΙΚΗΣ ΕΛΛΑΔΑΣ
ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ
ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ Τ.Ε.

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ 1637



ΜΕΛΕΤΗ ΤΟΥ ΜΙΚΡΟΕΛΕΓΚΤΗ ESP8266 ΚΑΙ ΑΠΟΜΑΚΡΥΣΜΕΝΟΣ ΕΛΕΓΧΟΣ ΗΛΕΚΤΡΟΝΟΜΟΥ ΜΕ ΤΗ ΧΡΗΣΗ ΤΟΥ ESP8266 WIFI MODULE DEVELOPMENT BOARD

ΣΠΟΥΔΑΣΤΗΣ: Χοϊδάς Διονύσιος (6443)

ΕΠΟΠΤΕΥΩΝ ΚΑΘΗΓΗΤΗΣ: Χριστόπουλος Κωνσταντίνος

ΠΑΤΡΑ 2017

ΠΡΟΛΟΓΟΣ/ ΠΕΡΙΛΗΨΗ

Η παρούσα πτυχιακή εργασία έχει ως σκοπό τη μελέτη και τον προγραμματισμό ενός νέου, χαμηλού κόστους Wi-fi chip με πολλές TCP/IP διευθύνσεις και MCU (Micro Controller Unit), το οποίο δημιουργήθηκε από την κινεζική βιομηχανία με την ονομασία ESP8266.

Θα γίνει προσπάθεια να αναλυθούν οι λειτουργίες του ESP8266 και να εφαρμοστούν σε πραγματικό περιβάλλον, έτσι ώστε να αυτοματοποιηθούν ορισμένες διαδικασίες που μέχρι σήμερα ο έλεγχός τους απαιτούσε χρόνο και χρήμα. Ειδικότερα, θα παρουσιαστεί ο τρόπος με τον οποίο μπορεί να πραγματοποιηθεί έλεγχος ηλεκτρονόμου (ρελέ) με μεταγωγική επαφή από οποιοδήποτε σημείο. Μέσω της απομακρυσμένης οδήγησης του ηλεκτρονόμου -αυτή επιτυγχάνεται με τη βοήθεια οποιασδήποτε συσκευής με πρόσβαση στο Internet- παρέχεται η δυνατότητα ελέγχου του κυκλώματος το οποίο είναι συνδεδεμένο σε αυτόν. Ένα παράδειγμα είναι η εξ αποστάσεως ενεργοποίηση/απενεργοποίηση ηλεκτρικών οικιακών συσκευών.

Συνοψίζοντας, ο μικροελεγκτής θα προγραμματιστεί σε περιβάλλον ESPlorer - Integrated Development Environment (IDE). Η γλώσσα προγραμματισμού ονομάζεται LUA. Θα ακολουθήσουν αναλυτικές εικόνες, καθώς και εμφάνιση των αποτελεσμάτων του προγράμματος.

ΠΕΡΙΕΧΟΜΕΝΑ

| | |
|--|----------------|
| Πρόλογος/ Περίληψη..... | Σελ. iii |
| 1. Εισαγωγή..... | Σελ. 1 |
| 2. Προεπισκόπηση Μικροελεγκτών | |
| 2.1 Γενικές πληροφορίες μικροελεγκτών..... | Σελ. 2 |
| 2.2 Διαφορές από τον μικροεπεξεργαστή..... | Σελ. 2 |
| 2.3 Πλεονεκτήματα μικροελεγκτών..... | Σελ. 3 |
| 2.4 Συνήθη υποσυστήματα..... | Σελ. 4 |
| 2.5 Η λειτουργία του μικροελεγκτή..... | Σελ. 6 |
| 2.6 Πρόσθετες λειτουργίες..... | Σελ. 7 |
| 2.7 Διαδεδομένες κατηγορίες μικροελεγκτών..... | Σελ. 8 |
| 2.8 Γλώσσες προγραμματισμού και εργαλεία ανάπτυξης..... | Σελ. 9 |
| 2.9 Κατασκευαστές μικροελεγκτικών μονάδων..... | Σελ. 9 |
| 3. Μονάδες ESP8266 | |
| 3.1 Οικογένεια μικροελεγκτών ESP..... | Σελ. 10 |
| 3.2 NodeMcu Development kit..... | Σελ. 11 |
| 4. Μικροελεγκτής ESP8266 | |
| 4.1 Θέτοντας σε λειτουργία τον μικροελεγκτή..... | Σελ. 12 |
| 4.2 Συμβατοί με το ESP αισθητήρες..... | Σελ. 15 |
| 4.3 Συμβατοί με το ESP ενεργοποιητές..... | Σελ. 18 |
| 5. ESP8266 Wi-Fi Module Development Board | |
| 5.1 Αναφορά και ανάλυση στοιχείων της πλακέτας..... | Σελ. 20 |
| 6. Βασικές έννοιες | |
| 6.1 Ηλεκτρονόμος..... | Σελ. 24 |
| 6.2 Λειτουργία ηλεκτρονόμου..... | Σελ. 25 |
| 6.3 Πρότυπο 802.11..... | Σελ. 27 |
| 7. Προγραμματισμός μικροελεγκτή | |
| 7.1 Διαδικασία αρχικοποίησης (flashing)..... | Σελ. 28 |
| 7.2 ESPloer IDE..... | Σελ. 29 |
| 7.3 Κατασκευή ιστοσελίδας για το χειρισμό του ESP8266..... | Σελ. 40 |
| 8. Συμπεράσματα..... | Σελ. 44 |
| 9. Βιβλιογραφία..... | Σελ. 45 |

ΚΕΦΑΛΑΙΟ 1

ΕΙΣΑΓΩΓΗ

Η ανάγκη για εξοικονόμηση χρόνου και η επιθυμία του ανθρώπου να διευκολύνει την καθημερινότητά του συμβάλλουν στη συνεχή εξέλιξη της τεχνολογίας. Όσον αφορά τον τομέα της μετακίνησης για παράδειγμα, ο άνθρωπος κατασκεύασε την άμαξα και στη συνέχεια το αυτοκίνητο, με αποτέλεσμα την ταχύτερη και πιο άνετη μεταφορά του. Επίσης, με την έλευση της ηλεκτρικής ενέργειας στις κατοικίες, δημιουργήθηκαν ηλεκτρικές συσκευές, οι οποίες ικανοποιούν αρκετές ανάγκες του ανθρώπου. Επιπλέον, μια σύγχρονη εξέλιξη της τεχνολογίας αποτελούν τα ηλεκτρονικά και μικροϋπολογιστικά συστήματα, τα οποία συνθέτουν τις βάσεις για τη δημιουργία αυτοματισμών και τηλεχειρισμών. Στην πλειοψηφία των ηλεκτρικών/ηλεκτρονικών συστημάτων είναι χρήσιμο να υπάρχει εξ αποστάσεως χειρισμός, ρύθμιση και παρακολούθηση. Ο μικροελεγκτής αποτελεί ένα μικροϋπολογιστικό σύστημα που εξυπηρετεί τη μετάδοση, την επεξεργασία και την αποθήκευση αναλογικών και ψηφιακών σημάτων. 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 που είναι 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.

ΚΕΦΑΛΑΙΟ 2

Προεπισκόπηση μικροελεγκτών

2.1 Γενικές πληροφορίες 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 αυτών, είχε σαν αποτέλεσμα την ανάπτυξη του μικροεπεξεργαστή. Ο μικροελεγκτής είναι ένα προγραμματιζόμενο ολοκληρωμένο κύκλωμα, το οποίο διαθέτει μικροεπεξεργαστή, μνήμη, θύρες εισόδου – εξόδου για την επικοινωνία με εξωτερικές συσκευές, καθώς και διάφορα περιφερειακά κυκλώματα. Χρησιμοποιείται ευρύτατα σε όλα τα ενσωματωμένα συστήματα (embedded systems) ελέγχου χαμηλού και μεσαίου κόστους, όπως αυτά που χρησιμοποιούνται σε αυτοματισμούς, ηλεκτρονικά καταναλωτικά προϊόντα (από ψηφιακές φωτογραφικές μηχανές έως παιχνίδια), ηλεκτρικές συσκευές και κάθε είδους αυτοκινούμενα τροχοφόρα οχήματα. Οι περισσότεροι μικροελεγκτές βασίζονται στην αρχιτεκτονική Van Neumann. Η αρχιτεκτονική αυτή καθόρισε τα τέσσερα βασικά συστατικά που συνθέτουν ένα ψηφιακό σύστημα. Αυτό αποτελείται από έναν επεξεργαστή (CPU), τη μνήμη (RAM), τον χώρο μόνιμης αποθήκευσης (FLASH), καθώς και τις θύρες εισόδου – εξόδου για την επικοινωνία με τις εξωτερικές περιφερειακές συσκευές. Αξίζει να σημειωθεί πως το πρόγραμμα που εκτελεί ο μικροελεγκτής αποθηκεύεται μόνιμα στην μνήμη προγράμματος, και σε περίπτωση που διακοπεί η τροφοδοσία του, δεν σβήνεται. Όταν επανέλθει η τροφοδοσία το πρόγραμμα εκτελείται κανονικά.

2.2 Διαφορές από τον μικροεπεξεργαστή

Στους σύγχρονους μικροεπεξεργαστές για μη ενσωματωμένα συστήματα, όπως για παράδειγμα οι μικροεπεξεργαστές των προσωπικών υπολογιστών, δίνεται έμφαση στην υπολογιστική ισχύ. Η ευελιξία ανάπτυξης διαφορετικών εφαρμογών είναι μεγάλη, καθώς η λειτουργικότητα του τελικού συστήματος καθορίζεται από τα εξωτερικά περιφερειακά, τα οποία διασυνδέονται με την κεντρική μονάδα (μικροεπεξεργαστή), η οποία δεν είναι εξειδικευμένη. Αντίθετα, στους μικροεπεξεργαστές για ενσωματωμένα συστήματα (μικροελεγκτές), οι οποίοι έχουν μικρότερες ή και μηδαμινές δυνατότητες συνεργασίας με εξωτερικά περιφερειακά, αυτού του είδους, η ευελιξία είναι περιορισμένη, καθώς και η υπολογιστική ισχύς. Οι μικροελεγκτές δίνουν έμφαση στο μικρό αριθμό ολοκληρωμένων κυκλωμάτων που απαιτείται για τη λειτουργία μιας συσκευής, το χαμηλό κόστος και την εξειδίκευση.

2.3 Πλεονεκτήματα μικροελεγκτών

- Αυτονομία, μέσω της ενσωμάτωσης σύνθετων περιφερειακών υποσυστημάτων όπως μνήμες και θύρες επικοινωνίας. Με αυτόν τον τρόπο, πολλοί μικροελεγκτές δεν χρειάζονται κανένα άλλο ολοκληρωμένο κύκλωμα για να λειτουργήσουν.
- Η ενσωμάτωση περιφερειακών σημαίνει ευκολότερη υλοποίηση εφαρμογών λόγω των απλούστερων διασυνδέσεων. Επίσης, οδηγεί σε χαμηλότερη κατανάλωση ισχύος, μεγιστοποιώντας τη φορητότητα και ελαχιστοποιεί το κόστος της συσκευής στην οποία ενσωματώνεται ο μικροελεγκτής.
- Χαμηλό κόστος.
- Μεγαλύτερη αξιοπιστία, και πάλι λόγω των λιγότερων διασυνδέσεων.
- 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.
- Περισσότεροι διαθέσιμοι ακροδέκτες για ψηφιακές εισόδους-εξόδους (για δεδομένο μέγεθος ολοκληρωμένου κυκλώματος), λόγω της μη δέσμευσής τους για τη σύνδεση εξωτερικών περιφερειακών.
- Μικρό μέγεθος συνολικού υπολογιστικού συστήματος.
- Η βασική αρχιτεκτονική των μικροελεγκτών δεν διαφέρει από αυτή των κοινών μικροεπεξεργαστών, αν και στους πρώτους απαντάται συχνά η αρχιτεκτονική μνήμης τύπου Harvard, η οποία χρησιμοποιεί διαφορετικές αρτηρίες σύνδεσης της μνήμης προγράμματος και της μνήμης δεδομένων, όπως παραδείγματος χάρη οι σειρές AVR από την Atmel και PIC από την Microchip. Στους κοινούς μικροεπεξεργαστές συνηθίζεται η ενιαία διάταξη μνήμης τύπου Van Neumann.

2.4 Συνήθη υποσυστήματα

Το ολοκληρωμένο κύκλωμα που αποτελεί τον μικροεπεξεργαστή 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, για τη λειτουργία ενός πλήρους ενσωματωμένου υπολογιστικού συστήματος απαιτούνται πολλά εξωτερικά υποσυστήματα και περιφερειακά. Τέτοια είναι τα εξής:

- ◆ Κύκλωμα συνδετικής λογικής (glue logic) για τη σύνδεση των εξωτερικών μνημών και άλλων περιφερειακών παράλληλης σύνδεσης στην αρτηρία δεδομένων (bus) του επεξεργαστή.
- ◆ Μνήμη προγράμματος τύπου ROM, FLASH, EPROM, 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.
- ◆ Μεγάλο μέγεθος μνήμης RAM.
- ◆ Μόνιμη μνήμη αποθήκευσης παραμέτρων λειτουργίας (τύπου EEPROM ή NVRAM). 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 για αυτό πρέπει να έχει πολύ χαμηλή κατανάλωση ρεύματος.
- ◆ Σειρά ανεξάρτητων ψηφιακών εισόδων και εξόδων (Parallel Input-Output, PIO).

Γενικά, όλες οι οικογένειες μικροελεγκτών ενσωματώνουν τα περισσότερα από τα παραπάνω περιφερειακά, με διαφοροποιήσεις κυρίως στην ύπαρξη ή μη εσωτερικής μνήμης προγράμματος και στο είδος της. Έτσι, υπάρχουν:

- Μικροελεγκτές χωρίς μνήμη προγράμματος, οι οποίοι χαρακτηρίζονται ως ROM-less. Αυτοί παρέχουν πάντοτε μια παράλληλη αρτηρία (bus) δεδομένων, πάνω στην οποία συνδέονται εξωτερικές μνήμες προγράμματος και RAM. Τέτοιοι τύποι μικροελεγκτών προορίζονται για πιο ισχυρά υπολογιστικά συστήματα ελέγχου, με μεγαλύτερες απαιτήσεις μνήμης.

- Μικροελεγκτές με μνήμη ROM η οποία κατασκευάζεται με το λογισμικό της (Mask ROM) ή γράφεται μόνο μια φορά (One Time Programmable, OTP). Παρέχουν τη δυνατότητα πολύ χαμηλού κόστους, όταν αγοράζονται σε πολύ μεγάλες ποσότητες.

- Μικροελεγκτές με μνήμη FLASH η οποία μπορεί συνήθως να προγραμματιστεί πολλές φορές. Αυτή είναι η πιο διαδεδομένη κατηγορία. Συχνά ο προγραμματισμός της μνήμης μπορεί να γίνει ακόμη και πάνω στο κύκλωμα της ίδιας της ενσωματωμένης (embedded) εφαρμογής (δυνατότητα In Circuit Programming, ISP). Αυτοί οι μικροελεγκτές έχουν ουσιαστικά αντικαταστήσει τους παλαιότερους τύπους EPROM που έσβηναν με υπεριώδη ακτινοβολία.

