

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

**ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ 1534**

# **Αυτοματοποιημένο σύστημα ελέγχου ροής νερού βασισμένο στην πλατφόρμα Arduino.**

**ΨΥΛΛΑΣ ΔΗΜΗΤΡΙΟΣ-ΑΘΑΝΑΣΙΟΣ  
ΔΗΜΗΤΡΙΟΥ-ΚΟΜΜΑΤΑ ΚΩΝΣΤΑΝΤΙΝΑ**

**ΕΠΟΠΤΕΥΟΝ ΚΑΘΗΓΗΤΗΣ : ΚΑΡΕΛΗΣ ΔΗΜΗΤΡΙΟΣ**

**ΠΑΤΡΑ ΦΕΒΡΟΥΑΡΙΟΣ 2016**

## ΠΡΟΛΟΓΟΣ

Η ραγδαία εξέλιξη της τεχνολογίας τα τελευταία χρόνια, είχε σαν αποτέλεσμα την ανάπτυξη των υπολογιστικών συστημάτων σε μικρότερο μέγεθος και φυσικά μικρότερο κόστος. Επίσης, μέσω των δυνατοτήτων που μας παρέχονται μπορούμε να προσαρμόσουμε στο σύστημά μας τα δεδομένα και τα αποτελέσματα που εμείς έχουμε επιλέξει.

Οι σύγχρονες πλατφόρμες μας προσφέρουν λογισμικό ανοιχτού κώδικα και μας επιτρέπουν να κάνουμε τις κατάλληλες αλλαγές στο κυρίως πρόγραμμα όπου το κρίνουμε απαραίτητο. Αυτό σημαίνει, ότι μπορούμε να φτιάξουμε εφαρμογές που καλύπτουν ευρύ φάσμα από τις πιο απαιτητικές μέχρι τις πιο απλές.

Εξαιτίας του χαμηλού κόστους, είμαστε σε θέση να αποκτήσουμε έναν μικροελεγκτή και να τον χρησιμοποιήσουμε για το αντικείμενο που θέλουμε να υλοποιήσουμε. Επί προσθέτως, χάρη στα εργαλεία που είναι ενσωματωμένα στην ανάπτυξη των εφαρμογών των μικροελεγκτών ο προγραμματισμός γίνεται 'παιχνιδάκι'!

Επιπλέον, με την χρήση αισθητήρων, οι οποίοι διαθέτουν πολλές λειτουργίες μπορούμε να έχουμε ακρίβεια στα αποτελέσματα αλλά και να έχουμε γραφικές παραστάσεις έτσι ώστε να βγάζουμε χρήσιμα συμπεράσματα και να τα βελτιώνουμε.

## ΠΕΡΙΛΗΨΗ

Σε αυτή την πτυχιακή εργασία, θα παρουσιάσουμε την κατασκευή και τον έλεγχο ενός ολοκληρωμένου συστήματος αυτόματου ποτίσματος μέσω κατάλληλου λογισμικού-υλικού και με την βοήθεια αυτού θα γίνεται έλεγχος των αισθητήρων και της ροής του νερού. Κατά συνέπεια, θα φτάσουμε στον αρχικό μας στόχο, το πότισμα!

Αναλυτικότερα, θα αναφερθούμε και στους μικροελεγκτές καθώς η πλατφόρμα Arduino ανήκει στην οικογένεια τους. Το Arduino είναι μία πλακέτα ανοιχτού κώδικα, έχει αναλογικές και ψηφιακές εισόδους/εξόδους και προγραμματίζεται με την γλώσσα προγραμματισμού C++.

Η πλακέτα Arduino βασίζεται στον μικροελεγκτή Atmel AVR ο οποίος έχει διάφορες εκδόσεις. Χάρη στην εύκολη χρήση του, μπορεί να προγραμματιστεί και να χρησιμοποιηθεί σε περίπλοκα κυκλώματα. Με την βοήθεια των αισθητήρων και των ελεγχόμενων συσκευών (π.χ. αντλία νερού) μπορούμε να συλλέξουμε διάφορες τιμές και να καταλήξουμε στα επιθυμητά αποτελέσματα.

Ο προγραμματισμός του Arduino γίνεται μέσω ενός περιβάλλοντος ανάπτυξης (IDE). Μπορεί να χρησιμοποιηθεί σε όλα τα λειτουργικά συστήματα υπολογιστών αφού η εφαρμογή είναι γραμμένη σε JAVA. Αυτό έχει ως αποτέλεσμα ο προγραμματισμός και ο σχεδιασμός να γίνεται με ευκολία από το χρήστη.

Κατά συνέπεια η εφαρμογή μας, θα έχει σαν στόχο μέσω των αισθητήρων και των παραμέτρων που έχουμε εμείς ορίσει, να λειτουργεί είτε αυτόματα είτε χειροκίνητα. Από την άλλη πλευρά, μέσω αυτού του συστήματος θα ελέγχουμε την κατανάλωση νερού.

Τέλος για την πραγματική αναπαράσταση του συστήματός μας, θα πραγματοποιήσουμε κατασκευή.

Πίνακας περιεχομένων	
ΠΡΟΛΟΓΟΣ .....	2
ΠΕΡΙΛΗΨΗ .....	3
ΚΕΦΑΛΑΙΟ 1 .....	7
1.1 ΓΕΝΙΚΑ ΓΙΑ ΤΟΥΣ ΜΙΚΡΟΠΕΞΕΡΓΑΣΤΕΣ .....	8
1.2 ΣΧΕΤΙΚΑ ΜΕ ΤΟΥΣ ΜΙΚΡΟΠΕΞΕΡΓΑΣΤΕΣ .....	8
1.3 ΣΧΕΔΙΑΣΤΕΣ ΜΙΚΡΟΠΕΞΕΡΓΑΣΤΩΝ .....	9
1.4 ΕΙΣΑΓΩΓΗ ΣΤΟΥΣ ΜΙΚΡΟΕΛΕΓΚΤΕΣ .....	10
1.5 ΒΑΣΙΚΑ ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ ΜΙΚΡΟΕΛΕΓΚΤΩΝ .....	10
1.6 ΚΑΤΗΓΟΡΙΕΣ ΜΙΚΡΟΕΛΕΓΚΤΩΝ .....	11
1.7 ΑΠΟ ΤΙ ΑΠΟΤΕΛΕΙΤΑΙ ΕΝΑΣ ΜΙΚΡΟΕΛΕΓΚΤΗΣ .....	12
1.8 ΚΑΤΑΣΚΕΥΑΣΤΕΣ ΜΙΚΡΟΕΛΕΓΚΤΩΝ .....	16
1.9 ΣΥΓΚΡΙΣΗ ΜΙΚΡΟΕΛΕΓΚΤΗ ΜΕ ΜΙΚΡΟΠΕΞΕΡΓΑΣΤΗ .....	16
1.10 ΠΕΡΙΓΡΑΦΗ ΤΩΝ ΜΙΚΡΟΕΛΕΓΚΤΩΝ AVR .....	18
1.10.1 ΓΕΝΙΚΑ ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ ΜΙΚΡΟΕΛΕΓΚΤΩΝ .....	18
1.10.2 ΜΙΚΡΟΕΛΕΓΚΤΕΣ AVR .....	19
1.10.3 ΜΝΗΜΕΣ ΠΡΟΓΡΑΜΜΑΤΟΣ ΚΑΙ ΔΕΔΟΜΕΝΩΝ .....	22
1.10.4 ΜΝΗΜΗ SRAM .....	24
1.10.4.1 ΕΞΩΤΕΡΙΚΗ ΜΝΗΜΗ SRAM .....	25
1.10.4.2 ΕΣΩΤΕΡΙΚΗ ΜΝΗΜΗ SRAM .....	25
1.10.5 ΚΑΤΑΧΩΡΗΤΕΣ .....	26
1.10.6 ΑΡΙΘΜΗΤΙΚΗ ΚΑΙ ΛΟΓΙΚΗ ΜΟΝΑΔΑ .....	27
ΚΕΦΑΛΑΙΟ 2: ΣΧΕΤΙΚΑ ΜΕ ΤΟ ARDUINO .....	29
2.1 ΓΕΝΙΚΕΣ ΠΛΗΡΟΦΟΡΙΕΣ .....	30
2.2 ΠΛΕΟΝΕΚΤΗΜΑΤΑ ΤΟΥ ARDUINO .....	30
2.3 Εκδόσεις αναπτυξιακής πλατφόρμας Arduino .....	31
2.4 Arduino Shield .....	35
2.5 ARDUINO UNO .....	37
2.5.1 ΤΕΧΝΙΚΑ ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ .....	37
2.5.2 ΜΝΗΜΗ .....	38
2.5.3 ΤΡΟΦΟΔΟΣΙΑ .....	39
2.5.4 ΕΙΣΟΔΟΙ-ΕΞΟΔΟΙ .....	39
2.6 ARDUINO IDE .....	42
2.6.1 ΔΟΜΗ ΚΑΙ ΓΛΩΣΣΑ ΠΡΟΓΡΑΜΜΑΤΟΣ .....	45
2.6.2 ΒΙΒΛΙΟΘΗΚΕΣ .....	49

2.7 Fritzing.....	50
ΚΕΦΑΛΑΙΟ 3:ΕΡΓΑΛΕΙΑ ΠΟΥ ΘΑ ΧΡΗΣΙΜΟΠΟΙΗΣΟΥΜΕ ΓΙΑ ΤΗΝ ΕΦΑΡΜΟΓΗ ΜΑΣ	53
3.1 ΠΕΡΙΓΡΑΦΗ ΤΟΥ ΣΥΣΤΗΜΑΤΟΣ .....	54
3.2 ΠΕΡΙΓΡΑΦΗ ΤΩΝ ΑΙΣΘΗΤΗΡΩΝ.....	54
3.3 ΠΕΡΙΓΡΑΦΗ ΑΙΣΘΗΤΗΡΩΝ ΠΕΡΙΒΑΛΛΟΝΤΟΣ .....	58
3.3.1 Αισθητήρας υγρασίας χώματος .....	59
3.3.2 Αισθητήρας βροχής.....	61
3.3.3 Συσκευές ελέγχου .....	63
ΚΕΦΑΛΑΙΟ 4: ΠΕΡΙΓΡΑΦΗ ΣΥΝΔΕΣΕΩΝ ΤΗΣ ΕΦΑΡΜΟΓΗΣ ΜΑΣ. ....	67
4.1 ΠΕΡΙΓΡΑΦΗ ΕΦΑΡΜΟΓΗΣ .....	68
4.2 ΑΠΑΙΤΗΣΕΙΣ ΤΟΥ ΣΥΣΤΗΜΑΤΟΣ.....	68
4.3 ΚΟΣΤΟΣ ΥΛΙΚΩΝ- ΚΑΤΑΣΚΕΥΗ ΜΑΚΕΤΑΣ .....	69
4.4 ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΚΑΙ ΣΧΕΔΙΑ ΓΙΑ ΤΗΝ ΥΛΟΠΟΙΗΣΗ ΤΟΥ ΣΥΣΤΗΜΑΤΟΣ .....	69
4.5 ΜΕΛΛΟΝΤΙΚΕΣ ΑΛΛΑΓΕΣ ΤΟΥ ΣΥΣΤΗΜΑΤΟΣ ΜΑΣ.....	70
4.6 ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΕΦΑΡΜΟΓΗΣ .....	71
ΒΙΒΛΙΟΓΡΑΦΙΑ .....	74