2.5 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 ο χρήστης μάθει να χειρίζεται έναν από αυτούς, τότε θα μπορεί να χειριστεί κι 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 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 απεριθμητή προγράμματος αυξάνεται κατά 1 και ως εκ τούτου στέλνει τη διεύθυνση της επόμενης εντολής στον αποκωδικοποιητή εντολών και εκτελεί τις επόμενες οδηγίες.

2.6 Πρόσθετες λειτουργίες

Ανάλογα με την εφαρμογή για την οποία προορίζεται ένας μικροελεγκτής, μπορεί να περιέχει και:

- ▶ Μία ή περισσότερες ασύγχρονες σειριακές θύρες επικοινωνίας (Universal Asynchronous Receiver Transmitter, UART).
- ▶ Σύγχρονες σειριακές θύρες επικοινωνίας (για παράδειγμα I2C, SPI, Ethernet).
- ▶ Ολόκληρα υποσυστήματα για την άμεση υποστήριξη από λογισμικό (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).
- ▶ Ελεγκτή οθόνης υγρών κρυστάλλων (Liquid Crystal Display, LCD).
- ▶ Υποσύστημα προγραμματισμού πάνω στο κύκλωμα (τύπου ISP). Χάρη σε αυτό το κύκλωμα, είναι δυνατός ο επαναπρογραμματισμός (αναβάθμιση λογισμικού) της εφαρμογής, συνδέοντας στη συσκευή μία εξωτερική συσκευή προγραμματισμού (συνήθως σε θύρα UART RS-232) ή ακόμη και από το διαδίκτυο. Αυτή η δυνατότητα απαιτεί την προϋπαρξη λογισμικού υποδοχής (bootstrap) μέσα στη μνήμη προγράμματος και επομένως δεν μπορεί να γίνει σε τελείως άδεια μνήμη προγράμματος.
- ▶ Υποσύστημα προγραμματισμού (τύπου ISP) και διάγνωσης (συνήθως είναι το καθιερωμένο πρότυπο JTAG). Χάρη σε αυτό, είναι δυνατός ο προγραμματισμός της μνήμης προγράμματος χωρίς να προ απαιτείται κάποιο πρόγραμμα υποδοχής. Για αυτόν το λόγο, είναι ιδιαίτερα χρήσιμο στον αρχικό προγραμματισμό, παραδείγματος χάρη κατά τη συναρμολόγηση, ή σε περίπτωση σφάλματος (bug) στο λογισμικό υποδοχής, το οποίο να καθιστά αδύνατη την κανονική αναβάθμιση.

2.7 Διαδεδομένες κατηγορίες μικροελεγκτών

Λόγω του ισχυρότατου ανταγωνισμού αλλά και της τάσης ενσωμάτωσης των μικροελεγκτών σε κάθε ηλεκτρική και ηλεκτρονική συσκευή, η βιομηχανία μικροελεγκτών έχει καταλήξει στην παραγωγή ανταγωνιστικών μοντέλων μαζικής παραγωγής καθώς και μικροελεγκτών για πιο εξειδικευμένες εφαρμογές. Έτσι διακρίνονται οι εξής κυρίως 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). Σχεδιάζονται με έμφαση στη χαμηλή κατανάλωση ισχύος και την αυτάρκεια, ώστε να χρειάζονται ελάχιστα ή και καθόλου εξωτερικά εξαρτήματα και να μη μπορεί να αντιγραφεί εύκολα το εσωτερικό λογισμικό τους. 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, συνηθίζεται η ενσωμάτωση ελεγκτών οθόνης υγρών κρυστάλλων και πληκτρολογίου. Μερικές φορές παρέχουν δυνατότητα εξωτερικής επέκτασης της 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 πρωτόκολλο επικοινωνίας που 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 και βελτιωμένες επιδόσεις σε σχέση με το παρελθόν.

2.8 Γλώσσες προγραμματισμού και εργαλεία ανάπτυξης

Η επιτυχία μιας οικογένειας μικροελεγκτών 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 (debuggers). Στους μικροελεγκτές, τα εργαλεία αυτά δεν αποτελούνται μόνον από λογισμικό, καθώς δεν υπάρχει τυποποιημένος τρόπος επικοινωνίας με αυτούς. Έτσι, διατίθενται προγραμματιστές της εσωτερικής μνήμης (συνήθως μέσω θύρας JTAG ή USB) καθώς και έτοιμες πλακέτες (evaluation boards) με ψηφιακές ή και αναλογικές εξόδους. Οι πλακέτες αυτές έχουν ως κεντρική τους μονάδα τον εκάστοτε μικροελεγκτή, δυνατότητα εύκολου προγραμματισμού και συνήθως συνοδεύονται από λογισμικό ανάπτυξης εφαρμογών με έτοιμα παραδείγματα. Επομένως, ο χρήστης μπορεί να δοκιμάσει τις περισσότερες δυνατότητες του μικροελεγκτή προτού καν σχεδιάσει τη δική του πλακέτα. Στον τομέα των εργαλείων ανάπτυξης, δραστηριοποιούνται όχι μόνο οι ίδιοι οι κατασκευαστές μικροελεγκτών αλλά και εξειδικευμένες εταιρείες.

Οι πιο διαδεδομένες γλώσσες προγραμματισμού των μικροελεγκτών είναι η C, η C++ και οι παραλλαγές τους. Σε τμήματα του λογισμικού όπου απαιτείται ταχύτητα ή μικρό μέγεθος χρησιμοποιούμενης μνήμης, μπορεί να χρησιμοποιείται η Assembly. 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 γενικά εκτοπίσει την Assembly από τις περισσότερες εφαρμογές.

2.9 Κατασκευαστές μικροελεγκτικών μονάδων:

Μερικοί από τους γνωστότερους κατασκευαστές μικροελεγκτών είναι οι εξής:

- ARM (δεν κατασκευάζει αλλά παραχωρεί δικαιώματα χρήσης του πυρήνα)
- Atmel
- Epson
- Freescale Semiconductor (πρώην Motorola)
- Hitachi
- Microchip
- Toshiba
- Texas Instruments
- Espressif Systems

ΚΕΦΑΛΑΙΟ 3

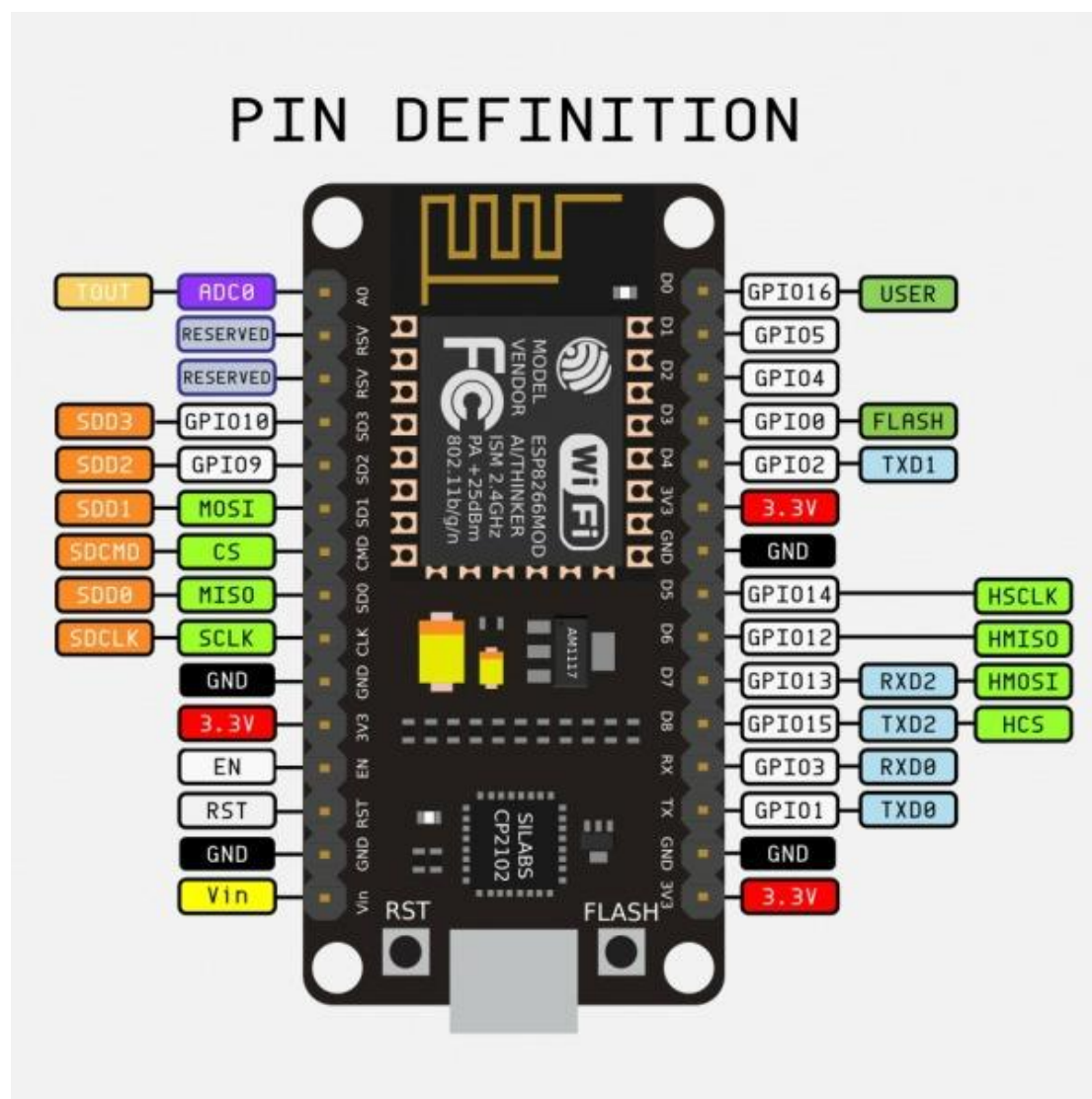
Μονάδες ESP8266

3.1 Οικογένεια μικροελεγκτών ESP

Το ESP αποτελεί έναν καινοτόμο μικροελεγκτή, ο οποίος διαθέτει πληθώρα δυνατοτήτων, που διαρκώς αναπτύσσονται, συνδυάζοντας μικρό κόστος και χαμηλή κατανάλωση. Επίσης, έχει μικρές διαστάσεις και γρήγορες ταχύτητες επεξεργασίας. Πιο συγκεκριμένα, είναι ένα χαμηλού κόστους Wi-Fi chip με ενσωματωμένα πρωτόκολλα TCP/IP, το οποίο μπορεί να δώσει πρόσβαση στο wifi σε οποιοδήποτε μικροελεγκτή. Το ESP μπορεί να τρέξει το ίδιο μία εφαρμογή, ή να φορτώσει μία από άλλη συσκευή. Κατασκευάζεται στη Σανγκάη από την εταιρία “Espressif”. Πρόκειται για ένα αρκετά σύγχρονο chip, το οποίο εμφανίστηκε στη δυτική αγορά τον Αύγουστο του 2014, με το μοντέλο ESP-01 από την AI-Thinker. Είναι η πρώτη οικονομική μονάδα που επιτρέπει τη σύνδεση στο wifi αν και αρχικά δεν υποστήριζε την αγγλική γλώσσα. Η χαμηλή τιμή του και το μικρό μέγεθός του προσέλκυσε το ενδιαφέρον hackers, οι οποίοι τελικά το μετέφρασαν από την κινεζική στην αγγλική γλώσσα. Έκτοτε, ολοένα και περισσότεροι προγραμματιστές το χρησιμοποιούν σε ποικίλες εφαρμογές. Η ικανότητα για γρήγορη επεξεργασία και ο μεγάλος αποθηκευτικός χώρος (1MB), επιτρέπουν τη διασύνδεση με πολλούς αισθητήρες και άλλες συσκευές εφαρμογών (για παράδειγμα arduino), μέσω των GPIOs, με ελάχιστο χρόνο φόρτωσης κατά τη διάρκεια λειτουργίας. Παράλληλα, το κύκλωμά του είναι σχεδιασμένο ώστε να καταλαμβάνει τον ελάχιστο δυνατό χώρο πάνω σε μία πλακέτα PCB. Υπάρχει πληθώρα πληροφοριών τόσο για τις δυνατότητες και τον τρόπο διασύνδεσής του με άλλες συσκευές και αισθητήρες, όσο και για τον προγραμματισμό του, οι οποίες παρέχονται κυρίως από την κοινότητα ανθρώπων που το χρησιμοποιούν. Η σειρά μικροελεγκτών ESP αποτελείται από μονάδες αναφερθείσες ως ESP-xx. Για τη λειτουργία τους απαιτείται μία γέφυρα “USB to UART” καθώς και ένα τροφοδοτικό αποδιδόμενης τάσης 3,3 Volt. Όλα τα παραπάνω συστατικά στοιχεία εμφανίζονται ενσωματωμένα σε ESP8266 Development Boards, όπως για παράδειγμα στο NodeMcu, το οποίο περιλαμβάνει γέφυρα “USB to UART”, βύσμα “micro USB” και ρυθμιστή ισχύος της τάξης των 3,3 Volt. Μονάδες σαν την NodeMcu είναι κατάλληλες για να αποκτήσει κανείς πρώτη επαφή με τον συγκεκριμένο κλάδο τεχνολογίας. Εντούτοις, μία απλή μονάδα ESP8266 αποτελεί οικονομικότερη και ενεργειακά αποδοτικότερη λύση καθώς δεν χρειάζεται τα περιφερειακά στοιχεία για να λειτουργήσει από τη στιγμή ολοκλήρωσης του εκάστοτε έργου.

3.2 NodeMcu Development kit

Τέλη του 2013, η ίδια εταιρία (Espressif) κατασκευάζει ένα software development kit (SDK), το οποίο επιτρέπει τον προγραμματισμό του ESP χωρίς τη χρήση microcontroller. Από τότε, πολλές εταιρίες άρχισαν να διανέμουν τέτοια kit βασισμένα στο ESP. Ένα από τα διασημότερα, οικονομικότερα και αποτελεσματικότερα SDK στην αγορά είναι το NodeMcu development board με firmware NodeMcu. Στην εικόνα 1 παρουσιάζεται το NodeMcu development kit board, βασισμένο στο ESP-12E, διαθέτει θύρα “micro usb” για απευθείας σύνδεση με τον υπολογιστή, χωρίς τη χρήση μικροελεγκτή. Ο προγραμματισμός γίνεται με μία script γλώσσα ανοιχτού κώδικα, την Lua-json.



Εικόνα 1 NodeMcu development kit board

ΚΕΦΑΛΑΙΟ 4

Μικροελεγκτής ESP8266

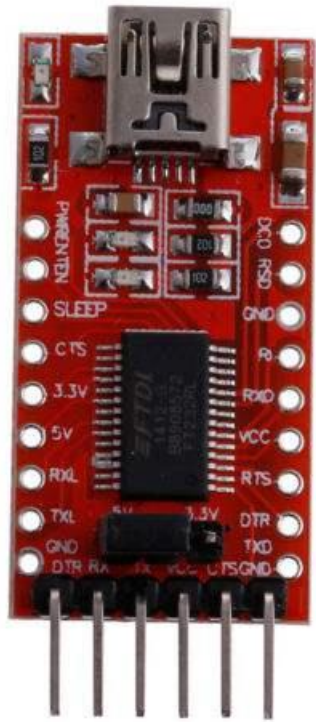
4.1 Θέτοντας σε λειτουργία τον μικροελεγκτή

Ιδιαίτερη προσοχή πρέπει να δοθεί κατά την έναρξη της λειτουργίας του μικροελεγκτή. Είναι πολύ εύκολο, από έναν λανθασμένο ή βιαστικό χειρισμό, να εμφανιστεί βλάβη (για παράδειγμα ένα βραχυκύκλωμα) σε κάποιο επιμέρους ηλεκτρονικό στοιχείο, με αποτέλεσμα τη δυσλειτουργία του μικροελεγκτή ή ακόμη και την ολική καταστροφή του. Βασική προϋπόθεση της λειτουργίας του ESP8266 είναι η ύπαρξη κατάλληλης τροφοδοσίας. Η τάση λειτουργίας των μονάδων ESP8266 είναι 3,3 Volt. Για να εξυπηρετήσει κάποιο σκοπό ή έργο, ο μικροελεγκτής πρέπει να συνδεθεί πρώτα με έναν ηλεκτρονικό υπολογιστή, ώστε να προγραμματιστεί με τον απαραίτητο κώδικα μέσω κατάλληλου λογισμικού (ο κώδικας αποθηκεύεται στη μνήμη του μικροελεγκτή). Εκτός από τον προγραμματισμό, η επικοινωνία της πλακέτας με ηλεκτρονικό υπολογιστή μπορεί να φανεί χρήσιμη για την παρακολούθηση αποτελεσμάτων του εκάστοτε έργου. Για να είναι εφικτή η σύνδεση του μικροελεγκτή με την θύρα USB του ηλεκτρονικού υπολογιστή, παρεμβάλλεται ένας ρυθμιστής τάσης με είσοδο 5 Volt (τάση εξόδου θύρας USB) και έξοδο 3,3 Volt (τάση λειτουργίας ESP8266 μονάδας).

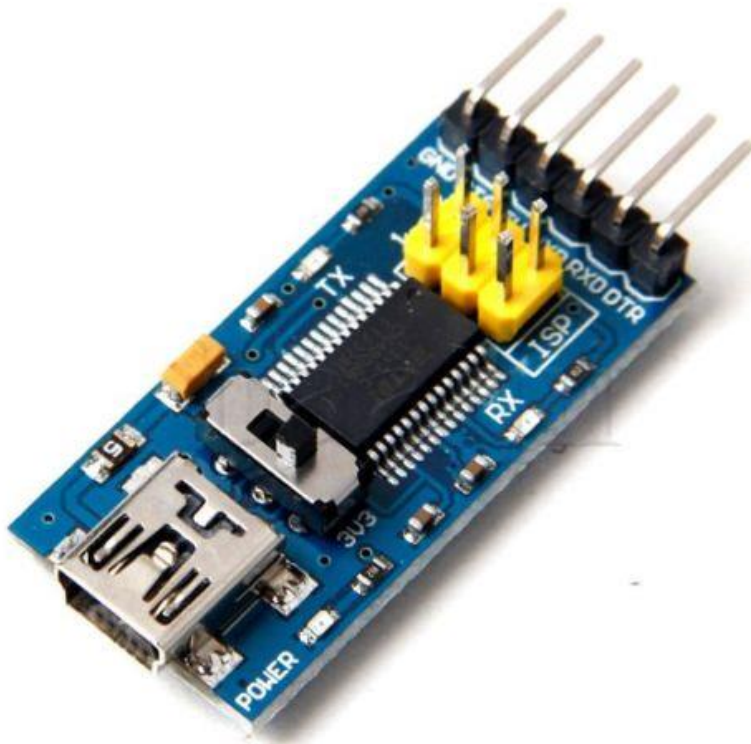
Ρυθμιστές τάσης με είσοδο 5 Volt και έξοδο 3.3 Volt παρουσιάζονται στις εικόνες 2,3 και 4.



Εικόνα 2 CH340

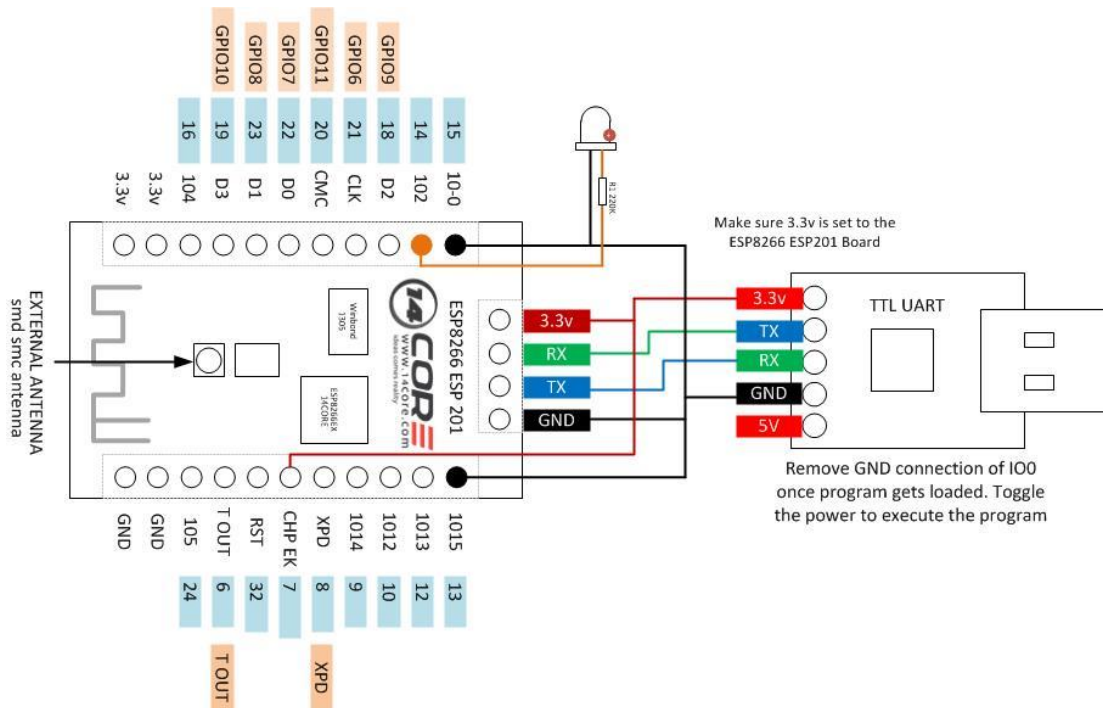


Εικόνα 3 FTDi



Εικόνα 4 FT232RL

Στην εικόνα 5 απεικονίζεται σχηματικά η σύνδεση του μικροελεγκτή ESP8266 με τον μετατροπέα “USB to TTL” (CH340) για μεταφορά δεδομένων:



Εικόνα 5 Σύνδεση μικροελεγκτή ESP8266 με μετατροπέα USB to TTL

Το “UART” είναι η συντομογραφία του universal asynchronous receiver/transmitter, και είναι ένα κύκλωμα των υπολογιστών το οποίο διαμεσολαβεί στη σειριακή επικοινωνία υπολογιστών, ή υπολογιστών με συσκευές, ή και Ενσωματωμένων υπολογιστικών συστημάτων - Embedded Computer Systems, όπως μικροελεγκτές. Η επικοινωνία των UART γίνεται δια μέσου των καθορισμένων θυρών RS-232, RS-422 ή RS-485. Η ταχύτητα επικοινωνίας (ρυθμός μετάδοσης) μπορεί να παραμετροποιηθεί και μετριέται σε baudrate (bits/sec). Σήμερα το κύκλωμα UART είναι ενσωματωμένο ως κύκλωμα τσιπ μέσα στον υπολογιστή ή στην περιφερειακή συσκευή και παρουσιάζεται στην εικόνα 6.



Εικόνα 6 UART

4.2 Συμβατοί με το ESP αισθητήρες



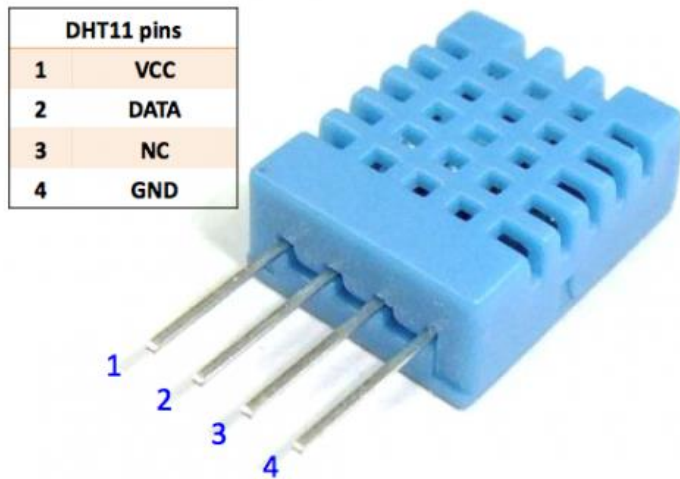
Εικόνα 7 Αισθητήρας έντασης φωτισμού BH1750



Εικόνα 8 Αισθητήρας χρωμάτων TCS34725



Εικόνα 9 Φωτοευαίσθητη αντίσταση



Εικόνα 10 Αισθητήρας θερμοκρασίας/ υγρασίας DHT-11



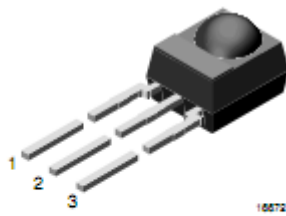
Εικόνα 11 Αισθητήρας wiegand παλμών



Εικόνα 12 Αισθητήρας υπερήχων HC-SR04



Εικόνα 13 Αισθητήρας παθητικών υπεράχων PIR



MECHANICAL DATA

Pinning

1 = OUT, 2 = GND, 3 = V_S

Εικόνα 14 Αισθητήρας κίνησης TSOP4838



Εικόνα 15 Αισθητήρας καπνού GP2Y10



Εικόνα 16 Αισθητήρας κίνησης, γυροσκόπιο και επιταχυνσιόμετρο σε μία πλακέτα MPU 6050

4.3 Συμβατοί με το ESP ενεργοποιητές



Εικόνα 17 Ηλεκτρονόμος (Typical 2-Relay board)



Εικόνα 18 Σερβοκινητήρες



Εικόνα 19 Επιπλέον εισοδοί/έξοδοι με τη χρήση του MCP23017



Εικόνα 20 Επιπλέον εισοδοί/έξοδοι με τη χρήση του PCF8574

ΚΕΦΑΛΑΙΟ 5

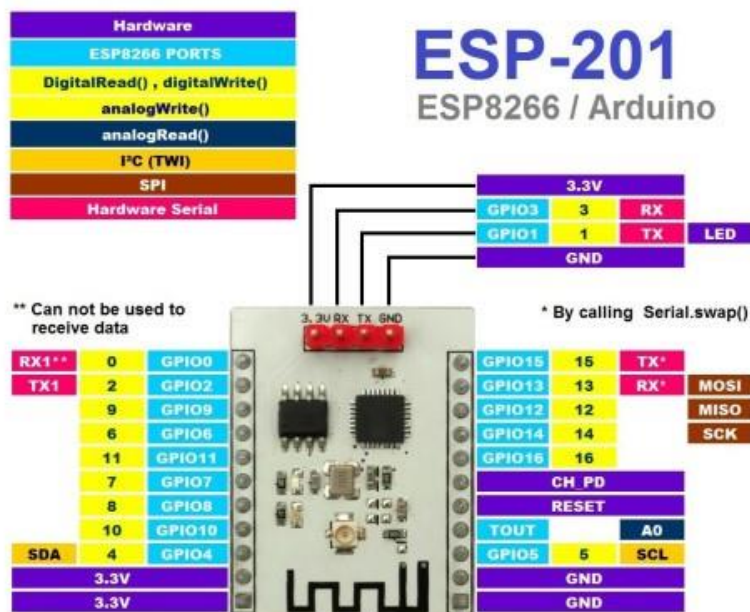
ESP8266 Wi-Fi Module Development Board

Αναφορά και ανάλυση στοιχείων της πλακέτας

Ο μικροελεγκτής που αποτελεί τον πυρήνα της μελέτης της συγκεκριμένης εργασίας παρουσιάζεται στις εικόνες 21 και 22.



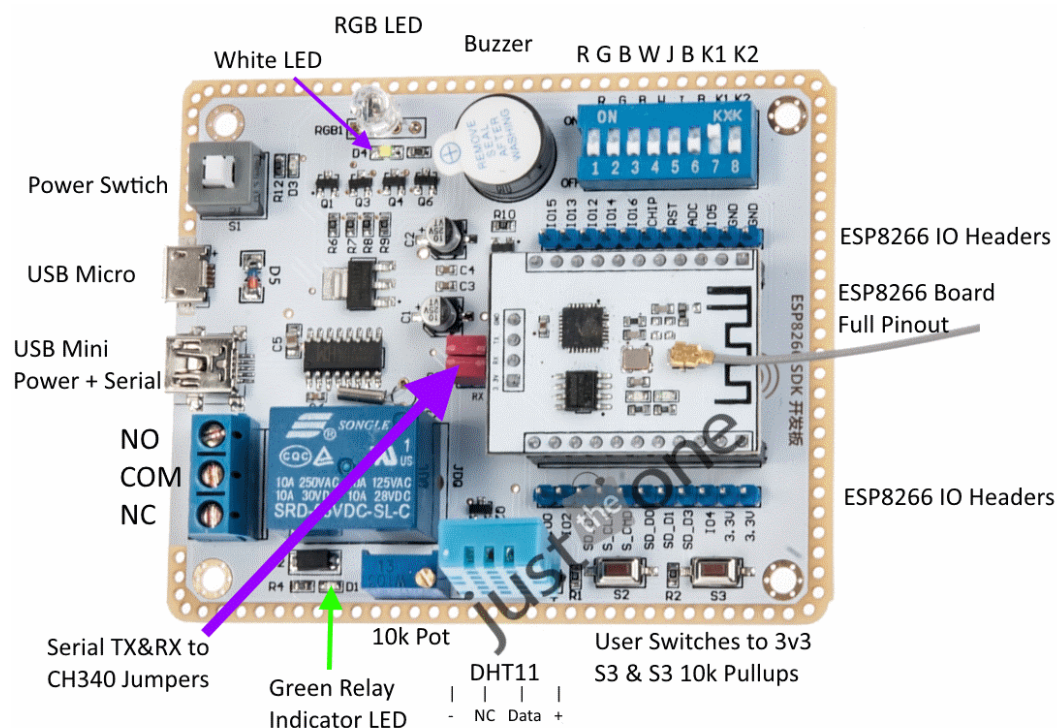
Εικόνα 21 ESP-201



Εικόνα 22 Αναλυτικό σχέδιο περιγραφής εισόδων/εξόδων του ESP-201

Ο συγκεκριμένος μικροελεγκτής αποτελεί μία ασύρματη μονάδα Wi-Fi. Επίσης, διαθέτει 9 ακροδέκτες γενικού σκοπού εισόδου/εξόδου “GPIO” (ακροδέκτες ολοκληρωμένου κυκλώματος που η συμπεριφορά τους ρυθμίζεται μέσω λογισμικού). Ακόμη, έχει ψηφιακοαναλογικό μετατροπέα (A/D), διαμορφωτή εύρους παλμού (PWM), χρονιστή (TIMER), διπλή ασύγχρονη σειριακή θύρα επικοινωνίας (DUART), σύγχρονες σειριακές θύρες επικοινωνίας (I2C & SPI), και κύκλωμα επιτήρησης λειτουργίας, το οποίο αρχικοποιεί το σύστημα αν αυτό εμφανίσει σημάδια δυσλειτουργίας λόγω κολλήματος (WATCHDOG TIMER). Το μέγεθος της μνήμης RAM είναι 35KB, αφού συνδεθεί με το διαδίκτυο, ενώ αντί για μνήμη ROM ο μικροελεγκτής αποθηκεύει τον κώδικα προγραμματισμού σε μνήμη FLASH. Η κεντρική μονάδα επεξεργασίας (32-bit) έχει συχνότητα με προκαθορισμένη τιμή στα 80 MHz αλλά μπορεί να ξεπεράσει τα 160 MHz.