## Πίνακας σχημάτων-πινάκων-εικόνων

### ΣΧΗΜΑΤΑ

Σχήμα 1.1: Οι βαθμίδες ενός μικροϋπολογιστικού συστήματος .....	8
Εικόνα 1.2: Ολοκληρωμένα chip.....	9
Σχήμα 1.3: Σύνδεση μικροελεγκτή με εξωτερικές διατάξεις.....	13
Σχήμα 1.4: Ένας 8-bit μικροελεγκτής.....	15
Σχήμα 1.5: Σύνδεση της κεντρικής μονάδας επεξεργασίας με εξωτερικές διατάξεις.....	19
Σχήμα 1.6: Αρχιτεκτονική διάταξη περιφερειακών μονάδων και μικροεπεξεργαστών AVR.....	21
Σχήμα 1.7: Τύποι από τις μνήμες των μικροελεγκτών AVR.....	23
Σχήμα 1.8: Ταυτόχρονη εκτέλεση και ανάκληση των εντολών.....	23
Σχήμα 1.9: Διαδικασία εκτέλεσης εντολής στην Αριθμητική και λογική μονάδα.....	24
Σχήμα 1.10: Φάσεις ανάγνωσης- εγγραφής στο εσωτερικό της μνήμης SRAM.....	26
Σχήμα 1.11: Ένα παράδειγμα αριθμητικής και λογικής μονάδας των 2-bit που εκτελεί τις λογικές πράξεις AND, OR, XOR.....	28
Εικόνα 2.1: Arduino Leonardo.....	31
Εικόνα 2.2: Arduino Uno Rev3.....	32
Εικόνα 2.3: Arduino Mega 2560 Rev3.....	32
Εικόνα 2.4: Arduino Due.....	33
Εικόνα 2.5: Arduino Esplora.....	33
Εικόνα 2.6: Arduino Duemilanove.....	34
Εικόνα 2.7: Arduino Nano.....	34
Εικόνα 2.8: Arduino Lilypad.....	35

Εικόνα 2.9: Arduino NG.....	35
Εικόνα 2.10: Ethernet shield.....	36
Εικόνα 2.11: Motor shield.....	36
Εικόνα 2.12: GSM shield.....	36
Εικόνα 2.13: wifi shield.....	37
Εικόνα 2.14: Relay shield .....	37
Εικόνα 2.15: ATMEGA 328, 8-bit μικροελεγκτή .....	38
Εικόνα 2.16: Είσοδοι- έξοδοι της τροφοδοσίας .....	39
Εικόνα 2.17: Ακροδέκτες .....	40
Σχήμα 2.18: Σχηματικό διάγραμμα αρχιτεκτονικής Arduino .....	41
Εικόνα 2.19: Ολοκληρωμένο περιβάλλον ανάπτυξης.....	43
Εικόνα 2.20: Πρόγραμμα .....	44
Εικόνα 2.21: Αρχική σελίδα Arduino IDE .....	46
Εικόνα 2.22: Αρχική σελίδα για breadboard.....	50
Εικόνα 2.23: Αρχική σελίδα τυπωμένου PCB .....	51
Εικόνα 2.24: Αρχική σελίδα για τον κώδικα .....	51
Σχήμα 3.1: Σχηματική διάταξη ενός αισθητήρα .....	54
Σχήμα 3.2: Χαρακτηριστική υστέρησης.....	55
Σχήμα 3.3: Γραφική παράσταση εξόδου ως προς είσοδο του αισθητήρα .....	56
Σχήμα 3.4: Χαρακτηριστική νεκρής ζώνης .....	57
Σχήμα 3.5: Χαρακτηριστική σφάλματος επαναληψιμότητας .....	57
Σχήμα 3.6: Χαρακτηριστική κόρου .....	58
Εικόνα 3.7: Αισθητήρας υγρασίας χώματος .....	59
Σχήμα 3.8: Σύνδεση αισθητήρα υγρασίας χώματος.....	59
Εικόνα 3.9: Αισθητήρας βροχής.....	61
Σχήμα 3.10: Σύνδεση αισθητήρα βροχής.....	61
Εικόνα 3.11: Relay .....	63
Εικόνα 3.12: Αισθητήρες ελέγχου ροής νερού.....	64
Σχήμα 3.13: Σύνδεση αισθητήρα ελέγχου ροής νερού.....	65
Εικόνα 3.14: Ηλεκτροβάννα.....	66
Εικόνα 3.15: Εσωτερικό ηλεκτροβάννας.....	66
Σχήμα 4.2: Πλήρης σύνδεση του συστήματος.....	70

## ΠΙΝΑΚΕΣ

Πίνακας 1.1: Διάφοροι τύποι μικροελεγκτών AVR.....	20
Πίνακας 4.1: Υλικά και κόστος .....	69

## **ΚΕΦΑΛΑΙΟ 1: ΜΙΚΡΟΕΛΕΓΚΤΕΣ ΚΑΙ ΜΙΚΡΟΕΠΕΞΕΡΓΑΣΤΕΣ**

## 1.1 ΓΕΝΙΚΑ ΓΙΑ ΤΟΥΣ ΜΙΚΡΟΕΠΕΞΕΡΓΑΣΤΕΣ

Οι μικροϋπολογιστές είναι μία πολύ σημαντική ανακάλυψη. Αποτελούν το κύριο μέρος ενός ηλεκτρονικού υπολογιστή αλλά και πολλών ηλεκτρικών συσκευών. Περιέχουν τα παρακάτω μέρη:

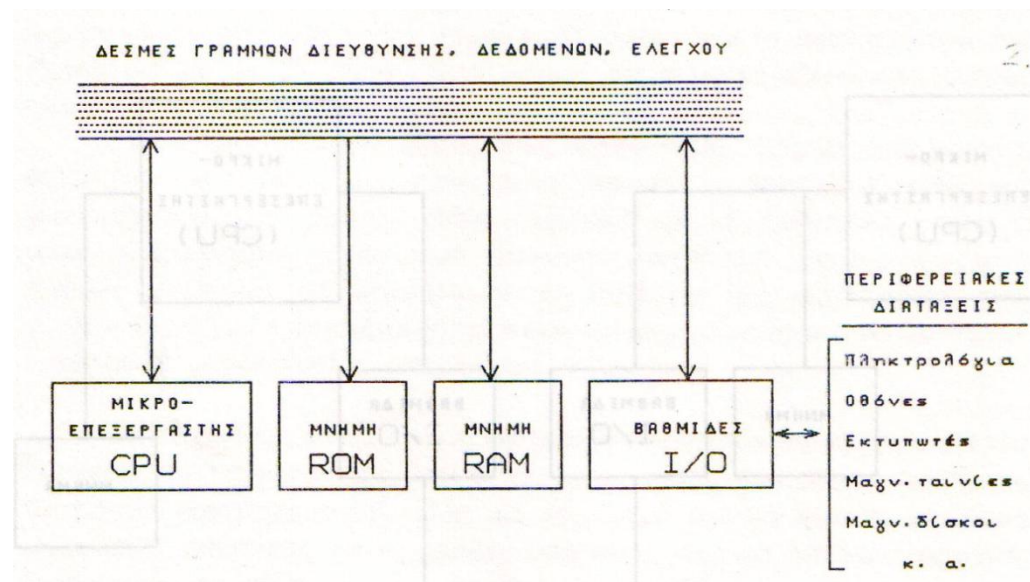
- ✓ Την κεντρική μονάδα επεξεργασίας (CPU)
- ✓ Τις μνήμες (RAM,ROM,Flash)
- ✓ Τις εισόδους/εξόδους που μας χρησιμεύουν στην σύνδεση των εξωτερικών περιφερειακών συσκευών όπως το πληκτρολόγιο, η οθόνη, ο εκτυπωτής και άλλα.

Όλα αυτά τα στοιχεία είναι τοποθετημένα πάνω σε μία πλακέτα. Οι μικροϋπολογιστές είναι το κύριο υπολογιστικό σύστημα, πάνω στο οποίο αναπτύχθηκαν οι ηλεκτρονικοί υπολογιστές και στην συνέχεια τα σύγχρονα υπολογιστικά συστήματα.

## 1.2 ΣΧΕΤΙΚΑ ΜΕ ΤΟΥΣ ΜΙΚΡΟΕΠΕΞΕΡΓΑΣΤΕΣ

Η κεντρική μονάδα επεξεργασίας ή πιο απλά ο μικροεπεξεργαστής είναι η καρδιά ενός ηλεκτρονικού υπολογιστή. Οι πρώτοι επεξεργαστές εμφανίστηκαν περίπου το 1970. Η εξέλιξή τους ήταν απότομη, αφού αυξήθηκαν οι απαιτήσεις από τις γλώσσες υψηλού επιπέδου.

Ο μικροεπεξεργαστής επικοινωνεί με την διάταξη εισόδου/εξόδου και την μνήμη μέσω γραμμών επικοινωνίας που ονομάζονται buses. Η κεντρική μονάδα επεξεργασίας έχει τον κύριο έλεγχο. Μέσω της δέσμης των γραμμών επικοινωνίας, μεταφέρονται όλα τα σήματα έτσι ώστε να εκτελεστεί μία εντολή. Η μνήμη και οι εισοδοί/έξοδοι δεν δίνουν εντολές στην CPU αλλά μόνο δέχονται από αυτή. Όλα αυτά μπορούμε να τα δούμε στο ακόλουθο σχήμα.



**ΣΧΗΜΑ 1.1:** Οι βαθμίδες ενός μικροϋπολογιστικού συστήματος.

Οι σύγχρονοι μικροεπεξεργαστές αποτελούνται από:

- Την Αριθμητική και Λογική Μονάδα (Arithmetic and Logical Unit): Είναι η μονάδα στην οποία εκτελούνται οι αριθμητικές ή λογικές πράξεις, από τις εντολές που έχουν δοθεί στον υπολογιστή.