Στην εικόνα 23 παρουσιάζεται η πλακέτα “ESP8266 Wi-Fi Module Development Board”. Στο εμπόριο, μπορεί κανείς να την αποκτήσει, έναντι 12 περίπου ευρώ, ενώ όπως φαίνεται, ο μικροελεγκτής είναι προσαρμοσμένος στους κατάλληλους ακροδέκτες.



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)

Εικόνα 23 ESP8266 Wifi Module Development Board

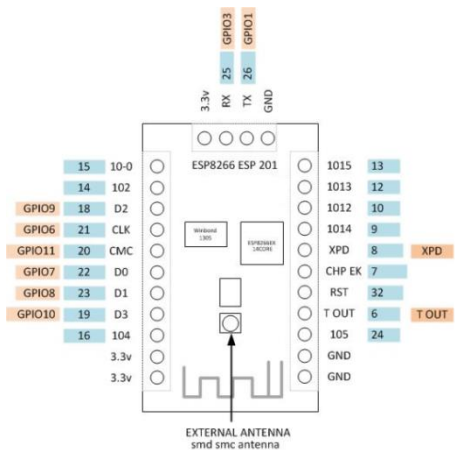
Ο κεντρικός διακόπτης όλου του συστήματος είναι ο διακόπτης S1. Για την τροφοδοσία του συστήματος χρησιμοποιείται η θύρα “USB Micro” ή η θύρα “USB Mini”. Εκτός από τροφοδότηση του συστήματος, μέσω της θύρας “USB Mini”, επιτυγχάνεται και μεταφορά δεδομένων από και προς τον μικροελεγκτή. Μόλις τροφοδοτηθεί το κύκλωμα και πατηθεί ο διακόπτης S1, ώστε να ενεργοποιηθεί το σύστημα, ενεργοποιείται μία δίοδος εκπομπής φωτός (LED). Με αυτό, όπως και με άλλα στοιχεία που είναι ενσωματωμένα στην πλακέτα (λόγου χάρι ένα RGB Led, ένα λευκό Led και ένα βομβητή (buzzer)), ο χειριστής μπορεί πολύ εύκολα να εξοικειωθεί με τον μικροελεγκτή και τον προγραμματισμό του, κάνοντας πειραματικές διαδικασίες (δοκιμάζοντας να αναβοσβήσει ένα Led για παράδειγμα).

Η ύπαρξη μπουτόν (S1, S2, S3) και διακοπών (R, G, B, W, J, B, K1, K2), που είναι συνδεδεμένοι με θύρες του μικροελεγκτή, με GND και Led, διευκολύνουν τους χειριστές στον προγραμματισμό. Μέσω των διακοπών δίδεται η ευκαιρία στον χειριστή να ενεργοποιήσει και να απενεργοποιήσει χειροκίνητα κάποια ηλεκτρονικά στοιχεία της πλακέτας, έτσι ώστε να κατανοήσει βασικές αρχές προγραμματισμού.

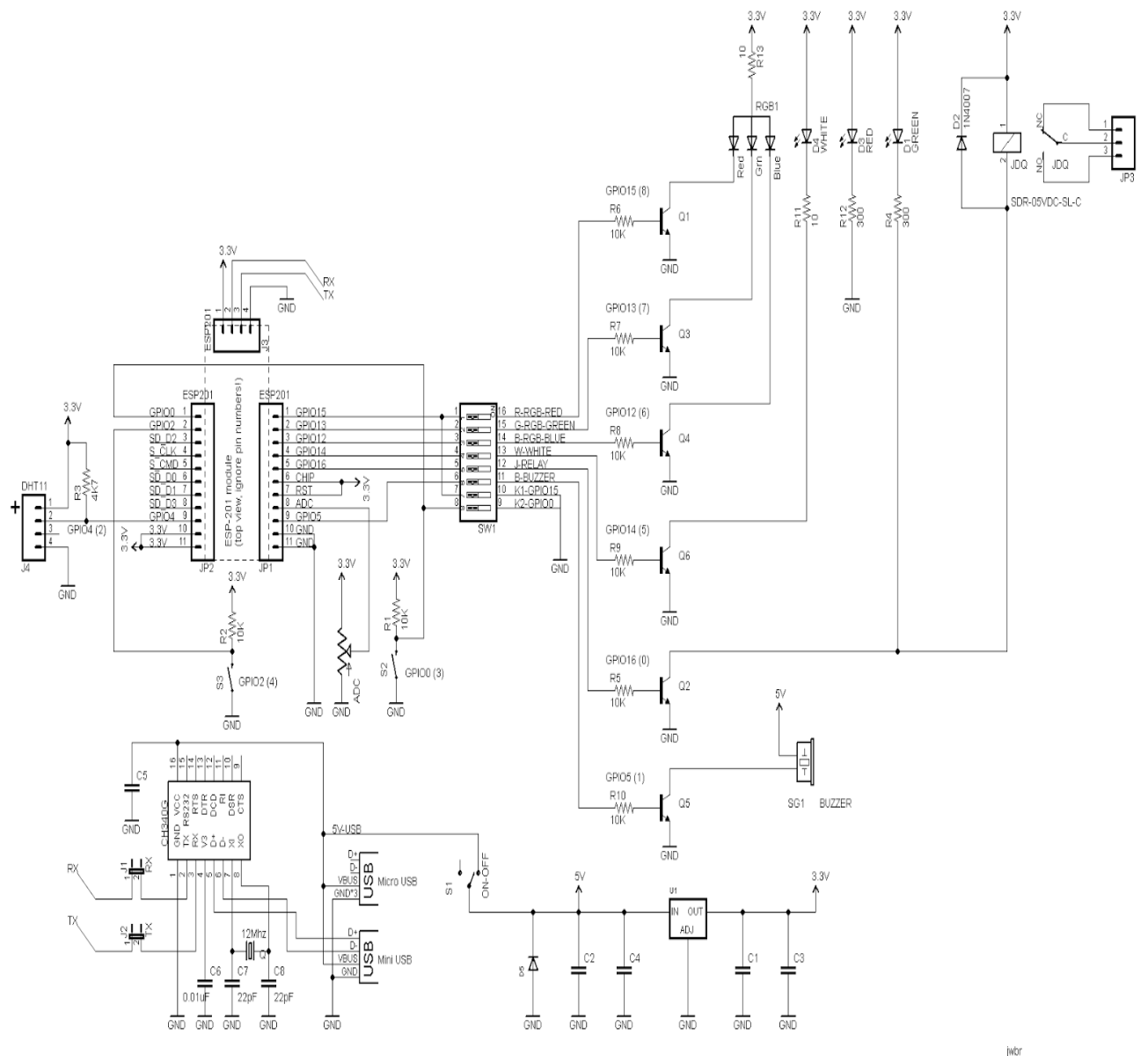
Επιπροσθέτως, ένα χρήσιμο στοιχείο που περιλαμβάνει η πλακέτα είναι ο ηλεκτρονόμος (“Single Relay”), ο οποίος περιλαμβάνει μία μεταγωγική επαφή ή αλλιώς επαφή μορφής “C”, μπορεί δηλαδή να ελέγχει δύο κυκλώματα και ισοδυναμεί με μία επαφή “κανονικά-ανοικτή” και μία επαφή “κανονικά-κλειστή” που διαθέτουν έναν κοινό ακροδέκτη. Το ονομαστικό φορτίο που μπορεί να ελέγξει ο συγκεκριμένος ηλεκτρονόμος είναι: 10A 250VAC/28VDC ή 10A 125VAC/28VDC ή 10A 125VAC/28VDC. Η τροφοδοσία του ηλεκτρονόμου είναι της τάξης των 5V. Εάν συνδεθεί με υπολογιστή για παράδειγμα, τροφοδοτείται από τη θύρα USB (έχει έξοδο 5V) μέσω της πλακέτας, στην οποία και είναι ενσωματωμένος.

Εν συνεχεία, αντιστάσεις, πυκνωτές, ρυθμιστές τάσης και άλλα ηλεκτρονικά στοιχεία αποτελούν μέρος του συστήματος της πλακέτας, όπως επίσης και μία βάση, όπου προσαρμόζεται το ESP-201, στα άκρα της οποίας υπάρχουν μικροί ακροδέκτες για τις εισόδους/εξόδους γενικού σκοπού (GPIO) του ESP8266.

Τέλος, όπως φαίνεται στην εικόνα 24, στο πάνω μέρος του μικροελεγκτή ESP-201 βρίσκονται 4 ακροδέκτες. Οι 2 από αυτούς τους ακροδέκτες χρησιμοποιούνται για να τροφοδοτείται ο μικροελεγκτής, ενώ οι άλλοι 2 είναι υπεύθυνοι για την αποστολή (TX) και την παραλαβή (RX) δεδομένων. Στα δεξιά και αριστερά άκρα του ESP-201, υπάρχουν γενικού σκοπού εισοδοί/εξοδοί, αναλογική είσοδος, reset pin, pin τάσης 3,3V και GND, δίνοντας την ευκαιρία για τροφοδότηση κάποιου στοιχείου. Η κεραία για την ασύρματη επικοινωνία με το router μπορεί να είναι είτε εσωτερική είτε εξωτερική, ανάλογα με την έκδοση του μικροελεγκτή.



Εικόνα 24 Εξωτερική κεραία Wifi και εισόδου/έξοδου του ESP-201



Εικόνα 25 Μονογραμμικό σχέδιο πλακέτας ESP8266 Wifi Module Development Board

ΚΕΦΑΛΑΙΟ 6

Βασικές έννοιες

6.1 Ηλεκτρονόμος

Ο ηλεκτρονόμος ή ρελέ (relay) είναι ένας ηλεκτρικός διακόπτης που ανοίγει και κλείνει ένα ηλεκτρικό κύκλωμα κάτω από τον έλεγχο ενός άλλου ηλεκτρικού κυκλώματος. Στην αρχική μορφή του, ένας ηλεκτρομαγνήτης ενεργοποιούσε το διακόπτη, με το άνοιγμα ή κλείσιμο μιας ή περισσότερων επαφών. Εφευρέθηκε από τον Joseph Henry το 1835. Επειδή ένας ηλεκτρονόμος είναι ικανός να ελέγχει ένα κύκλωμα εξόδου υψηλότερης ισχύος από το κύκλωμα εισόδου, μπορεί να θεωρηθεί, γενικά, μια μορφή ηλεκτρικού ενισχυτή.

Κάθε επαφή ενός ηλεκτρονόμου μπορεί να είναι “Κανονικά-Ανοικτή” (Normally Open, NO), “Κανονικά-Κλειστή” (Normally Closed, NC) ή μεταγωγική (change-over), ανάλογα με τον τύπο της.

Μία επαφή “Κανονικά-Ανοικτή” συνδέει το κύκλωμα όταν ο ηλεκτρονόμος ενεργοποιείται, ενώ το κύκλωμα αποσυνδέεται όταν ο ηλεκτρονόμος είναι ανενεργός. Μία τέτοια επαφή καλείται επίσης Επαφή Μορφής Α ή επαφή "make". Η επαφή μορφής Α είναι ιδανική για εφαρμογές που απαιτούν την ενεργοποίηση μιας πηγής υψηλής τάσης από απόσταση.

Μία επαφή “Κανονικά-Κλειστή” αποσυνδέει το κύκλωμα όταν ο ηλεκτρονόμος ενεργοποιείται, ενώ το κύκλωμα συνδέεται όταν ο ηλεκτρονόμος είναι ανενεργός. Μία τέτοια επαφή καλείται επίσης Επαφή Μορφής Β ή επαφή "break". Η επαφή μορφής Β είναι ιδανική για εφαρμογές που απαιτούν το κύκλωμα να παραμένει κλειστό (ενεργό) μέχρι ο ηλεκτρονόμος να ενεργοποιηθεί.

Μία επαφή “Μεταγωγική” μπορεί να ελέγχει δύο κυκλώματα. Ισοδυναμεί με μία επαφή “Κανονικά-Ανοικτή και μια επαφή “Κανονικά-Κλειστή” που έχουν ένα κοινό ακροδέκτη. Μία τέτοια επαφή καλείται επίσης “Επαφή Μορφής C” (τέτοια διαθέτει ο ηλεκτρονόμος της πλακέτας που χρησιμοποιήθηκε στην συγκεκριμένη εργασία).

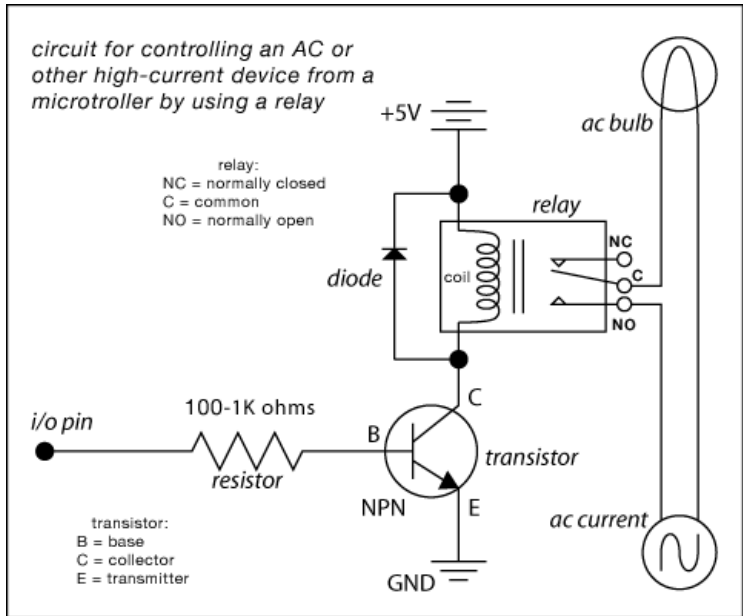
Συνήθως ένας ηλεκτρονόμος αποτελείται από περισσότερες από μία ελεγχόμενες επαφές. Οι επαφές χωρίζονται σε κύριες και βοηθητικές. Οι κύριες διαρρέονται συχνά από ισχυρότερα ρεύματα κι έτσι είναι αυτές που διακόπτουν το κύριο κύκλωμα και συνήθως είναι “Κανονικά-Ανοικτές”. Οι βοηθητικές έχουν όπως υπονοεί και το όνομά τους επικουρικό χαρακτήρα και ο ρόλος τους είναι να βοηθούν στον έλεγχο των αυτοματισμών (που είναι ο κύριος τομέας χρήσης των ηλεκτρονόμων). Για παράδειγμα βοηθούν στην ενεργοποίηση/απενεργοποίηση βοηθητικών κυκλωμάτων, όπως ενδεικτικές λυχνίες (η πλακέτα που χρησιμοποιήθηκε στην συγκεκριμένη εργασία διαθέτει ενδεικτική λυχνία Led, η οποία παραμένει ενεργή όσο ο ηλεκτρονόμος είναι οπλισμένος).