- Τον Αποκωδικοποιητή Εντολών (Command Decoder): Είναι χρήσιμος για να γίνει η μετατροπή μίας εντολής προγράμματος σε μία σειρά λειτουργιών.
- Τους Καταχωρητές (Registers): Είναι μικρά κελιά μνήμης που βρίσκονται στο εσωτερικό του επεξεργαστή και χρησιμοποιούνται για την προσωρινή αποθήκευση των δεδομένων καθώς εμείς τα επεξεργαζόμαστε.
- Τους Εσωτερικούς Διαύλους Επικοινωνίας (Internal Bus): Είναι απαραίτητοι για την επικοινωνία όλων των μονάδων μέσα στην κεντρική μονάδα επεξεργασίας.
- Τα Κυκλώματα Χρονισμού και Ελέγχου (Timing and Control): Έχοντας ψηφιακό ρολόι και ψηφιακούς απαριθμητές παράγουν διάφορα σήματα για να ελέγχουν την σωστή λειτουργία της εκτέλεσης των εντολών μέσα σε ένα πρόγραμμα.
- Την Διασύνδεση των Εξωτερικών Διαύλων με τους Εσωτερικούς Διαύλους (Data Address/Control Buses Interfaces) για την επικοινωνία της CPU με τα υπόλοιπα κομμάτια του μικροϋπολογιστή.

Η κεντρική μονάδα επεξεργασίας ξεκινά να λειτουργεί αφού συνεργαστούν όλες οι προηγούμενες μονάδες και βρίσκεται όλη μέσα σε ένα ολοκληρωμένο chip. Στην παρακάτω εικόνα βλέπουμε πως είναι αυτά τα chip.



**ΕΙΚΟΝΑ 1.2:** Ολοκληρωμένα chip.

### 1.3 ΣΧΕΔΙΑΣΤΕΣ ΜΙΚΡΟΕΠΕΞΕΡΓΑΣΤΩΝ

Η εξέλιξη των μικροεπεξεργαστών, μας θυμίζει την αντίστοιχη εξέλιξη των υπολογιστών. Αυτό σημαίνει πως οι μεσαίοι υπολογιστές εξελίχθηκαν σε μεγάλα υπολογιστικά συστήματα άρα και οι μικροεπεξεργαστές έκαναν δικά τους πολλά στοιχεία της αρχιτεκτονικής των μεγάλων συστημάτων. Στους πρόσφατους μικροεπεξεργαστές τα κοινά σημεία με τους υπολογιστές ήταν πάρα πολλά έτσι δεν ήταν εύκολος ο διαχωρισμός τους. Οι πιο γνωστές εταιρίες μικροεπεξεργαστών είναι οι ακόλουθες:

- ❖ Advanced Micro Devices: Κατασκευάζουν επεξεργαστές για προσωπικούς υπολογιστές.
- ❖ ARM Ltd: Δεν κατασκευάζει επεξεργαστές όμως παραχωρεί δικαιώματα κατασκευής επεξεργαστών σε τρίτους. Η αρχιτεκτονική ARM είναι από τις πλέον δημοφιλείς για τα ενσωματωμένα συστήματα.
- ❖ Freescale Semiconductor: Σχεδιάζουν επεξεργαστές για PowerPC.
- ❖ IBM Microelectronics: Αναπτύσσουν επεξεργαστές που προορίζονται για Power υπολογιστές και για βιντεοπαιχνίδια.
- ❖ Intel Corp: Κατασκευάζουν μικροεπεξεργαστές όπως 8051, IA-32, IA-64 και XScale. Επίσης, παράγει chip για χρήση με τους επεξεργαστές της.

- ❖ MIPS Technologies: Η αρχιτεκτονική τους βασίζεται στην σχεδίαση των RISC επεξεργαστών.
- ❖ NEC electronics: Έχουν αναπτύξει την αρχιτεκτονική 78K 8-bit, την αρχιτεκτονική 78K0R 16-bit και την αρχιτεκτονική V850 32-bit.
- ❖ Sun Microsystems: Ανέπτυξαν την αρχιτεκτονική SPARC.
- ❖ Texas Instruments: Κατασκευάζουν μικροελεγκτές χαμηλής κατανάλωσης.
- ❖ Transmeta: Επίσης, κατασκευάζουν μικροελεγκτές χαμηλής κατανάλωσης και είναι συμβατοί με το σετ εντολών x86.
- ❖ Atmel: Κατασκευάζουν τους μικροελεγκτές CISC και RISC χαμηλού κόστους όπως η σειρά AVR.

## 1.4 ΕΙΣΑΓΩΓΗ ΣΤΟΥΣ ΜΙΚΡΟΕΛΕΓΚΤΕΣ

Ένας μικροελεγκτής είναι ένα ολοκληρωμένο κύκλωμα που συνήθως είναι μέρος ενός συστήματος. Μία άλλη ερμηνεία του, είναι ο χαρακτηρισμός του ως παραλλαγή ενός μικροεπεξεργαστή ο οποίος λειτουργεί με ελάχιστα εξωτερικά εξαρτήματα εξαιτίας των πολλών ενσωματωμένων υποσυστημάτων που διαθέτει. Περιλαμβάνει θύρες εισόδου/εξόδου, CPU, RAM, ROM, timers σαν έναν απλό υπολογιστή. Όμως, επειδή έχουν σχεδιαστεί για να εκτελούν μόνο μία συγκεκριμένη εργασία για τον έλεγχο ενός απλού συστήματος, έχουν σχεδιαστεί πολύ απλά έτσι ώστε να περιλαμβάνουν όλες τις λειτουργίες σε ένα μόνο σύστημα. Οι μικροελεγκτές χρησιμοποιούνται ευρύτατα σε όλα τα ενσωματωμένα συστήματα ελέγχου, χαμηλού και μεσαίου κόστους όπως για παράδειγμα σε αυτοματισμούς και σε ηλεκτρονικές συσκευές.

Πιο συγκεκριμένα, ο μικροελεγκτής είναι ένας υπολογιστής χωρίς περιφερειακά. Είναι μία κατηγορία επεξεργαστή που είναι πολύ συχνή στις τηλεπικοινωνίες και στη σύγχρονη βιομηχανία. Επίσης, παρατηρούμε πως ευθύνεται για την είσοδο και την έξοδο, την αποθήκευση την επεξεργασία και την μετάδοση των αναλογικών και ψηφιακών σημάτων μίας εφαρμογής.

Στην συνέχεια θα εξηγήσουμε πιο αναλυτικά πόσο σημαντικός είναι ένας μικροελεγκτής σε ένα μικροϋπολογιστικό σύστημα. Εάν λοιπόν έχουμε μόνο μία μονάδα κεντρικής επεξεργασίας (CPU), για να κατασκευάσουμε ένα σύστημα που να μπορεί να συνδεθεί με διάφορες εξωτερικές διατάξεις, θα χρειαστούμε κάποιες περιφερειακές διατάξεις που είναι αναγκαίες για την σύνδεση της CPU με τις εξωτερικές μονάδες δηλαδή αισθητήρες, οθόνες απεικόνισης κ.λπ. Μία τέτοια σύνδεση θα ήταν πολύ δύσκολη αφού τα εξαρτήματα που θα χρειαζόμασταν θα ήταν πάρα πολλά.

## 1.5 ΒΑΣΙΚΑ ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ ΜΙΚΡΟΕΛΕΓΚΤΩΝ

Ένας μικροελεγκτής απαρτίζεται από κάποια χαρακτηριστικά που τον ξεχωρίζουν για την χρήση του σε εφαρμογές και τον καθιστούν προτιμότερο έναντι της χρήσης των επιμέρους στοιχείων που είναι ξεχωριστά (επεξεργαστής, μνήμες, συσκευές εισόδου- εξόδου). Παρακάτω θα αναλύσουμε κάποια πλεονεκτήματα:

- Χαμηλό κόστος: Είναι ένα βασικό χαρακτηριστικό που κάποιος λαμβάνει υπόψη του. Με την συνεχή εξέλιξη της τεχνολογίας έχουμε βελτίωση των μικροελεγκτών αλλά και μείωση των τιμών.
- Αυτονομία : Αυτό γίνεται μέσω της ενσωμάτωσης των σύνθετων περιφερειακών υποσυστημάτων. Έτσι πολλοί μικροελεγκτές δεν χρειάζονται κανένα άλλο ολοκληρωμένο κύκλωμα για να λειτουργήσουν.

- Μικρότερο μέγεθος: Η ολοκλήρωση των βασικών στοιχείων από τα οποία αποτελείται μείωσε τις διαστάσεις.
- Χαμηλή κατανάλωση ισχύος: Το ότι οι μικροελεγκτές λειτουργούν σε χαμηλές συχνότητες που φθάνουν τα 32kHz, μας οδηγεί στο συμπέρασμα ότι καταναλώνουν μικρά ποσά ισχύος της τάξης των mW ή ακόμα και των μW. Επιπλέον, έχουν την δυνατότητα να έρχονται σε κατάσταση αναμονής και να καταστέλλουν προσωρινά την λειτουργία της κεντρικής μονάδας επεξεργασίας και των περιφερειακών, οπότε αυτό μπορεί να γίνει μειώνοντας πολύ την κατανάλωση ισχύος του μικροελεγκτή.
- Μετρήσεις σε πραγματικό χρόνο: Τη στιγμή που οι ηλεκτρονικοί υπολογιστές χρειάζεται να τρέξουν λειτουργικά συστήματα πραγματικού χρόνου για να πετύχουν μετρήσεις, οι μικροελεγκτές δεν απαιτούν επιπλέον λογισμικό.
- Μειωμένες εκπομπές ηλεκτρομαγνητικών παρεμβολών και μειωμένη ευαισθησία σε αντίστοιχες παρεμβολές από άλλες ηλεκτρικές συσκευές: Αυτό προκύπτει από το μικρότερο αριθμό και μήκος των εξωτερικών διασυνδέσεων όπως επίσης και των χαμηλότερων ταχυτήτων λειτουργίας.
- Περισσότεροι διαθέσιμοι ακροδέκτες για ψηφιακές εισόδους/εξόδους: Για το δεδομένο μέγεθος του ολοκληρωμένου κυκλώματος λόγω της μη δέσμευσής τους για την σύνδεση εξωτερικών περιφερειακών.
- Εύκολη υλοποίηση εφαρμογών: Έχουν απλές διασυνδέσεις άρα και μεγαλύτερη αξιοπιστία.