6.2 Λειτουργία ηλεκτρονόμου

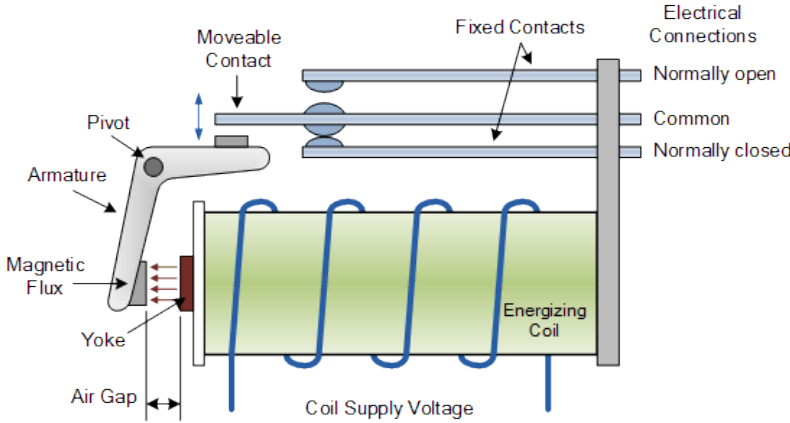
Όταν ηλεκτρικό ρεύμα διαρρέει το πηνίο του ηλεκτρονόμου, το παραγόμενο μαγνητικό πεδίο έλκει έναν οπλισμό που είναι μηχανικά συνδεδεμένος σε μια κινούμενη επαφή. Έτσι, η κινούμενη επαφή είτε συνδέεται με μια σταθερή επαφή είτε αποσυνδέεται από τη σταθερή επαφή. Μόλις το ηλεκτρικό ρεύμα στο πηνίο διακοπεί, ο οπλισμός επιστέφει στη θέση ηρεμίας του εξαιτίας μιας δύναμης επαναφοράς, που είναι ίση με το ήμισυ της μαγνητικής. Η δύναμη επαναφοράς παρέχεται συνήθως από ένα ελατήριο, αλλά και η βαρύτητα χρησιμοποιείται συχνά σε βιομηχανικούς εκκινητές μηχανών. Η μεταβολή της μαγνητικής ροής στο πηνίο γεννά ένα ηλεκτρικό ρεύμα, το λεγόμενο "επαγωγικό", που έχει αντίθετη φορά από εκείνο που παρέχεται στο πηνίο. Για τη λειτουργία του πηνίου και τη μετακίνηση των επαφών απαιτείται σχετικά μεγάλη ένταση ηλεκτρικού ρεύματος, αλλά μόλις ο οπλισμός κλείσει, το ηλεκτρικό ρεύμα που απαιτείται για να κρατήσει τον οπλισμό κλειστό είναι ένα μικρό κλάσμα του αρχικού, τυπικά το 1/10. Οι ηλεκτρονόμοι κατασκευάζονται για να λειτουργούν γρήγορα. Σε μια εφαρμογή χαμηλής τάσης, αυτό γίνεται για τη μείωση του θορύβου. Σε μια εφαρμογή υψηλής τάσης ή υψηλής έντασης ρεύματος, αυτό γίνεται για τη μείωση των σπινθηρισμών (ηλεκτρικών εκφορτίσεων μορφής τόξου).

Εάν το πηνίο διεγείρεται με συνεχές ρεύμα (DC), ανεξάρτητα από το ηλεκτρικό ρεύμα που ρέει διαμέσου των επαφών, μια διάδος μπαίνει συνήθως παράλληλα με το πηνίο. Όταν το πηνίο διεγείρεται, αποκαθίσταται ένα μαγνητικό πεδίο. Όταν το πηνίο αποδιεγείρεται, το καταρρέον μαγνητικό πεδίο δημιουργεί μια αιχμή ηλεκτρικού ρεύματος που θα μπορούσε να βλάψει το υπόλοιπο κύκλωμα. Αν το πηνίο διεγείρεται με εναλλασσόμενο ρεύμα (AC), ένα μικρό χάλκινο δαχτυλίδι πτυχώνεται στο άκρο του σωληνοειδούς πηνίου. Το εναλλασσόμενο ρεύμα μηδενίζεται 100 φορές το δευτερόλεπτο. Σε κάθε χρονική στιγμή μηδενισμού, δεν υπάρχει καμιά μαγνητική δύναμη που να συγκρατεί τις επαφές κλειστές. Το μικρό χάλκινο δαχτυλίδι παρέχει ένα μικρό ρεύμα εκτός φάσεως που καλείται shadow pole (σκιώδης πόλος). Το άθροισμα του εναλλασσόμενου ρεύματος και του shadow pole εξασφαλίζει τη συγκράτηση του οπλισμού στη θέση εμπλοκής σε όλες τις χρονικές στιγμές.

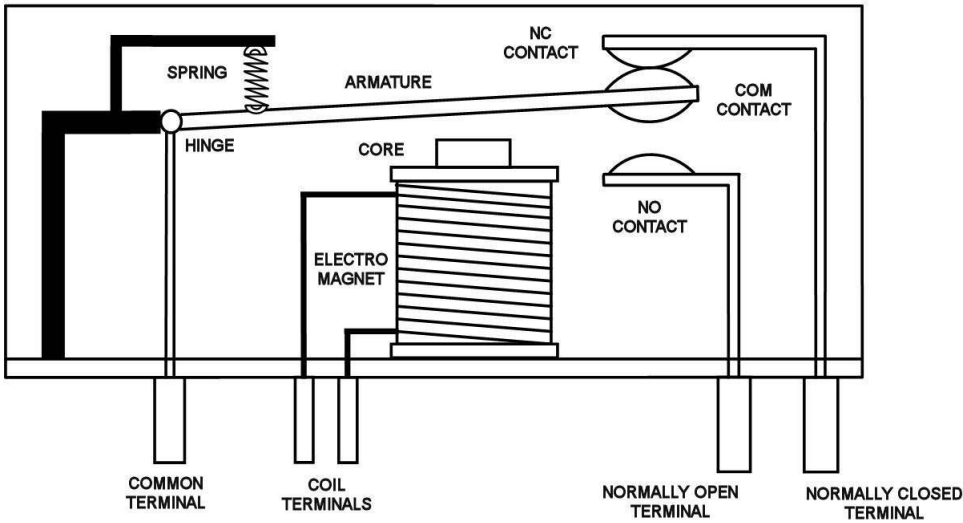
Σε αναλογία με τις λειτουργίες της πρωτότυπης ηλεκτρομαγνητικής συσκευής, ένας ηλεκτρονόμος στερεάς κατάστασης κατασκευάζεται με ένα θυρίστορ ή άλλη συσκευή διακοπής στερεάς κατάστασης. Για να επιτευχθεί ηλεκτρική απομόνωση, μια διάδος φωτοεκπομπής LED χρησιμοποιείται με ένα φωτοτρανζίστορ. Παρακάτω εμφανίζονται οι εικόνες 26,27 και 28, οι οποίες παρουσιάζουν αναλυτικά τον τρόπο λειτουργίας του ηλεκτρονόμου.



Εικόνα 26 Διάταξη ηλεκτρονόμου



Εικόνα 27 Συστατικά στοιχεία ενός ηλεκτρονόμου

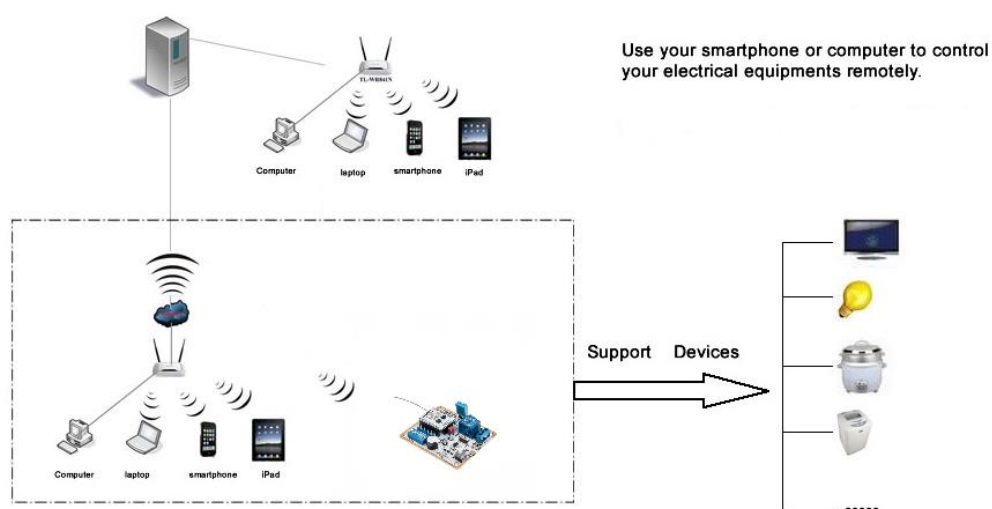


Εικόνα 28 Κάτοψη ηλεκτρονόμου

6.3 Πρότυπο 802.11

Ο μικροελεγκτής της συγκεκριμένης εργασίας χρησιμοποιεί τα πρότυπα 802.11 b/g/n για να επικοινωνεί ασύρματα με το διαδίκτυο (μέσω router). Το IEEE 802.11 είναι μια οικογένεια προτύπων της IEEE για ασύρματα τοπικά δίκτυα (WLAN) που είχαν ως σκοπό να επεκτείνουν το 802.3 (Ethernet, το συνηθέστερο πρωτόκολλο ενσύρματης δικτύωσης υπολογιστών) στην ασύρματη περιοχή. Τα πρότυπα 802.11 είναι ευρύτερα γνωστά ως «WiFi», επειδή η WiFi Alliance, ένας οργανισμός ανεξάρτητος της IEEE, παρέχει την πιστοποίηση για τα προϊόντα που εμπίπτουν στις προδιαγραφές του 802.11. Αυτή η οικογένεια πρωτοκόλλων αποτελεί το καθιερωμένο πρότυπο της βιομηχανίας στο χώρο των ασύρματων τοπικών δικτύων.

Η ονομασία WiFi χρησιμοποιείται για να προσδιορίσει τις συσκευές WLAN που βασίζονται στην προδιαγραφή IEEE 802.11 b/g/n και εκπέμπουν σε συχνότητες 2.4GHz. Ωστόσο το WiFi έχει επικρατήσει και ως όρος αναφερόμενος συνολικά στα ασύρματα τοπικά δίκτυα. Συνήθεις εφαρμογές του είναι η παροχή ασύρματων δυνατοτήτων πρόσβασης στο Internet, τηλεφωνίας μέσω διαδικτύου (VoIP) και διασύνδεσης μεταξύ ηλεκτρονικών συσκευών, όπως τηλεοράσεις, ψηφιακές κάμερες, DVD Player και ηλεκτρονικοί υπολογιστές. Σε φορητές ηλεκτρονικές συσκευές το 802.11 βρίσκει εφαρμογές ασύρματης μετάδοσης, όπως για παράδειγμα στη μεταφορά φωτογραφιών από ψηφιακές κάμερες σε υπολογιστές για περαιτέρω επεξεργασία και εκτύπωση, αν και σε αυτόν τον τομέα έχει υποσκελιστεί από το πρωτόκολλο Bluetooth για τα πολύ μικρότερης εμβέλειας ασύρματα προσωπικά δίκτυα. Στη εικόνα 29 παρουσιάζεται ένα παράδειγμα ολοκληρωμένου συστήματος.



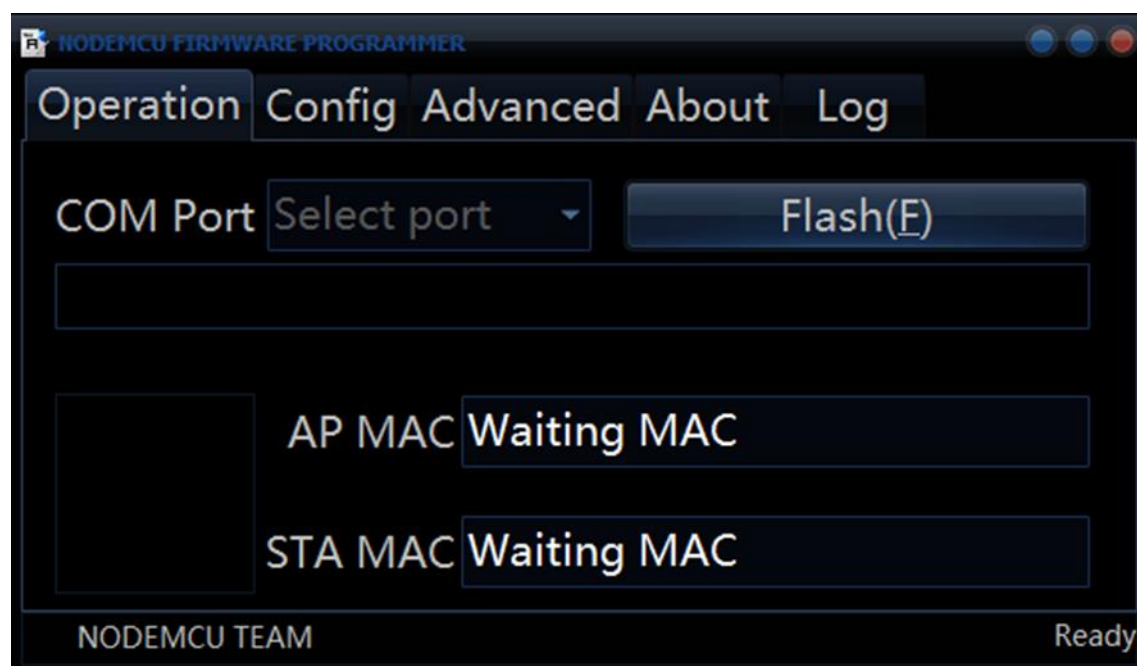
Εικόνα 29 Ολοκληρωμένο σύστημα απομακρυσμένου ελέγχου

ΚΕΦΑΛΑΙΟ 7

Προγραμματισμός Μικροελεγκτή

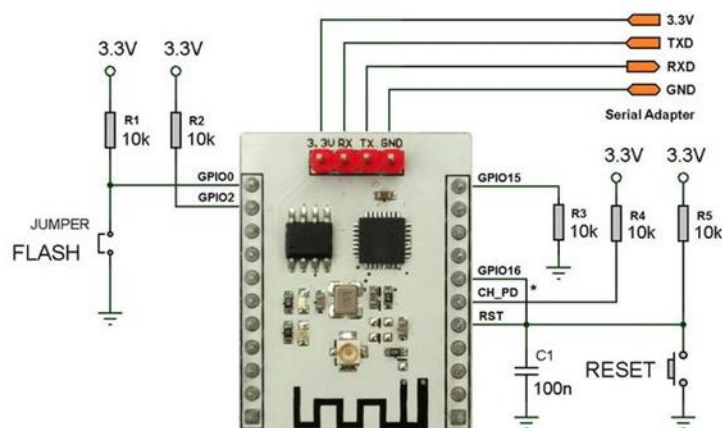
7.1 Διαδικασία αρχικοποίησης (flashing)

Η πλακέτα ESP8266 WiFi Module Development Board έχει προεγκατεστημένη την έκδοση λογισμικού “NodeMCU Lua firmware”. Η συγκεκριμένη έκδοση είναι ανοιχτού κώδικα και προσφέρει μία διαδραστική γλώσσα προγραμματισμού έχοντας τη δυνατότητα δημιουργίας εύκολων εφαρμογών. Το πρώτο βήμα πριν τη συγγραφή του κώδικα προγραμματισμού είναι να αρχικοποιηθεί το λογισμικό. Για να πραγματοποιηθεί αυτή η ενέργεια, απαιτείται το κατέβασμα του “NodeMcu Flasher” από την ηλεκτρονική σελίδα “Github” (<https://github.com/nodemcu/nodemcu-flasher>). Μόλις κατέβει και ανοιχτεί το πρόγραμμα, ο χρήστης του ηλεκτρονικού υπολογιστή κάνει τα εξής βήματα: συνδέει την πλακέτα στον υπολογιστή και στη συνέχεια στην καρτέλα για προχωρημένους (Advanced) επιλέγει στην επιλογή “ρυθμός μετάδοσης” (Baudrate) την τιμή 9600. Στη συνέχεια, ο χρήστης επιλέγει τον αριθμό της θύρας USB στην οποία είναι συνδεδεμένη η πλακέτα στην επιλογή “COM Port”. Στην εικόνα 30 παρουσιάζεται η μορφή του προγράμματος αρχικοποίησης.