Παρόλα τα παραπάνω πλεονεκτήματα οι μικροελεγκτές εμφανίζουν και κάποια μειονεκτήματα τα οποία είναι:

- Δεν γίνεται η αλλαγή του προγράμματος επειδή είναι γραμμένο στην μνήμη ROM.
- Προγραμματίζεται με δυσκολία.
- Ο χρόνος ανάπτυξής του είναι μεγάλος.

## 1.6 ΚΑΤΗΓΟΡΙΕΣ ΜΙΚΡΟΕΛΕΓΚΤΩΝ

Λόγω της ενσωμάτωσης των μικροελεγκτών σε κάθε ηλεκτρική συσκευή αλλά και σε συνδυασμό με τον ανταγωνισμό, υπάρχουν στην παραγωγή ανταγωνιστικά μοντέλα για πολύ εξειδικευμένες εφαρμογές. Έτσι, διακρίνονται οι παρακάτω κατηγορίες:

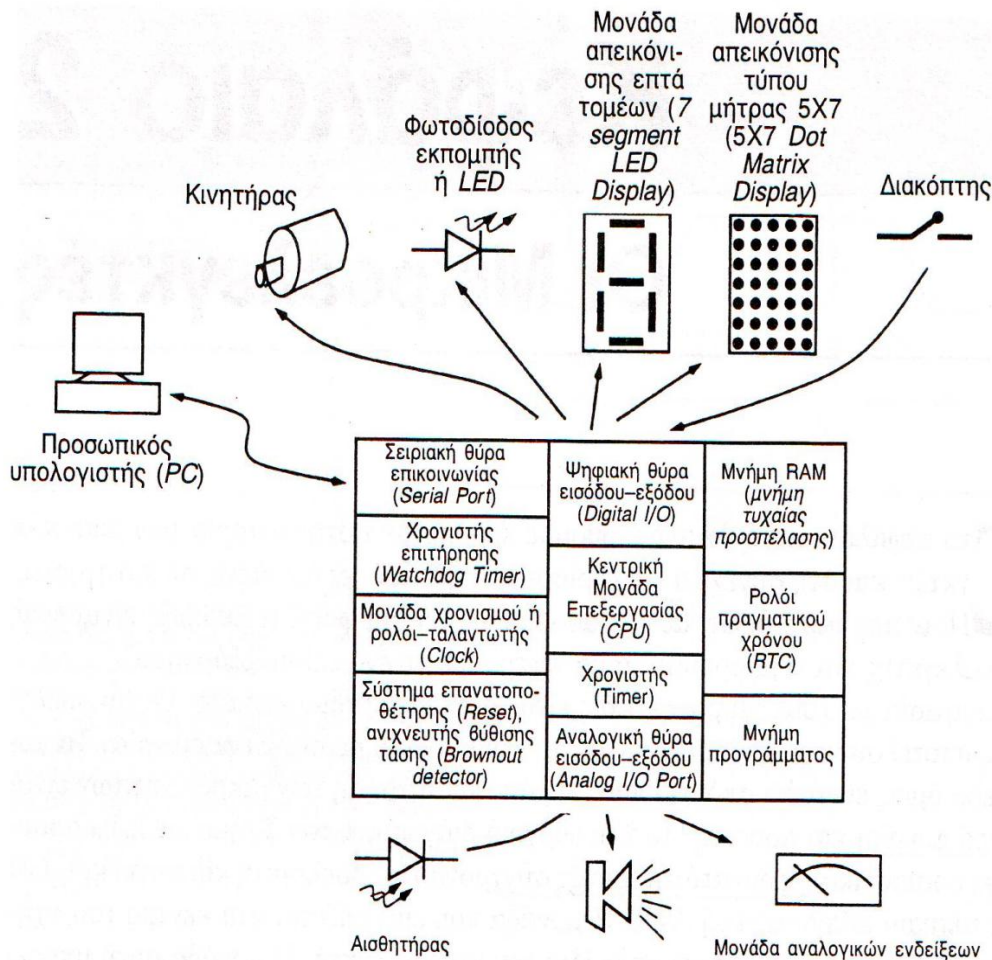
- ↵ Μικροελεγκτές (συνήθως 8-bit μερικές φορές και 4-bit) χαμηλού κόστους, γενικής χρήσης, με μικρό αριθμό ακροδεκτών(λιγότερους από 8). Για να μην αντιγράφεται εύκολα το εσωτερικό λογισμικό τους σχεδιάζονται έτσι ώστε να έχουν χαμηλή κατανάλωση και αυτάρκεια. Δεν υπάρχει η δυνατότητα επέκτασης της μνήμης τους. Μερικά μοντέλα είναι ευρέως γνωστά, όπως πχ οι μικροελεγκτές των σειρών PIC(Microchip), AVR(Atmel), 8051(Intel,Atmel,Dallas).
- ↵ Μικροελεγκτές (συνήθως 8-bit αλλά και 16 ή 32-bit) χαμηλού κόστους, γενικής χρήσης, με μέτριο έως μεγάλο αριθμό ακροδεκτών. Διαθέτουν μεγάλο αριθμό κοινών περιφερειακών, όπως θύρες UART, I<sup>2</sup>C, SPI ή CAN, μετατροπείς αναλογικού σε ψηφιακό και ψηφιακού σε αναλογικό. 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 αυτόνομη περιφερειακών και μεγάλες δυνατότητες εσωτερικής ή εξωτερικής μνήμης προγράμματος (FLASH) και RAM. Στο χώρο αυτό έχουν ισχυρή παρουσία οι αρχιτεκτονικές με υψηλή μεταφερισιμότητα λογισμικού (portability) από τον ένα στον άλλο κατασκευαστή. Πχ 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 Modem.

Οι μικροελεγκτές που πωλούνται αφορούν αυτούς των 8-bit, αφού είναι η κατηγορία με το χαμηλότερο κόστος και το μικρότερο μέγεθος λογισμικού επειδή οι οικογένειες των σύγχρονων 8-bit μικροελεγκτών έχουν βελτιωμένες επιδόσεις σε σχέση με το παρελθόν.

## 1.7 ΑΠΟ ΤΙ ΑΠΟΤΕΛΕΙΤΑΙ ΕΝΑΣ ΜΙΚΡΟΕΛΕΓΚΤΗΣ

Στο παρακάτω σχήμα που ακολουθεί εμφανίζονται οι δυνατότητες ενός μικροελεγκτή AVR. Η μονάδα που εμφανίζεται στο κέντρο του σχήματος αντιστοιχεί στον ίδιο τον μικροελεγκτή. Μπορεί να συνδεθεί κατάλληλα με ποικιλία διατάξεων απεικόνισης (οθόνη) ως συσκευή εξόδου, να έρθει σε επικοινωνία με έναν υπολογιστή, να διαβάσει διάφορες τιμές από εξωτερικούς αισθητήρες και επιπλέον μπορεί να συνδεθεί σε ένα είδος τοπικού δικτύου κάποιων άλλων μικροελεγκτών. Παρατηρούμε πως όλες αυτές οι δυνατότητες, δεν απαιτούν μεγάλο αριθμό εξωτερικών εξαρτημάτων. Αυτός ο λόγος οδηγεί σε πιο αξιόπιστα συστήματα με σχετικά χαμηλό κόστος εξαιτίας του μικρού αριθμού των εξωτερικών εξαρτημάτων και γενικότερα των λιγότερων συνδέσεων.



**ΣΧΗΜΑ 1.3:** Σύνδεση μικροελεγκτή με εξωτερικές διατάξεις.

Εδώ θα παρουσιάσουμε αναλυτικά τα μέρη από τα οποία αποτελείται ένας μικροελεγκτής (έχουν αναφερθεί επιγραμματικά στην εισαγωγή του κεφαλαίου):

- ✓ **Μονάδα Κεντρικής Επεξεργασίας – CPU (Central Processing Unit):** αποτελεί την «καρδιά» ενός μικροελεγκτή αφού εκτελεί την ανάκληση δεδομένων (fetch) από τη μνήμη προγράμματος υπό μορφή εντολών, ενώ παράλληλα αποκωδικοποιεί τις εντολές και στη συνέχεια τις εκτελεί. Η μονάδα CPU, αποτελείται από καταχωρητές (registers), την αριθμητική λογική μονάδα (ALU – Arithmetic Logic Unit), τον αποκωδικοποιητή εντολών (Instruction Decoder) κι διάφορα κυκλώματα ελέγχου.
- ✓ **Μνήμη Προγράμματος:** σε αυτήν αποθηκεύονται οι εντολές που σχηματίζουν τον κορμό του προγράμματος. Διακρίνεται στην εσωτερική και την εξωτερική μνήμη προγράμματος.
- ✓ **Μνήμη RAM:** η μνήμη τυχαίας προσπέλασης αποτελεί τη μνήμη δεδομένων του ελεγκτή, γεγονός το οποίο σημαίνει ότι χρησιμοποιείται από τον ελεγκτή για την αποθήκευση δεδομένων. Η CPU, χρησιμοποιεί τη μνήμη RAM για την αποθήκευση μεταβλητών, καθώς επίσης και τη λεγόμενη Στοίβα (Stack). Η Στοίβα με τη σειρά της χρησιμοποιείται από τη CPU, για την προσωρινή αποθήκευση των λεγόμενων διευθύνσεων επιστροφής, με σκοπό να συνεχίσει την εκτέλεση ενός

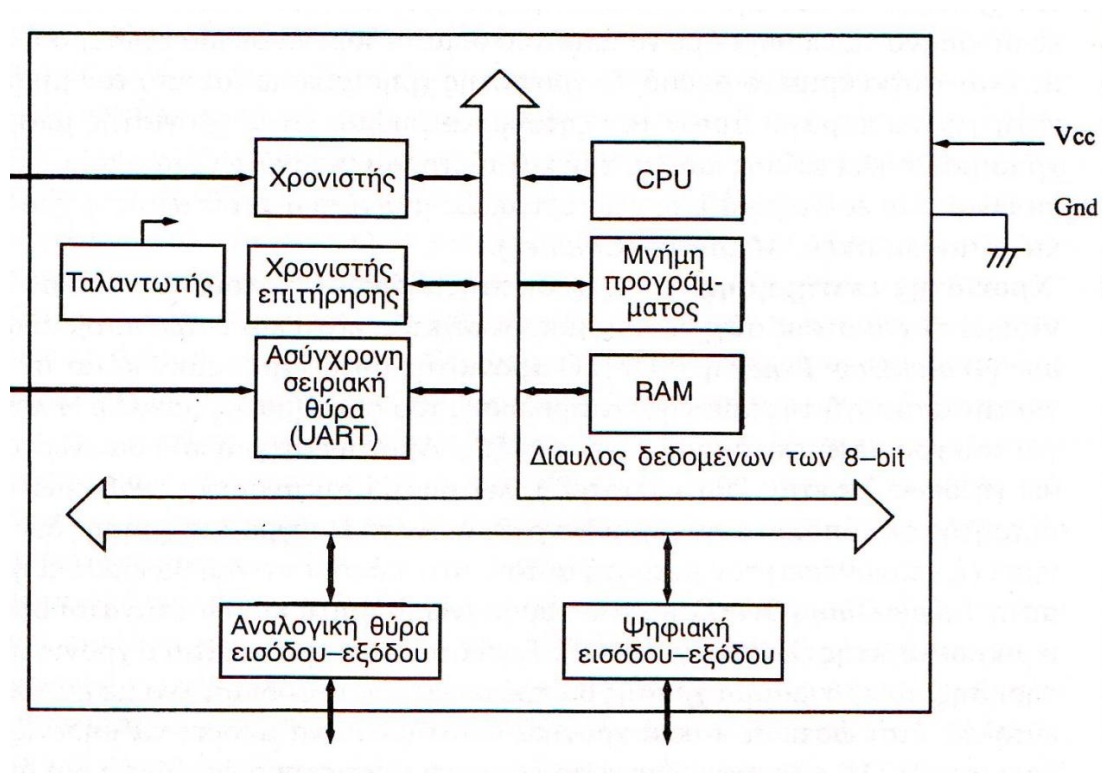
προγράμματος το οποίο έχει διακοπεί.