Εικόνα 30 Πρόγραμμα αρχικοποίησης NodeMcu Flasher

Είναι σημαντικό να τονιστεί ότι για να γίνει η αρχικοποίηση σωστά, θα πρέπει πρώτα να συνδεθεί ο ακροδέκτης “GND” με τον ακροδέκτη “GPIO0”, ή να κρατηθεί πατημένο το κουμπί “S2” κατά τη διάρκεια της διαδικασίας της αρχικοποίησης (flashing). Η σύνδεση αυτή παρουσιάζεται στην εικόνα 31.



Εικόνα 31 Σύνδεση "GPIO0" με "GND" ως προϋπόθεση για αρχικοποίηση του μικροελεγκτή

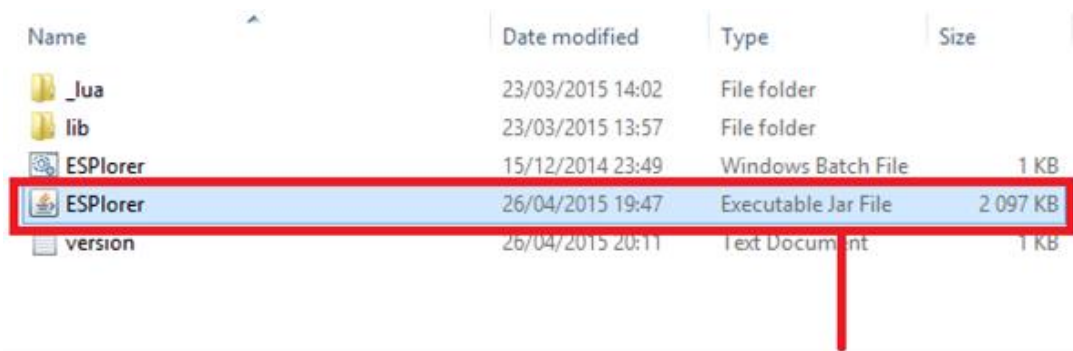
Επόμενο βήμα είναι να πατηθεί η επιλογή “Flash” και ξεκινά η διαδικασία. Μετά την ολοκλήρωση αυτής της διαδικασίας θα πρέπει να εμφανίζεται ένας πράσινος κύκλος στο κάτω αριστερό μέρος του παραθύρου του προαναφερθέντος προγράμματος.

Μόλις ολοκληρωθεί η διαδικασία της αρχικοποίησης, θα πρέπει να διακοπεί η τροφοδοσία της πλακέτας από τον κεντρικό διακόπτη “S1” και στη συνέχεια να αποσυνδεθεί ο ακροδέκτης “GND” από τον ακροδέκτη “GPIO0”.

7.2 ESPlorer IDE

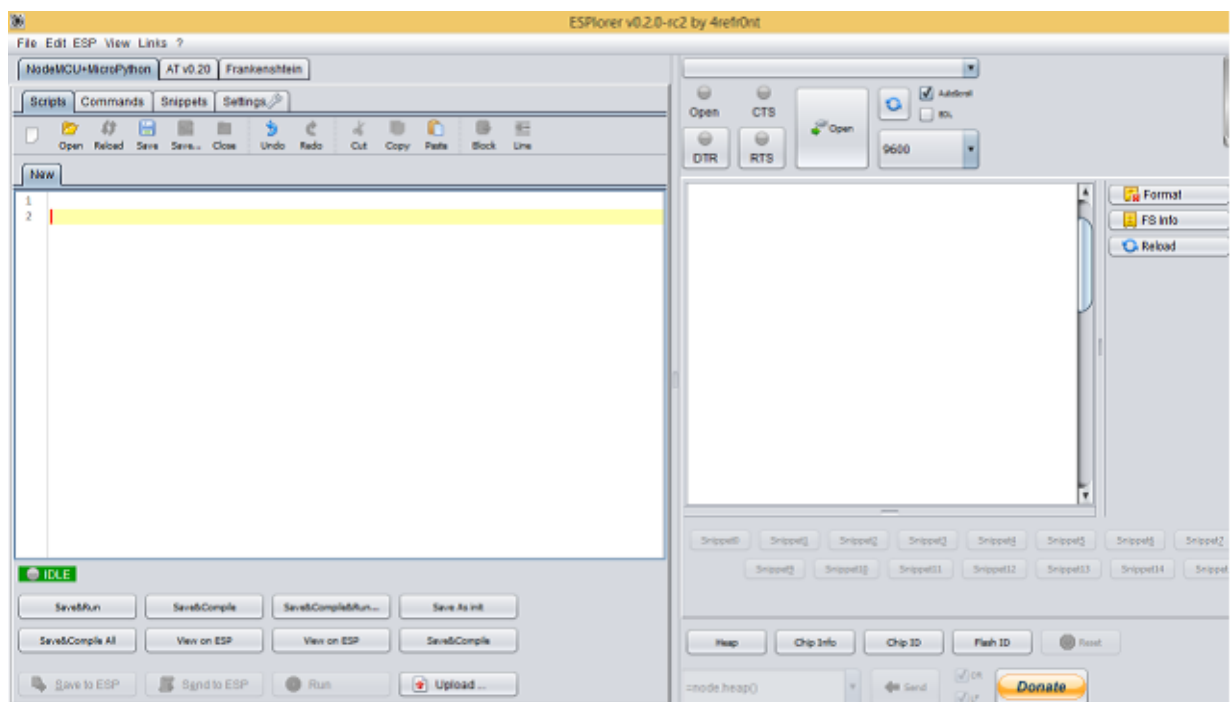
Ένα ολοκληρωμένο περιβάλλον ανάπτυξης (integrated development environment, IDE) είναι μία σουίτα λογισμικού που βοηθάει στην ανάπτυξη προγραμμάτων υπολογιστή. Συνήθως ένα IDE περιλαμβάνει κάποιον επεξεργαστή πηγαίου κώδικα, έναν μεταγλωττιστή, τα εργαλεία αυτόματης παραγωγής κώδικα, τον Debugger, το σύστημα ελέγχου εκδόσεων και εργαλεία κατασκευής γραφικών διασυνδέσεων χρήστη για τις υπό ανάπτυξη εφαρμογές. Για την συγκεκριμένη εφαρμογή θα χρησιμοποιηθεί το “ESPlorer IDE”.

Βασική προϋπόθεση για την εγκατάσταση του προγράμματος “ESPlorer IDE” είναι η εξής: ο ηλεκτρονικός υπολογιστής που θα χρησιμοποιηθεί για τον προγραμματισμό του μικροελεγκτή να διαθέτει “JAVA”. Σε περίπτωση που δε διαθέτει, ο χρήστης μπορεί εύκολα να επισκεφτεί την ηλεκτρονική σελίδα <https://java.com/en/download/> και να κατεβάσει την πιο πρόσφατη έκδοση δωρεάν. Επίσης, ο χρήστης πρέπει να κατεβάσει το πρόγραμμα “ESPlorer IDE” από την ηλεκτρονική σελίδα <https://esp8266.ru/esplorer/#download>. Μόλις ολοκληρωθεί το κατέβασμα, η επόμενη ενέργεια είναι η αποσυμπίεση του φακέλου (“ESPlorer.zip”). Κάθε φορά που ο χρήστης επιθυμεί να ανοίξει το πρόγραμμα “ESPlorer IDE”, το μόνο που πρέπει να κάνει είναι να εκτελεί το πρόγραμμα ”ESPlorer.jar”, το οποίο φαίνεται στην εικόνα 32.



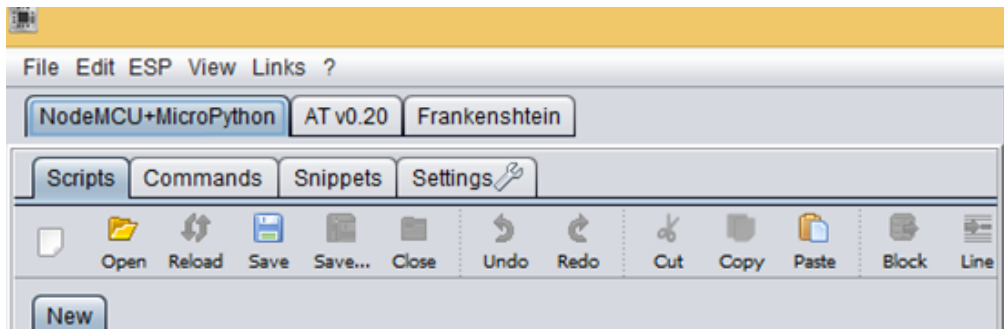
Εικόνα 32 Πρόγραμμα ESPlorer.jar

Εν συνεχεία, μόλις ανοίξει το πρόγραμμα εμφανίζεται το παράθυρο με τη μορφή της εικόνας 33.



Εικόνα 33 Παράθυρο του ESPlorer IDE

Το πρόγραμμα διαθέτει δύο κύρια τμήματα. Στο αριστερό τμήμα υπάρχουν κάποιες επιλογές όπως η δημιουργία νέου αρχείου, η αποθήκευση του αρχείου, και άλλα. Οι επιλογές αυτές παρουσιάζονται στην εικόνα 34.



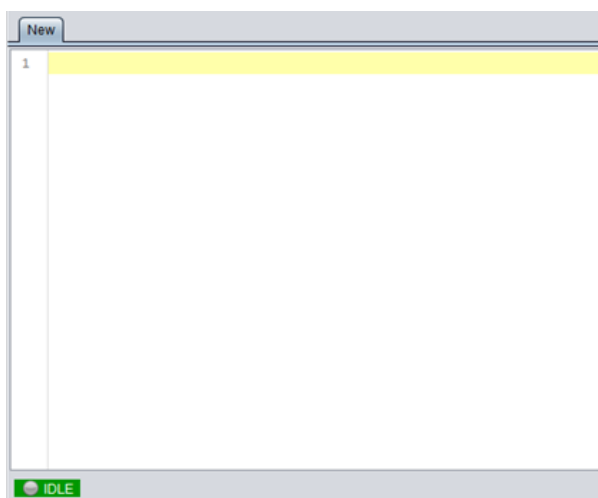
Εικόνα 34 Αριστερό τμήμα παραθύρου ESPlorer IDE

Στο δεξί τμήμα υπάρχουν επιλογές αναφορικά με την σειριακή επικοινωνία της πλακέτας με το προγραμματιστικό περιβάλλον του προγράμματος. Στην εικόνα 35 παρουσιάζονται αυτές οι επιλογές.



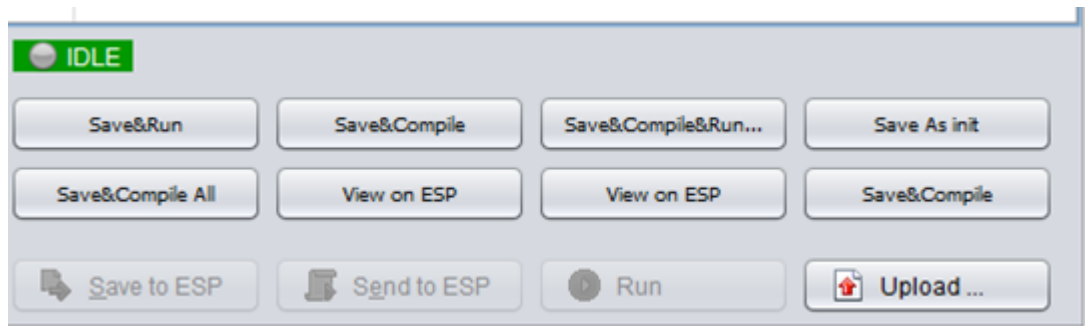
Εικόνα 35 Δεξί τμήμα παραθύρου ESPlorer IDE

Στην εικόνα 36 παρουσιάζεται το σημείο, στο οποίο γράφεται ο κώδικας με τις εντολές (στο αριστερό τμήμα του προγράμματος).



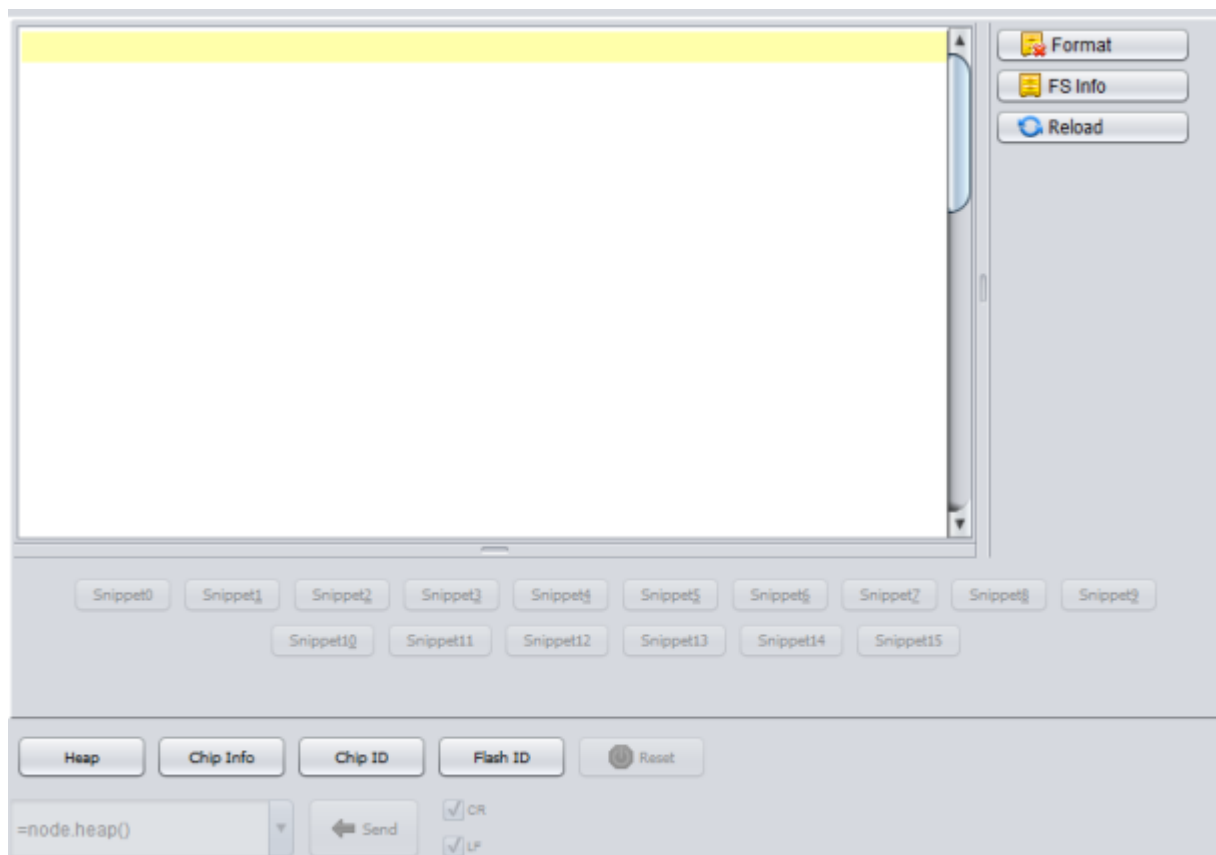
Εικόνα 36 Τμήμα προγράμματος όπου γράφεται ο κώδικας

Ακριβώς κάτω από το σημείο που γράφεται ο κώδικας, όπως φαίνεται στην εικόνα 37, υπάρχουν κάποιες επιλογές, οι οποίες αφορούν τη μεταφορά και αποστολή του κώδικα στον μικροελεγκτή.