- ✓ Ταλαντωτής Χρονισμού: ένα πρόγραμμα εκτελείται από το μικροελεγκτή με έναν καθορισμένο ρυθμό. Ο ρυθμός αυτός καθορίζεται από τη συχνότητα λειτουργίας του ταλαντωτή χρονισμού, ο οποίος μπορεί να είναι ένας εσωτερικός ταλαντωτής τύπου RC ή ένας ταλαντωτής που υλοποιείται με κάποιο εξωτερικό στοιχείο χρονισμού, όπως για παράδειγμα ένας κρύσταλλος χαλαζία, ένα κύκλωμα συντονισμού LC ή ακόμη και ένα απλό κύκλωμα RC. Η λειτουργία του ταλαντωτή ξεκινά αμέσως μετά την εφαρμογή της τάσης τροφοδοσίας.
- ✓ Σύστημα επανατοποθέτησης και Κύκλωμα ανίχνευσης βύθισων τάσης: το σύστημα επανατοποθέτησης ή μηδενισμού ή απλά Reset, εξασφαλίζει το γεγονός ότι όλες οι εσωτερικές μονάδες και τα κυκλώματα ελέγχου του μικροελεγκτή θα ξεκινήσουν να λειτουργούν κατά την εφαρμογή της τροφοδοσίας, από κάποια προκαθορισμένη αρχική κατάσταση, ενώ όλοι οι καταχωρητές του συστήματος βρίσκονται σε κατάλληλες αρχικές τιμές. Το Κύκλωμα ανίχνευσης βύθισης της τάσης τροφοδοσίας είναι ένα επίσης εσωτερικό κύκλωμα ελέγχου το οποίο παρακολουθεί συνεχώς το επίπεδο της τάσης τροφοδοσίας και εφόσον ανιχνευτεί κάποια στιγμιαία βύθιση στην τάση αυτή τότε αυτόματα θέτει τον μικροελεγκτή σε λειτουργία επανατοποθέτησης, έτσι ώστε να προστατευθούν τα περιεχόμενα των καταχωρητών και της μνήμης από πιθανή καταστροφή ή αλλοίωση, πράγμα που θα οδηγούσε τον μικροελεγκτή σε εσφαλμένη λειτουργία.
- ✓ Σειριακή Θύρα Επικοινωνίας: εξυπηρετεί διαμεσολαβητικό σκοπό καθώς η λειτουργία της βασίζεται στο ότι λαμβάνει δεδομένα από το μικροελεγκτή τα οποία ολισθαίνει προς την έξοδο υπό μορφή ενός δυαδικού στοιχείου bit τη φορά. Παρομοίως, λαμβάνει δεδομένα από την αντίστοιχη είσοδό της και πάλι με τη μορφή ενός bit τη φορά, σχηματίζοντας έτσι με 8 τέτοια bits μια λέξη του 1 byte, την οποία και αντιγράφει στο εσωτερικό του ελεγκτή. Μπορεί να λειτουργήσει σε οποιαδήποτε ταχύτητα μετάδοσης δεδομένων τυχόν απαιτηθεί.
- ✓ Ψηφιακή Θύρα Εισόδου – Εξόδου: χρησιμοποιούνται για την ανταλλαγή δεδομένων από και προς το εξωτερικό περιβάλλον. Τα δεδομένα ανταλλάσσονται υπό τη μορφή ομάδων των 8 bits ή του 1 byte.
- ✓ Αναλογική θύρα εισόδου – εξόδου: η ύπαρξη αναλογικών εισόδων προϋποθέτει Μετατροπείς Αναλογικού Σήματος σε Ψηφιακό. Ένας μικροελεγκτής είναι δυνατόν να διαθέτει μία ενσωματωμένη μονάδα μετατροπής, η οποία χρησιμοποιείται για την ανάγνωση δεδομένων από αισθητήρες, όπως για παράδειγμα αισθητήρες πίεσης και θερμοκρασίας. Παράλληλα, μπορούμε να έχουμε και αναλογικές εξόδους, χρησιμοποιώντας κάποιες μονάδες οι οποίες καλούνται Μετατροπείς Ψηφιακού Σήματος σε Αναλογικό. Οι μετατροπείς αυτοί χρησιμοποιούνται για την οδήγηση κινητήρων, ειδικών μονάδων απεικόνισης, για την αναπαραγωγή σημάτων ήχου ή μουσικής και αλλού.
- ✓ Χρονιστής (Timer) χρησιμοποιείται για το χρονισμό ή τη σηματοδότηση διαφόρων γεγονότων, όπως για παράδειγμα για να παράγει το ζητούμενο ρυθμό που απαιτείται προκειμένου να καταφέρουμε να στείλουμε επιτυχώς δεδομένα σε μια εξωτερική οθόνη. Μπορεί να χρησιμοποιηθεί επίσης και για την καταμέτρηση γεγονότων, τα οποία μπορούν να είναι είτε εσωτερικά είτε εξωτερικά.
- ✓ Χρονιστής Επιτήρησης (Watchdog Timer): χρησιμοποιείται προκειμένου να αποφευχθεί μια πιθανή κατάρρευση του συστήματος (crash), οπότε και ενεργοποιείται η λειτουργία ενός αυξανόμενα εσωτερικού μετρητή σε συγκεκριμένο

ρυθμό. Αν το πρόγραμμα χρήσης δε μηδενίσει (ή επαναθέσει το μετρητή αυτό) τότε κάποια στιγμή θα επέλθει η λεγόμενη υπερχείλιση του παραπάνω μετρητή και θα επανατοποθετηθεί ο ελεγκτής σε λειτουργία reset. Η λογική αυτής της τεχνικής ελέγχου στηρίζεται στην υπόθεση ότι αν το πρόγραμμα χρήσης δε μηδενίσει το χρονιστή επιτήρησης αυτό πιθανότατα σημαίνει ότι το πρόγραμμα έχει αποτύχει σε κάποια προσπάθειά του είτε εξαιτίας πιθανής κατάρρευσης ή γενικότερα κάποιας απρόβλεπτης συμπεριφοράς οπότε είναι προτιμότερο να εκκινήσει διαδικασία επανατοποθέτησης.

- ✓ Ρολόι Πραγματικού Χρόνου: ο σκοπός του είναι η μέτρηση και η διατήρηση της τρέχουσας ώρας της ημέρας, της ημερομηνίας κ.λπ. Μπορεί να χρησιμοποιηθεί για τη σηματοδότηση συγκεκριμένων γεγονότων με γνώμονα την τρέχουσα ώρα.

Μια τυπική μορφή μικροελεγκτή είναι αυτή που παρουσιάζεται στο σχήμα που ακολουθεί. Από τις διάφορες κατηγορίες μικροελεγκτών, εκείνη που γενικά έχει τη μεγαλύτερη απήχηση στη διεθνή αγορά είναι η κατηγορία των 8 bit, τόσο εξαιτίας του χαμηλού τους κόστους, όσο και λόγω της διαθεσιμότητας των διατάξεων αυτών σε μια μεγάλη ποικιλία, από την άποψη των επιδόσεων και των διαθέσιμων ολοκληρωμένων περιφερειακών μονάδων.



**ΣΧΗΜΑ 1.4:** Ένας 8-bit μικροελεγκτής.

## 1.8 ΚΑΤΑΣΚΕΥΑΣΤΕΣ ΜΙΚΡΟΕΛΕΓΚΤΩΝ

Κάποιοι από τους πιο γνωστούς κατασκευαστές μικροελεγκτών είναι οι παρακάτω:

- ARM (δεν κατασκευάζει αλλά παραχωρεί δικαιώματα χρήσης του πυρήνα της)
- Atmel
- Epson
- Freescale Semiconductor (πρώην Motorola)
- Hitachi
- Maxim (μετά την εξαγορά της Dallas)
- Microchip
- NEC
- Toshiba
- Texas Instruments

## 1.9 ΣΥΓΚΡΙΣΗ ΜΙΚΡΟΕΛΕΓΚΤΗ ΜΕ ΜΙΚΡΟΕΠΕΞΕΡΓΑΣΤΗ

Οι μικροελεγκτές και οι μικροεπεξεργαστές αποτελούν ένα σημαντικό κομμάτι της τεχνολογίας. Ωστόσο, οι δύο αυτές συσκευές διαφέρουν σε αρκετά σημεία. Μία σημαντική διαφορά είναι στην λειτουργικότητά τους. Για να λειτουργήσει ένας μικροεπεξεργαστής θα πρέπει να συνδεθεί και με άλλες συσκευές όπως η μνήμη ή μία συσκευή λήψης και αποστολής δεδομένων. Άρα, είναι η καρδιά του συστήματος.

Από την άλλη μεριά ο μικροελεγκτής, είναι σχεδιασμένος έτσι ώστε να περιέχει όλες τις παραπάνω συσκευές και να μην απαιτείται η παρουσία εξωτερικών συσκευών για να λειτουργήσει, αφού όλα τα απαραίτητα περιφερειακά υπάρχουν στο εσωτερικό του. Οπότε, με τον μικροελεγκτή εξοικονομούμε χώρο και χρόνο.