Εικόνα 37 Επιλογές προώθησης και αποθήκευσης κώδικα στον μικροελεγκτή

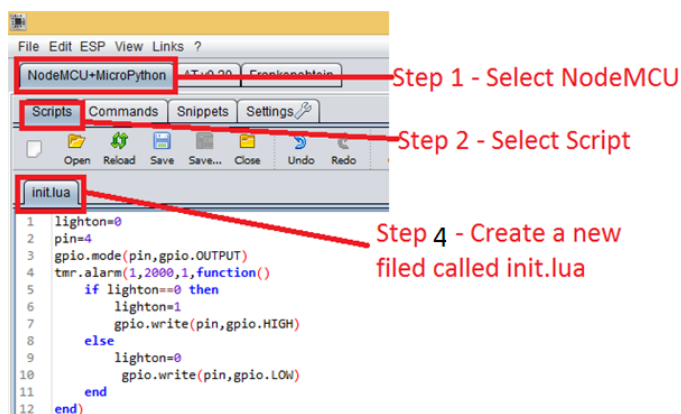
Στην εικόνα 38 παρουσιάζεται το παράθυρο εξόδου (δεξί τμήμα του προγράμματος), όπου εμφανίζονται λάθη που προκύπτουν στον κώδικα, όπως για παράδειγμα συντακτικά λάθη.



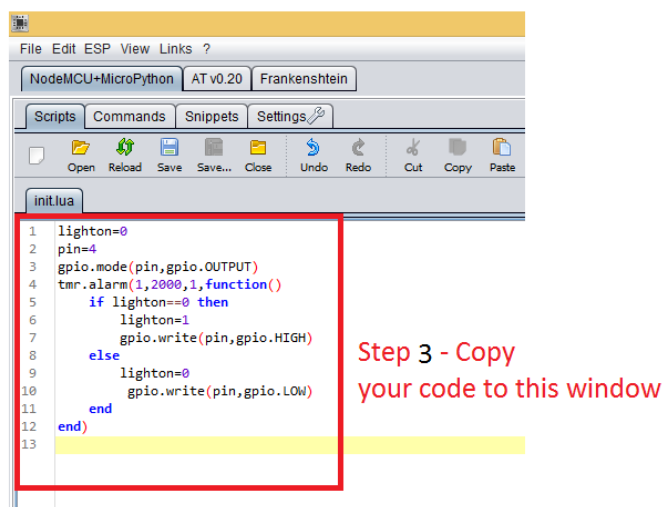
Εικόνα 38 Τμήμα debugger

Για να επικοινωνήσει το “ESP8266” με τον “ESPlorer IDE” πρέπει το πρόγραμμα να είναι σε λειτουργία και η πλακέτα “ESP8266 WiFi Module Development Board” να είναι συνδεδεμένη με το καλώδιο “USB mini”.

Στο αριστερό τμήμα του προγράμματος, ο χρήστης πρέπει: να επιλέξει την καρτέλα “NodeMCU”, να επιλέξει την καρτέλα “Scripts”, να γράψει ή να αντιγράψει κώδικα σε γλώσσα προγραμματισμού LUA. Ακολούθως, ο χρήστης πρέπει να αποθηκεύσει τον κώδικα στο πρόγραμμα με την επιλογή “Save” και την ονομασία “init.lua” αν επιθυμεί αυτός να είναι ο κώδικας που θα εκτελεί ο μικροελεγκτής ακόμη και μετά από “power reset”, αλλιώς το όνομα του αρχείου του κώδικα κατά την αποθήκευση στο πρόγραμμα δεν έχει ιδιαίτερη σημασία. Οι κινήσεις παρουσιάζονται στις εικόνες 39 και 40.

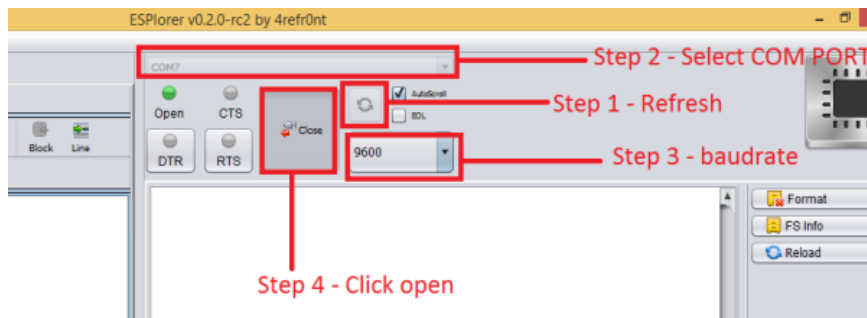


Εικόνα 39 Βήματα για αποθήκευση κώδικα στον μικροελεγκτή



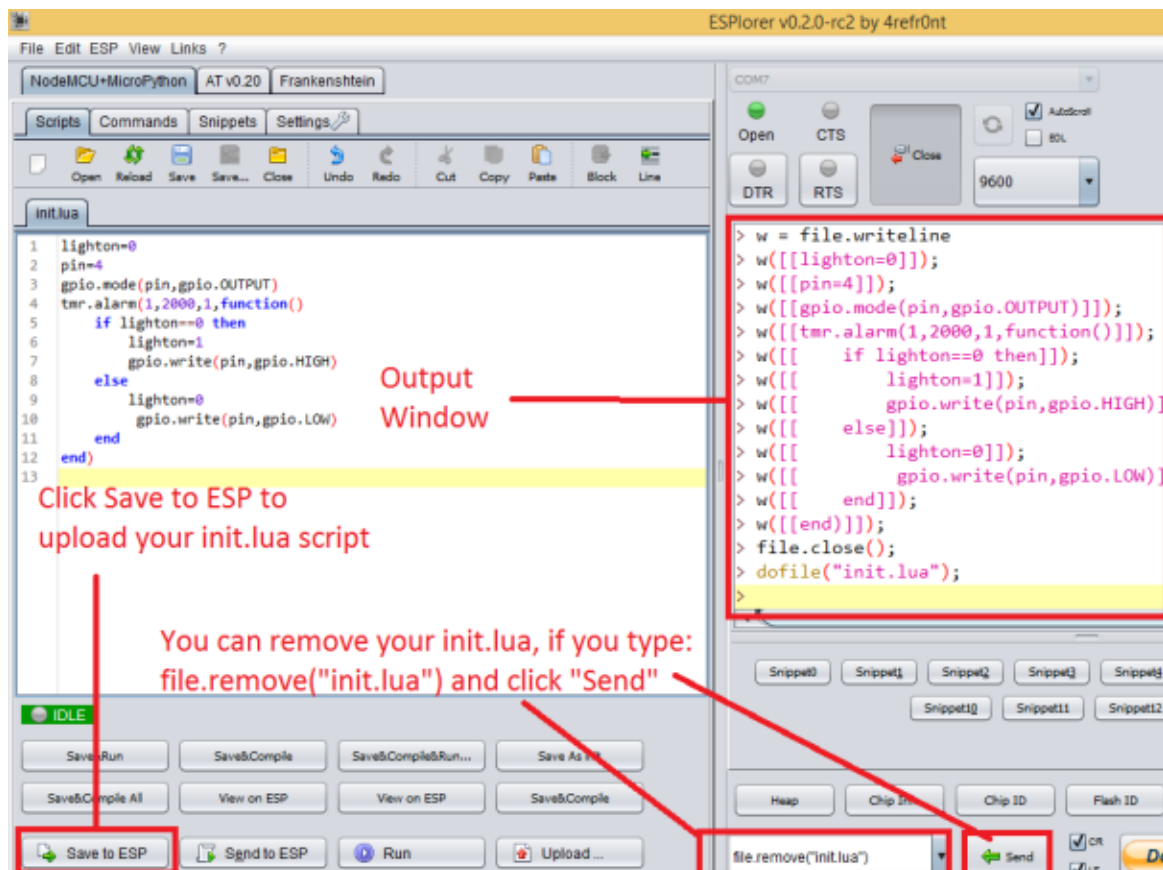
Εικόνα 40 Αντιγραφή κώδικα στο παράθυρο του ESPlorer IDE

Ακόμη, στο δεξί τμήμα του προγράμματος ο χρήστης πρέπει: να πατήσει την επιλογή “ανανέωση” (Refresh), να επιλέξει τη θύρα του ηλεκτρονικού υπολογιστή “USB Port” στην οποία είναι συνδεδεμένη η πλακέτα, να επιλέξει ρυθμό μετάδοσης “Baudrate” στην τιμή 9600, να πατήσει την επιλογή “Open” και να πατήσει τον διακόπτη “S1” της πλακέτας, ώστε να ενεργοποιηθεί το σύστημα. Τα βήματα αυτά παρουσιάζονται στην εικόνα 41.



Εικόνα 41 Επικοινωνία προγράμματος ESPlorer IDE με μικροελεγκτή

Στο αριστερό τμήμα του προγράμματος, ο χρήστης πρέπει να αποθηκεύσει τον κώδικα στο ESP8266. Αυτό γίνεται με την επιλογή “Save to ESP”. Μετά από αυτήν την ενέργεια, στο παράθυρο εξόδου παρουσιάζονται ποιες εντολές μεταφέρονται στη μνήμη του ESP8266 και εμφανίζονται συντακτικά και άλλα λάθη εφόσον υπάρχουν και πρέπει να διορθωθούν. Όλα αυτά παρουσιάζονται αναλυτικά στην εικόνα 42.



Εικόνα 42 Αποστολή, διόρθωση και διαγραφή κώδικα στο πρόγραμμα ESPlorer IDE

Επίσης όπως φαίνεται στην εικόνα 42, δίνεται η ευκαιρία στο χρήστη σε περίπτωση που επιθυμεί να διαγράψει το αρχείο “init.lua” για παράδειγμα, να πληκτρολογήσει την εντολή `file.remove(“init.lua”)` στο πλαίσιο κάτω δεξιά και στη συνέχεια να πατήσει την επιλογή “Send”. Επιπλέον, αν επιθυμεί να καταργήσει όλα τα αρχεία που βρίσκονται αποθηκευμένα στο ESP8266, αρκεί να πληκτρολογήσει την εντολή `file.format()` και την επιλογή “Send”.

Ο κώδικας που θα χρησιμοποιηθεί για τη συγκεκριμένη εφαρμογή είναι ο παρακάτω:

```
wifi.setmode(wifi.STATION)

-- replace with your network details

wifi.sta.config("username","password")

startRead = false

gpio_0=1

gpio_2=0

gpio.mode(gpio_0, gpio.OUTPUT)

gpio.mode(gpio_2, gpio.OUTPUT)

tmr.alarm(3,5000, 1, function()

    conn = nil

    conn=net.createConnection(net.TCP, 0)

    conn:on("receive", function(conn, payload)

        print( "package ticked" )

        -- print( payload )

        startRead = false

        gpioData = ""

        for i = 1, #payload do

            c = payload:sub(i,i)

            if (c=="[") then
```

```

    startRead=true
end
if (startRead) then
    gpioData=gpioData..c
end
if (c=="J") then
    startRead=false
    t=cjson.decode(gpioData)
    for k,v in pairs(t) do
        -- print(k, v)
        if (k==1) then
            if (v) then
                gpio.write(gpio_0, gpio.HIGH)
                -- print ( 'true_' )
            else
                gpio.write(gpio_0, gpio.LOW)
                -- print ( 'false_' )
            end
        elseif (k==2) then
            if (v) then
                gpio.write(gpio_2, gpio.HIGH)
                -- print ( 'true_' )
            else
                gpio.write(gpio_2, gpio.LOW)
                -- print ( 'false_' )
            end
        end
    end
end
end

```

```

        end

    end

end

end)

conn:connect(80,"arduino-wifi.herokuapp.com")

conn:on("connection", function(conn, payload)

    -- print( "after init" )

    conn:send("GET          /api          HTTP/1.1\r\nHost:          arduino-
wifi.herokuapp.com\r\nAccept: application/json\r\n\r\n") end)

    conn:close()

end)

```

Στην τρίτη σειρά του κώδικα, στη θέση του username γράφεται το όνομα του router στο οποίο θα είναι συνδεδεμένο το ESP8266 και στη θέση του password γράφεται ο κωδικός πρόσβασης του router.