Μία επιπλέον διαφορά βρίσκεται στην υπολογιστική τους ισχύ. Στους σύγχρονους μικροεπεξεργαστές για τα μη ενσωματωμένα συστήματα δίνεται έμφαση σε αυτό. Η ευελιξία ανάπτυξης διαφορετικών εφαρμογών είναι μεγάλη, καθώς η λειτουργικότητα του τελικού συστήματος καθορίζεται από τα εξωτερικά περιφερειακά τα οποία συνδέονται με την κεντρική μονάδα (μικροεπεξεργαστή) η οποία δεν είναι εξειδικευμένη. Αντίθετα, στους μικροεπεξεργαστές για ενσωματωμένα συστήματα (μικροελεγκτές) οι οποίοι έχουν μικρότερες δυνατότητες συνεργασίας με τα εξωτερικά περιφερειακά αυτού του είδους η ευελιξία είναι περιορισμένη όπως επίσης και η υπολογιστική ισχύς. Οι μικροελεγκτές δίνουν έμφαση στον μικρό αριθμό ολοκληρωμένων κυκλωμάτων που απαιτείται για την λειτουργία μίας συσκευής, το χαμηλό κόστος και την εξειδίκευση.

Η βασική αρχιτεκτονική των μικροελεγκτών δεν διαφέρει από αυτή των κοινών μικροεπεξεργαστών, αν και στους πρώτους είναι απαντάται συχνά η αρχιτεκτονική μνήμης



τύπου Harvard, η οποία χρησιμοποιεί διαφορετικές αρτηρίες σύνδεσης της μνήμης προγράμματος και της μνήμης δεδομένων (πχ οι σειρές AVR από την Atmel και PIC από την Microchip). Στους κοινούς μικροεπεξεργαστές συνηθίζεται η ενιαία διάταξη μνήμης τύπου von-Neumann.

Μία ακόμη διαφορά σχετίζεται με τα συνήθη υποσυστήματα, τα οποία πλαισιώνουν έναν μικροεπεξεργαστή και έναν μικροελεγκτή. Στον μικροεπεξεργαστή, το ολοκληρωμένο κύκλωμα που τον αποτελεί περιέχει μόνο την *Λογική και Αριθμητική Μονάδα (ALU)*, στοιχειώδεις καταχωρητές (registers), προσωρινή μνήμη RAM πολύ υψηλής ταχύτητας (cache memory) και, κάποιες φορές, τον ελεγκτή μνήμης (memory controller). Όμως, για τη λειτουργία ενός πλήρους ενσωματωμένου υπολογιστικού συστήματος, απαιτούνται πολλά εξωτερικά υποσυστήματα και περιφερειακά. Τέτοια είναι:

- 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 κλπ) η οποία περιέχει το λογισμικό του συστήματος. Σε κάποια μοντέλα, είναι δυνατό το κλείδωμα αυτής της μνήμης, μετά την εγγραφή της, ώστε να προστατευτεί το περιεχόμενό της από αντιγραφή.
  - Μεγάλη ποσότητα μνήμης RAM.
- Μόνιμη μνήμη αποθήκευσης παραμέτρων λειτουργίας (τύπου EEPROM ή NVRAM) η οποία να μπορεί να γράφεται τον πυρήνα του μικροελεγκτή. Αυτή η μνήμη έχει, έναντι της FLASH, το πλεονέκτημα της δυνατότητας διαγραφής και εγγραφής οποιουδήποτε μεμονωμένου byte.
  - Κύκλωμα αρχικοποίησης (reset).
  - Διαχειριστή αιτήσεων διακοπής (interrupt request controller) από τα περιφερειακά.
  - Κύκλωμα επιτήρησης τροφοδοσίας (brown-out detection) το οποίο παρακολουθεί την τροφοδοσία και αρχικοποιεί ολόκληρο το σύστημα όταν αυτή πέσει κάτω από τα ανεκτά όρια, προλαμβάνοντας έτσι την αλλοίωση των δεδομένων.
    - Κύκλωμα επιτήρησης λειτουργίας (watchdog timer) το οποίο αρχικοποιεί το σύστημα, αν αυτό εμφανίσει σημάδια δυσλειτουργίας λόγω κολληματος (hang).
    - Τοπικό ταλαντωτή για την παροχή παλμών χρονισμού (clock).
    - Έναν ή περισσότερους χρονιστές-απαριθμητές υψηλής ταχύτητας (hardware timer-counter) για τη δημιουργία καθυστερήσεων, μέτρηση διάρκειας γεγονότων, απαρίθμηση γεγονότων και άλλων λειτουργιών ακριβούς χρονισμού.
      - Ρολόι πραγματικού χρόνου (Real Time Clock, RTC) το οποίο τροφοδοτείται από ανεξάρτητη μπαταρία και γι αυτό πρέπει να έχει πολύ χαμηλή κατανάλωση ρεύματος.
      - Σειρά ανεξάρτητων ψηφιακών εισόδων και εξόδων (Parallel Input-Output , PIO).

Γενικά, όλες οι οικογένειες μικροελεγκτών ενσωματώνουν τα περισσότερα από τα παραπάνω περιφερειακά και έτσι δεν απαιτούνται εξωτερικά υποσυστήματα. Οι διαφοροποιήσεις εντοπίζονται 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, οι οποίοι μπορούν 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 (από το ειδικό τζαμάκι).

## 1.10 ΠΕΡΙΓΡΑΦΗ ΤΩΝ ΜΙΚΡΟΕΛΕΓΚΤΩΝ AVR

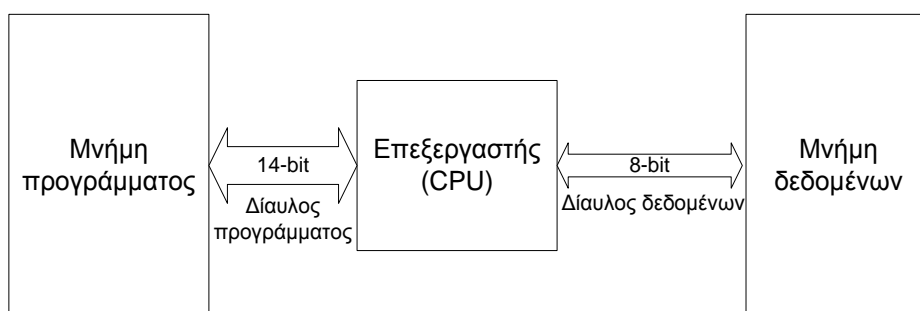
### 1.10.1 ΓΕΝΙΚΑ ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ ΜΙΚΡΟΕΛΕΓΚΤΩΝ

Κάθε μικροϋπολογιστικό σύστημα διακρίνεται για τα ιδιαίτερα χαρακτηριστικά και τις ιδιότητες που έχει, για αυτό το λόγο η αρχιτεκτονική του κάθε μικροελεγκτή αλλάζει. Παρακάτω, θα αναφέρουμε κάποιες κατηγορίες αρχιτεκτονικών σχημάτων:

- Λαμβάνει υπόψη όλο τον αριθμό των εντολών. Άρα έχουμε την αρχιτεκτονική CISC (Complex Instruction Set Computer) ή (Αρχιτεκτονική Σύνθετου Ρεπερτορίου Εντολών), την αρχιτεκτονική RISC (Reduced Instruction Set Computer) ή (Αρχιτεκτονική Μειωμένου Ρεπερτορίου Εντολών) και την MISC (Minimum Instruction Set Computer) ή (Αρχιτεκτονική Ελάχιστου Ρεπερτορίου Εντολών). Από την άλλη πλευρά, όλοι αυτοί οι όροι παρατηρούμε πως έχουν υποστεί πολλές παραφράσεις καθώς και πολλές κοινές ιδιότητες μεταξύ τους.
- Αφορά τον τρόπο με τον οποίο γίνεται η πρόσβαση στη μνήμη προγράμματος και τη μνήμη δεδομένων. Ένα τέτοιο μοναδικό μοντέλο μνήμης, το οποίο είναι γνωστό ως Αρχιτεκτονική Princeton ή Von Neumann έρχεται σε αντίθεση με την Αρχιτεκτονική Harvard, η οποία έχει ξεχωριστό χώρο μνήμης για την αποθήκευση προγράμματος και δεδομένων αντίστοιχα.
- Με βάση τον τρόπο αποθήκευσης και διαχείρισης των δεδομένων εντός της CPU. Ο κύριος σκοπός ενός μικροελεγκτή είναι η διαχείριση δεδομένων. Αυτή η εργασία επιτυγχάνεται με τη βοήθεια του προγράμματος χρήσης. Ο τρόπος διαχείρισης και αποθήκευσης των δεδομένων μέσα στην CPU, διαμορφώνει μία βάση με την οποία ταξινομούνται οι διάφορες αρχιτεκτονικές. Σύμφωνα με τα προηγούμενα, διακρίνουμε 4 κύρια μοντέλα:
  - 1) Το μοντέλο Στοίβας (STACK),
  - 2) Το μοντέλο Συσσωρευτή (ACCUMULATOR),
  - 3) Το μοντέλο Καταχωρητής – Μνήμη (Register-Memory)
  - 4) Το μοντέλο πολλών Καταχωρητών (Register) ή μοντέλο Φόρτωσης – Αποθήκευσης (Load -Store).

### 1.10.2 ΜΙΚΡΟΕΛΕΓΚΤΕΣ AVR

Ο μικροελεγκτής AVR αποτελείται από ένα πυρήνα (ΚΜΕ) αλλά παράλληλα περιλαμβάνει και έναν επεξεργαστή RISC (Reduced Instruction Set Computer). Η κεντρική μονάδα επεξεργασίας συνεργάζεται την ίδια στιγμή με τη μνήμη προγράμματος (program memory) και με την μνήμη δεδομένων (data memory). Αυτό το σημαντικό γεγονός κάνει την αρχιτεκτονική Harvard πολύ αποδοτική, αφού μπορεί να εκτελείται κάποια εντολή ενώ παράλληλα να διαβάζεται ή να εγγράφεται η μνήμη. Με αυτό τον τρόπο πετυχαίνουμε την εκτέλεση περισσότερων εντολών σε μόνο ένα κύκλο μηχανής. Στο παρακάτω σχήμα που ακολουθεί, μπορούμε να διακρίνουμε τον διαφορετικό διάδρομο για την μεταφορά των εντολών (instruction bus) και αντίστοιχα για την μεταφορά των δεδομένων (data bus).



**ΣΧΗΜΑ 1.5:** Σύνδεση της κεντρικής μονάδας επεξεργασίας με τις μνήμες προγράμματος και δεδομένων.

Οι μικροελεγκτές AVR διαθέτουν τα παρακάτω γενικά χαρακτηριστικά:

- ❖ 32 καταχωρητές εργασίας των 8-bit. Άρα, υπάρχουν πολλές θέσεις για να μπορούν να αποθηκεύονται οι διάφορες μεταβλητές μέσα στην κεντρική μονάδα επεξεργασίας.
- ❖ Ενσωματωμένη μνήμη δεδομένων (data memory) τύπου EEPROM και SRAM. Αυτές οι μνήμες χρησιμοποιούνται για την αποθήκευση σταθερών τιμών και μεταβλητών.
- ❖ Ενσωματωμένη μνήμη flash με δυνατότητα προγραμματισμού εντός του συστήματος ISP (In System Programmable) ως μνήμη προγράμματος. Αυτή η ενσωματωμένη μνήμη, δεν κάνει απαραίτητες τις εξωτερικές μνήμες τύπου EPROM ή ROM που περιέχουν τον κώδικα του προγράμματος.
- ❖ Ενσωματωμένο προγραμματιζόμενο χρονιστή με μονάδα διαίρεσης συχνότητας (prescaler). Αυτή η μονάδα χρησιμοποιείται σε περιπτώσεις που η μονάδα χρονισμού είναι απαραίτητη.
- ❖ Προγραμματιζόμενο χρονιστή επιτήρησης (Watch Dog Timer), όπου ο ρόλος του είναι η αποφυγή λάθους λειτουργιών σε περιπτώσεις κατάρρευσης του συστήματος.
- ❖ Δυνατότητα εξωτερικών και εσωτερικών διακοπών.
- ❖ Λειτουργίες ηρεμίας και αποκοπής. Ο επεξεργαστής μεταβαίνει σε αυτές τις καταστάσεις όταν δεν υπάρχει κάποια κινητικότητα, έτσι ώστε παράλληλα να έχουμε και εξοικονόμηση ενέργειας.
- ❖ Εσωτερικό κύκλωμα, που επανατοποθετεί το σύστημα κατά την εφαρμογή της τάσης τροφοδοσίας.

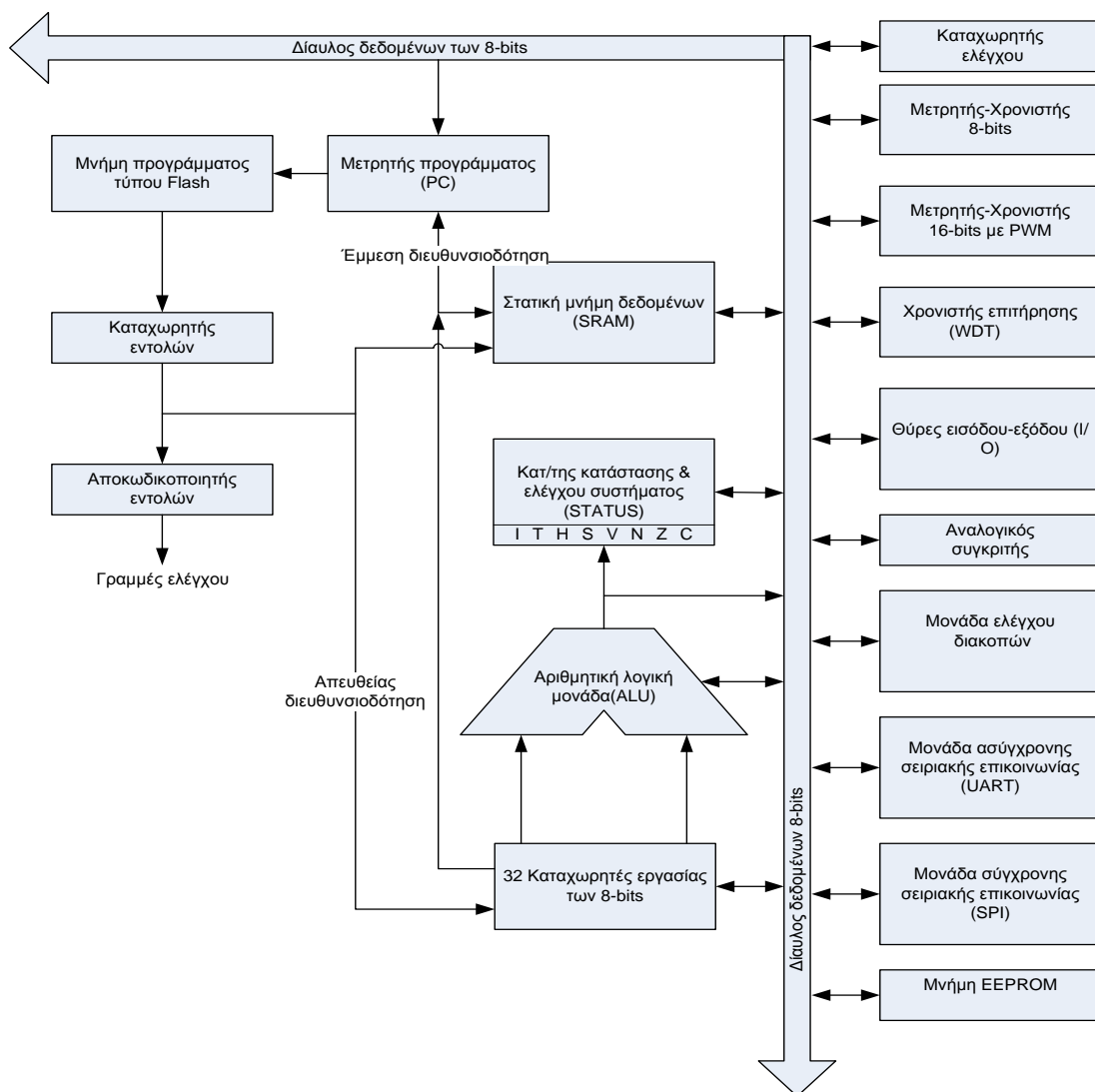
- ❖ Λειτουργεί σε συχνότητες χρονισμού από 0 έως 10MHz. Ενώ οι πιο πολλές εντολές τερματίζονται σε μία περίοδο του κεντρικού σήματος χρονισμού, η επίδοση είναι πιο υψηλή σε σύγκριση με τους κλασσικούς μικροελεγκτές που λειτουργούν στην ίδια συχνότητα χρονισμού αλλά χρειάζονται περισσότερους κύκλους για την εκτέλεση μίας εντολής.
- ❖ Κάποια μοντέλα διαθέτουν ενσωματωμένο ταλαντωτή τύπου RC για την επιλογή της συχνότητας χωρίς τη χρήση εξωτερικού κρυστάλλου.
- ❖ Τέλος, έχουν έως 4 παράλληλες θύρες των 8-bit και πολλά περιφερειακά όπως UART,SPI,μετατροπέας ADC,αναλογικός συγκριτής και άλλα. Έχει έως και 64 καταχωρητές για να υποστηρίξει τις θύρες εισόδου-εξόδου.

Παρακάτω ακολουθούν διάφοροι τύποι μικροελεγκτών και τα κύρια χαρακτηριστικά τους:

**ΠΙΝΑΚΑΣ 1.1:** Διάφοροι τύποι μικροελεγκτών AVR.

Μοντέλο	Ακροδέκτες	Flash	EEPROM	RAM	UART	ADC
90S1200	20	1K	64 bytes	0	Όχι	Όχι
90S2313	20	2K	128	128	Ναι	Όχι
90S2323	8	2K	128	128	Όχι	Όχι
90S2333	28	2K	128	128	Ναι	Ναι
90S4433	28	4K	256	128	Ναι	Ναι
90S4414	40	4K	256	256	Ναι	Όχι
90S8515	40	8K	512	512	Ναι	Όχι
90S4434	40	4K	256	256	Ναι	Ναι
90S2343	8	2K	128	128	Όχι	Όχι
Mega103	64	128K	4096	4096	Ναι	Ναι
Mega603	64	64K	2048	4096	Ναι	Ναι
Tiny10	8	1K	64	0	Όχι	Όχι
Tiny12	8	1K	64	0	Όχι	Όχι
Tiny13	8	2K	128	128	Όχι	Όχι

Επιπλέον, στο ακόλουθο σχήμα μπορεί να δει κανείς την αρχιτεκτονική διάταξη των μικροεπεξεργαστών AVR και των λειτουργικών μονάδων τους. Το σημαντικό σε όλους τους τύπους των μικροελεγκτών είναι ότι ο πυρήνας της αρχιτεκτονικής τους είναι ο ίδιος και έχουν το ίδιο σύνολο εντολών, όμως διαφέρουν ως προς τα περιφερειακά και το μέγεθος της μνήμης που έχουν. Όπως είναι σύνηθες σε αυτού του τύπου τους ελεγκτές, υπάρχουν διαφορετικοί εσωτερικοί δίαυλοι για τη μνήμη προγράμματος και τη μνήμη δεδομένων.



**ΣΧΗΜΑ 1.6:** Αρχιτεκτονική διάταξη περιφερειακών μονάδων και μικροεπεξεργαστών AVR.

Πιο συγκεκριμένα, θα αναλύσουμε και τα χαρακτηριστικά των μικροελεγκτών AVR:

- Συνδυάζουν την αρχιτεκτονική RISC με ως επί τω πλείστον σταθερού μήκους εντολές, διαδικασίες αποθήκευσης – φόρτωσης στη μνήμη και 32 καταχωρητές γενικής χρήσης.
- Διαθέτουν μηχανισμό συνεχούς διοχέτευσης εντολών (Instruction pipeline) σε δύο στάδια, που επιταχύνει σημαντικά τη διαδικασία εκτέλεσης.
- Οι περισσότερες από τις εντολές που περιλαμβάνει το ρεπερτόριό τους εκτελούνται στη διάρκεια μιας περιόδου του κεντρικού ρολογιού.
- Λειτουργούν σε συχνότητες χρονισμού έως 10 MHz.
- Διαθέτουν μεγάλη ποικιλία σε ότι αφορά ενσωματωμένες περιφερειακές μονάδες όπως, ψηφιακές εισόδους – εξόδους (I/O), μετατροπείς αναλογικού σήματος σε ψηφιακό ή ADC, μνήμη τύπου EEPROM, χρονιστές, μονάδες ασύγχρονης σειριακής επικοινωνίας ή UART (Universal Asynchronous Receiver Transmitter), ρολόγια πραγματικού χρόνου (RTC – Real Time Clock), μονάδες διαμόρφωσης εύρους παλμών (PWM – Pulse Width Modulation) και άλλα.
- Ενσωματωμένες μνήμες προγράμματος και δεδομένων.
- Δυνατότητα προγραμματισμού εντός του συστήματος (ISP – In-System Programmable).
- Διατίθενται σε συσκευασίες των 8 έως 64 ακροδεκτών οπότε κρίνονται κατάλληλοι για έναν μεγάλο αριθμό διαφορετικών εφαρμογών.
- Είναι περίπου 12 φορές ταχύτεροι και πιο αποδοτικοί σε σχέση με τους ελεγκτές κλασικής αρχιτεκτονικής CISC (Complex Instruction Set Computer).
- Ευρεία περιοχή τάσεων λειτουργίας από 2.7 V έως 6.0V.
- Η σχετικά απλή αρχιτεκτονική τους δίνει το πλεονέκτημα του μικρού σε απαίτηση κύκλου εκμάθησης στους αρχάριους.