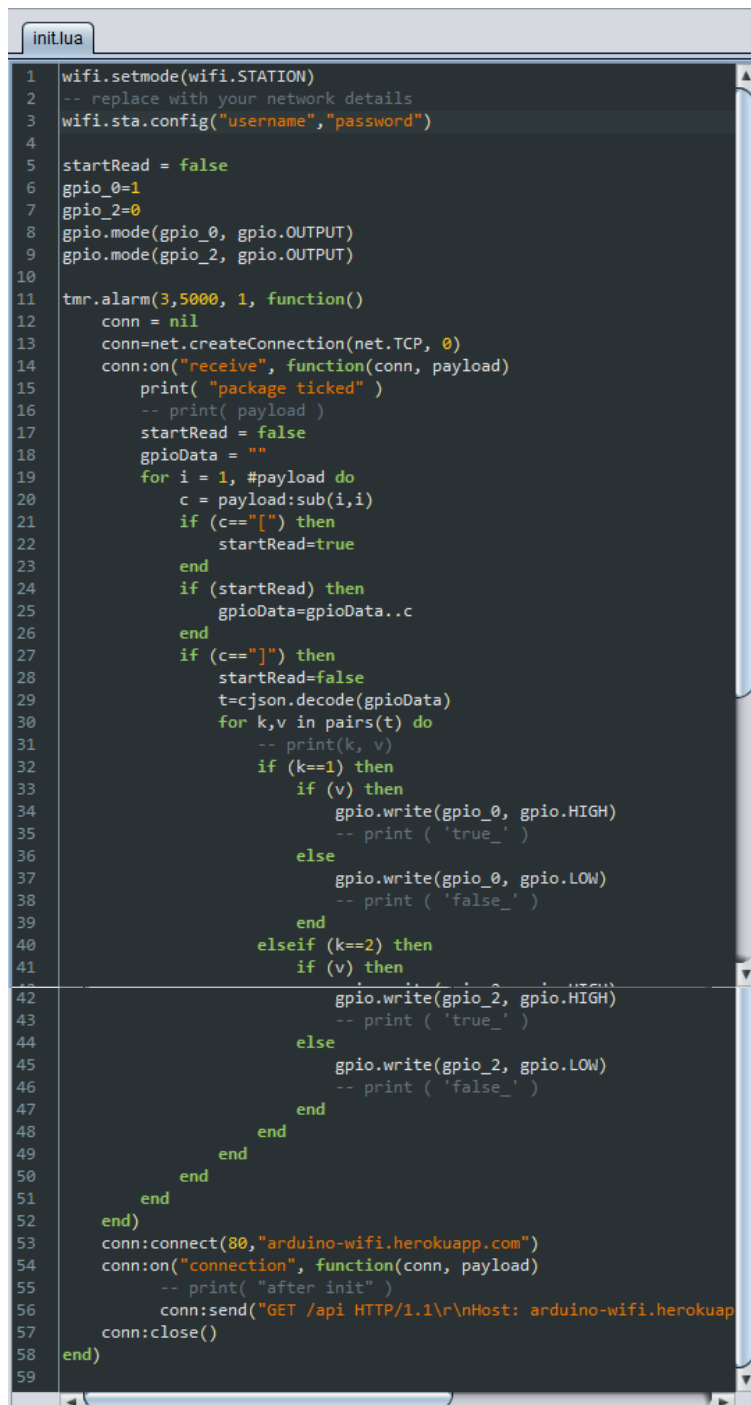
Στην έκτη σειρά του κώδικα, ο χρήστης θέτει στη μεταβλητή “gpio0” τον αριθμό “1” και στην έβδομη σειρά θέτει στη μεταβλητή “gpio2” τον αριθμό “0”. Η επιλογή του αριθμού “1” και του αριθμού “0” εξηγείται στις εικόνες 43 και 44.

| IO index (in code) | ESP8266 GPIO |
|--------------------|---------------|
| 0 [*] | GPIO 16 |
| 1 | GPIO 5 |
| 2 | GPIO 4 |
| 3 | GPIO 0 |
| 4 | GPIO 2 |
| 5 | GPIO 14 |
| 6 | GPIO 12 |
| 7 | GPIO 13 |
| 8 | GPIO 15 |
| 9 | GPIO 3 |
| 10 | GPIO 1 |
| 11 | GPIO 9 |
| 12 | GPIO 10 |

Εικόνα 43 Αντιστοίχιση εισόδων/εξόδων με αριθμούς για χρήση σε κώδικα

εύκολο να κατανοήσει ο χρήστης ότι η εφαρμογή είναι λειτουργική). Αντίστοιχα ο αριθμός “0” χρησιμοποιείται επειδή αναλογεί βάσει του πίνακα της εικόνας 43 στην GPIO 16, η οποία συνδέεται με τον ηλεκτρονόμο της πλακέτας, όπως φαίνεται στο μονογραμμικό σχέδιο της εικόνας 44. Στον ηλεκτρονόμο της πλακέτας μπορεί να συνδεθεί οποιαδήποτε ηλεκτρική/ηλεκτρονική συσκευή είναι συμβατή με τις προδιαγραφές του.

Στο προγραμματιστικό περιβάλλον του “ESPlorer IDE” παρουσιάζεται ο κώδικας όπως στην εικόνα 45.



```
1 wifi.setmode(wifi.STATION)
2 -- replace with your network details
3 wifi.sta.config("username","password")
4
5 startRead = false
6 gpio_0=1
7 gpio_2=0
8 gpio.mode(gpio_0, gpio.OUTPUT)
9 gpio.mode(gpio_2, gpio.OUTPUT)
10
11 tmr.alarm(3,5000, 1, function()
12   conn = nil
13   conn=net.createConnection(net.TCP, 0)
14   conn:on("receive", function(conn, payload)
15     print( "package ticked" )
16     -- print( payload )
17     startRead = false
18     gpioData = ""
19     for i = 1, #payload do
20       c = payload:sub(i,i)
21       if (c=="[") then
22         startRead=true
23       end
24       if (startRead) then
25         gpioData=gpioData..c
26       end
27       if (c=="]") then
28         startRead=false
29         t=cjson.decode(gpioData)
30         for k,v in pairs(t) do
31           -- print(k, v)
32           if (k==1) then
33             if (v) then
34               gpio.write(gpio_0, gpio.HIGH)
35               -- print ( 'true_' )
36             else
37               gpio.write(gpio_0, gpio.LOW)
38               -- print ( 'false_' )
39             end
40             elseif (k==2) then
41               if (v) then
42                 gpio.write(gpio_2, gpio.HIGH)
43                 -- print ( 'true_' )
44             else
45               gpio.write(gpio_2, gpio.LOW)
46               -- print ( 'false_' )
47             end
48           end
49         end
50       end
51     end
52   end)
53   conn:connect(80,"arduino-wifi.herokuapp.com")
54   conn:on("connection", function(conn, payload)
55     -- print("after init" )
56     conn:send("GET /api HTTP/1.1\r\nHost: arduino-wifi.herokuapp.com\r\n\r\n")
57   conn:close()
58 end)
59
```

Εικόνα 45 Κώδικας γραμμένος στο προγραμματιστικό περιβάλλον ESPlorer IDE

7.3 Κατασκευή ιστοσελίδας για τον χειρισμό του ESP8266

Η ιστοσελίδα την οποία επισκέπτεται οποιοσδήποτε επιθυμεί να ελέγξει τις εξόδους του ESP8266 δημιουργήθηκε μέσω της σύγχρονης ηλεκτρονικής πλατφόρμας “heroku” (host). Η συγκεκριμένη πλατφόρμα διαθέτει σαφείς οδηγίες αναφορικά με τη δημιουργία λογαριασμού του χρήστη του μικροελεγκτή. Επόμενο βήμα είναι η χρήση κώδικα σε γλώσσα προγραμματισμού της επιλογής του χρήστη, με σκοπό τη διαμόρφωση της ιστοσελίδας (εισαγωγή διακοπών για παράδειγμα). Τέλος, αρκεί να επικοινωνήσει το ESP8266 με τη σελίδα μέσω αιτημάτων επικοινωνίας (“requests”). Χαρακτηριστικά ιστοσελίδας: “front-end”: -HTML: mdl -JS: firebase και “back-end”: -nodejs: expressjs, firebase. Παρακάτω παρουσιάζεται ο κώδικας σε γλώσσα “HTML”, που χρησιμοποιήθηκε για τη διαμόρφωση της ιστοσελίδας, ο οποίος με δεξί “κλικ” του ποντικιού και την επιλογή “Προβολή κώδικα σελίδας”, είναι εμφανής στους επισκέπτες:

```
<!DOCTYPE html>

<html lang="en">

<head>

  <title></title>

  <meta charset="UTF-8">

  <meta name="viewport" content="width=device-width, initial-scale=1">

  <link rel="stylesheet"
href="https://fonts.googleapis.com/icon?family=Material+Icons">

  <link rel="stylesheet" href="https://code.getmdl.io/1.3.0/material.indigo-
pink.min.css">

  <script defer src="https://code.getmdl.io/1.3.0/material.min.js"></script>

  <style>

    body {

      margin: 2em

    }

  </style>

</head>
```

```

<body>

  <h1>Arduino GPIO</h1>

  <label id="0" style="margin-bottom: 2em" class="mdl-switch mdl-js-switch mdl-
js-ripple-effect" for="gpio-0">

    <input type="checkbox" id="gpio-0" class="mdl-switch__input"
onchange="toggle(this, event)">

    <span class="mdl-switch__label">GPIO 0</span>

  </label>

  <label id="1" style="margin-bottom: 2em" class="mdl-switch mdl-js-switch mdl-
js-ripple-effect" for="gpio-1">

    <input type="checkbox" id="gpio-1" class="mdl-switch__input"
onchange="toggle(this, event)">

    <span class="mdl-switch__label">GPIO 1</span>

  </label>

  <label id="2" style="margin-bottom: 2em" class="mdl-switch mdl-js-switch mdl-
js-ripple-effect" for="gpio-2">

    <input type="checkbox" id="gpio-2" class="mdl-switch__input"
onchange="toggle(this, event)">

    <span class="mdl-switch__label">GPIO 2</span>

  </label>

  <label id="3" style="margin-bottom: 2em" class="mdl-switch mdl-js-switch mdl-
js-ripple-effect" for="gpio-3">

    <input type="checkbox" id="gpio-3" class="mdl-switch__input"
onchange="toggle(this, event)">

    <span class="mdl-switch__label">GPIO 3</span>

  </label>

  <script src="https://www.gstatic.com/firebasejs/3.9.0/firebase.js"></script>

  <script>

    // Initialize Firebase

```

```

const config = {
  apiKey: "AIzaSyCdL5w6NO0rn6ouj6Dpd3FptXE-A0BPsb",
  authDomain: "arduino-wifi-c6fca.firebaseio.com",
  databaseURL: "https://arduino-wifi-c6fca.firebaseio.com",
  projectId: "arduino-wifi-c6fca",
  storageBucket: "arduino-wifi-c6fca.appspot.com",
  messagingSenderId: "41329244619"
};

firebase.initializeApp(config);

// database root reference
const dbRef = firebase.database().ref('switches');

// initialise switches state
dbRef.once('value').then((snapshot) => {
  const arr = snapshot.val();
  for (let i = 0; i < arr.length; ++i) {
    if (arr[i]) {
      document.getElementById(i).MaterialSwitch.on();
    }
  }
});

// toggle event db persistence
const toggle = (el, event) => {
  const index = parseInt(event.srcElement.id.split('-')[1]); // hack DOM tree
  dbRef.once('value').then((snapshot) => {

```



```
        dbRef.child(index).set(!snapshot.val()[index]);
    });
}
</script>
</body>

</html>
```

Η μορφή της σελίδας παρουσιάζεται στην εικόνα 46.



Εικόνα 46 Η μορφή της ηλεκτρονικής σελίδας που χρησιμοποιήθηκε στην εργασία
Όπως είναι φανερό, οι διακόπτες είναι σβηστοί στην εικόνα 46.

ΚΕΦΑΛΑΙΟ 8

Συμπεράσματα

Στόχος της εργασίας ήταν ο προγραμματισμός του μικροελεγκτή της πλακέτας ESP8266 Wifi Module Development Board για την ασύρματη σύνδεση και τη διαχείριση του ηλεκτρονόμου που διαθέτει η πλακέτα, ώστε να επιτυγχάνεται εξοικονόμηση χρόνου και διευκόλυνση στις καθημερινές ανάγκες που υπάρχουν αναφορικά με την ενεργοποίηση/απενεργοποίηση συσκευών, μηχανημάτων και γενικότερα ηλεκτρικών και ηλεκτρονικών συστημάτων.

Τα αποτελέσματα από το τελικό προϊόν είναι ικανοποιητικά, αφού επιτυγχάνεται ο αρχικός στόχος της εργασίας. Εντούτοις, υπάρχει δυνατότητα βελτίωσης του project αλλά και επιπλέον προτάσεις για παρόμοια project που διαθέτουν όμοιες βασικές αρχές.

Μία εναλλακτική πρόταση αναφορικά με το Hardware, αποτελεί η πλακέτα ESP-12E Wifi Development Board, η οποία διαθέτει: εσωτερική κεραία Wifi, ρυθμό μετάδοσης πληροφοριών (baudrate) 115200, μικρότερο μέγεθος και χαμηλότερη τιμή από την πλακέτα που χρησιμοποιήθηκε στην συγκεκριμένη εργασία.

Πρόταση για το μέλλον αποτελεί η ιδέα της σύνδεσης και χειρισμού όλων των γενικών εισόδων/εξόδων της πλακέτας με συμβατούς αισθητήρες και ενεργοποιητές που αναφέρθηκαν στο 4^ο κεφάλαιο, ανάλογα με τις ανάγκες των εκάστοτε εφαρμογών.

Όσον αφορά τον προγραμματισμό υπάρχουν πολλές ηλεκτρονικές πλατφόρμες, οι οποίες διαθέτουν έτοιμα παραδείγματα σε γλώσσες προγραμματισμού οδηγώντας έναν αρχάριο χρήστη με βασικές γνώσεις προγραμματισμού στην διεκπεραίωση αντίστοιχων projects. Οι μικροελεγκτές δίνουν δυνατότητες για αυτοματοποίηση και διευκόλυνση της καθημερινότητας σε πολλούς τομείς, ενώ παράλληλα σημειώνεται βελτίωση του περιβάλλοντος (μέσω του χειρισμού συσκευών εξ αποστάσεως μειώνονται οι άσκοπες μετακινήσεις για παράδειγμα).

Ο κόσμος των μικροελεγκτών γίνεται ολοένα και πιο ελκυστικός για εκατομμύρια ανθρώπους. Με τη συμβολή των μικροελεγκτών, γεννιούνται συνεχώς χαμηλού κόστους συστήματα αυτοματισμών, κυρίως από νέους ανθρώπους, την ίδια στιγμή που οι μεγάλες επιχειρήσεις διατηρούν πολύ υψηλό το κόστος των συστημάτων αυτών, συγκριτικά με αυτό που προσφέρουν.

Τέλος, η συγκεκριμένη εργασία αποτελεί, εκτός των υπολοίπων, ένα εγχειρίδιο κατασκευής ενός συστήματος απομακρυσμένου ελέγχου για οποιονδήποτε ενδιαφέρεται να κατασκευάσει μία παρόμοια εφαρμογή.

ΒΙΒΛΙΟΓΡΑΦΙΑ

Πεκμεστζή Κ., «Συστήματα Μικροϋπολογιστών», Τόμος 2^{ος}: Μικροελεγκτές AVR & PIC, Έκδοση 1^η, Εκδόσεις «Συμμετρία», έτος εκδόσεως: 2009

Santos Rui, «Home Automation Using ESP8266», Τόμος 2^{ος}

<https://el.wikipedia.org/wiki/%CE%9C%CE%B9%CE%BA%CF%81%CE%BF%CE%B5%CE%BB%CE%B5%CE%B3%CE%BA%CF%84%CE%AE%CF%82>

<https://en.wikipedia.org/wiki/ESP8266>

<https://el.wikipedia.org/wiki/UART>

<https://en.wikipedia.org/wiki/NodeMCU>

<https://blog.squix.org/>

<https://www.letscontrolit.com/wiki/index.php/ESPEasy>

https://espressif.com/sites/default/files/documentation/esp8266-technical_reference_en.pdf

<https://myesp8266.blogspot.gr/2015/03/other-interesting-board-is-this-board.html>

<https://el.wikipedia.org/wiki/%CE%97%CE%BB%CE%B5%CE%BA%CF%84%CF%81%CE%BF%CE%BD%CF%8C%CE%BC%CE%BF%CF%82>

https://el.wikipedia.org/wiki/IEEE_802.11

<http://randomnerdtutorials.com/esp8266-controlled-with-android-app-mit-app-inventor/>

<http://www.benlo.com/esp8266/esp8266QuickStart.html>