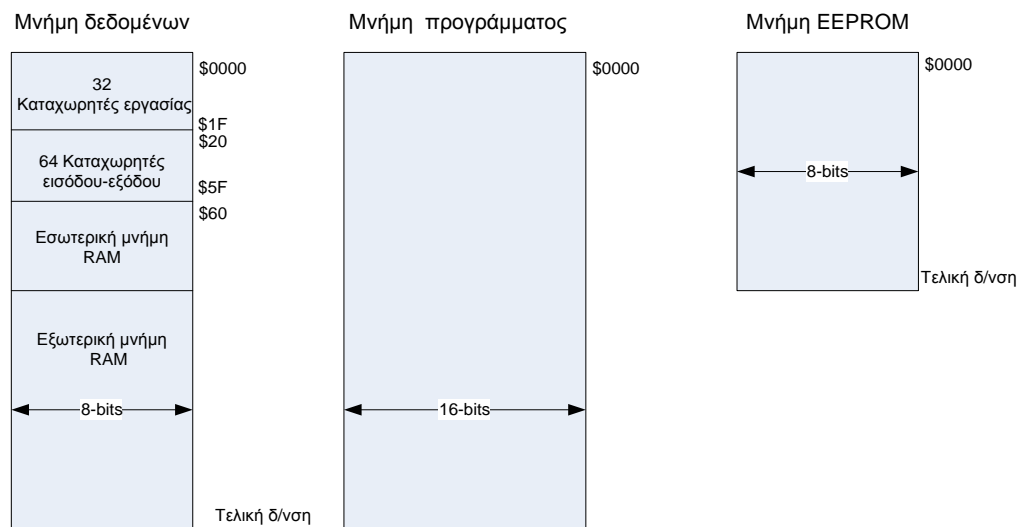
### 1.10.3 ΜΝΗΜΕΣ ΠΡΟΓΡΑΜΜΑΤΟΣ ΚΑΙ ΔΕΔΟΜΕΝΩΝ

Ένας μικροελεγκτής έχει τα πιο κάτω είδη μνήμης:

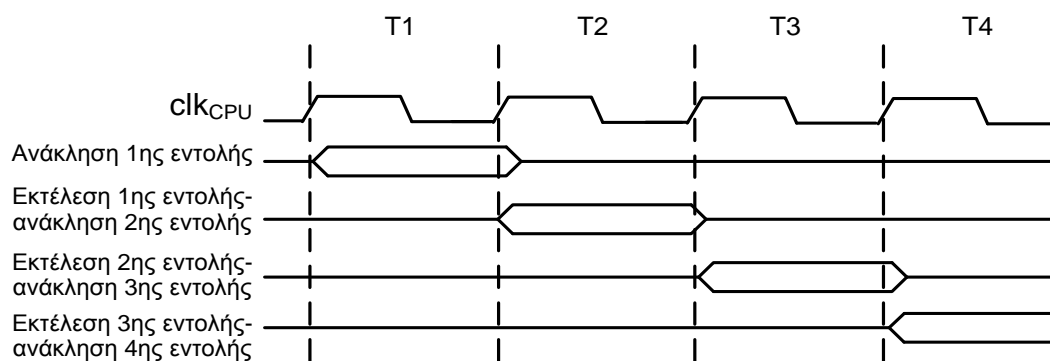
- ✓ μνήμη δεδομένων
- ✓ μνήμη προγράμματος τύπου flash
- ✓ μνήμη EEPROM με αρχική διεύθυνση \$0000 και χωρητικότητα από 64bytes ως 4Kbytes.

Η μνήμη προγράμματος είναι τύπου flash, πετυχαίνει ταχεία φόρτωση και αποθήκευση δεδομένων, και συνδέεται με τον δίαυλο των 16 bit για την τροφοδοσία του καταχωρητή εντολών. Ωστόσο, η ακριβής χωρητικότητα της μνήμης αυτής διαφέρει από μικροελεγκτή σε μικροελεγκτή. Στην μνήμη προγράμματος αποθηκεύονται οι εντολές και τα διανύσματα διακοπών (interrupt vectors). Όλες οι εντολές έχουν μήκος 16 bits ή 32 bits (δηλαδή 2 bytes=1 λέξη), άρα καταλαμβάνουν μία ή δύο θέσεις της μνήμης προγράμματος, αφού η μνήμη flash, είναι οργανωμένη σε διάταξη 8K x 16. Η πρόσβαση στη μνήμη προγράμματος γίνεται σε μια περίοδο του σήματος χρονισμού. Επειδή συντελείται μία συνεχής διοχέτευση εντολών (instruction pipeline), η εκτέλεση των περισσότερων εντολών γίνεται σε μια περίοδο του κεντρικού ρολογιού. Αυτό σημαίνει ότι κατά την εκτέλεση μιας εντολής γίνεται ταυτόχρονα ανάκληση από τη μνήμη της επόμενης εντολής μέσω μιας παράλληλης διαδικασίας, όπως

είναι προφανές στο ακόλουθο σχήμα. Σε αυτό απεικονίζεται η παράλληλη διαδικασία ανάκλησης (από τη μνήμη προγράμματος) και εκτέλεσης εντολών (με τη χρήση της μνήμης δεδομένων), η οποία καθίσταται δυνατή χάρη στην αρχιτεκτονική Harvard και το ταχείας πρόσβασης Register File.



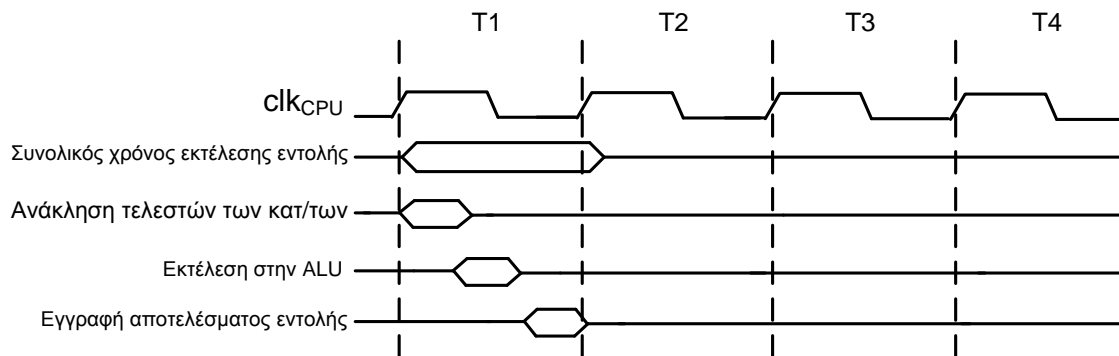
**ΣΧΗΜΑ 1.7:** Τύποι από τις μνήμες των μικροελεγκτών AVR.



**ΣΧΗΜΑ 1.8:** Ταυτόχρονη εκτέλεση και ανάκληση των εντολών.

Αφού η εντολή ανακαλείται από την μνήμη φορτώνεται στον καταχωρητή εντολών, ο οποίος με τη σειρά του συνδέεται με την ομάδα των καταχωρητών εργασίας και επιλέγει τους κατάλληλους καταχωρητές προκειμένου να χρησιμοποιηθούν από την αριθμητική μονάδα ALU για την εκτέλεση της συγκεκριμένης εντολής. Επιπρόσθετα, η έξοδος του καταχωρητή εντολών αποκωδικοποιείται μέσα από τη μονάδα του αποκωδικοποιητή εντολών, όπου προκύπτουν τα απαραίτητα σήματα ελέγχου για την ολοκλήρωση της εκτέλεσης της τρέχουσας εντολής. Η διάρκεια της μνήμης είναι τουλάχιστον 1000 κύκλοι εγγραφής / διαγραφής. Προκειμένου να προστατεύεται το λογισμικό, η μνήμη χωρίζεται στο τμήμα

εκκίνησης και το τμήμα εφαρμογών, τα οποία διαθέτουν bits κλειδώματος για ασφάλεια εγγραφής και ανάγνωσης / εγγραφής.



**ΣΧΗΜΑ 1.9:** Διαδικασία εκτέλεσης εντολής στην Αριθμητική και Λογική Μονάδα.

Η μνήμη δεδομένων του συστήματος διαιρείται σε πολλούς διαφορετικούς τύπους και συνδέεται με έναν διάδρομο των 8 bit για την επικοινωνία των περιφερειακών μονάδων με τους καταχωρητές ελέγχου. Χωρίζεται στα ακόλουθα πέντε τμήματα:

1. Ένα τμήμα που αντιστοιχεί στην ομάδα καταχωρητών εργασίας (register file) των 8 bits. Η ομάδα αυτή των καταχωρητών υπάρχει σε όλους τους μικροελεγκτές της οικογένειας των AVR.
2. Ένα τμήμα 64 καταχωρητών εισόδου – εξόδου των 8 bits. Δε διαθέτουν όλοι οι μικροελεγκτές της σειράς AVR και τους 64 παραπάνω καταχωρητές. Ανάλογα με το πλήθος των εσωτερικών περιφερειακών μονάδων κάποιοι διαθέτουν περισσότερους.
3. Ένα τμήμα εσωτερικής στατικής μνήμης SRAM.
4. Ένα τμήμα εξωτερικής στατικής μνήμης SRAM.
5. Ένα τμήμα εσωτερικής μνήμης τύπου EEPROM. Ο τύπος αυτός εσωτερικής μνήμης είναι διαθέσιμος σχεδόν σε όλους τους μικροελεγκτές AVR και η πρόσβασή του επιτυγχάνεται σε ιδιαίτερο χώρο διευθύνσεων μνήμης. Η μνήμη αυτή μπορεί να διαβαστεί και να γραφεί από οποιοδήποτε πρόγραμμα.

#### 1.10.4 ΜΝΗΜΗ SRAM

Η μνήμη SRAM περιέχεται στα περισσότερα μοντέλα μικροελεγκτών και είναι μία μνήμη η οποία δεν είναι απευθείας προσβάσιμη από την κεντρική μονάδα επεξεργασίας σε αντίθεση με τους καταχωρητές. Επίσης, έχει την δυνατότητα να διατηρεί τα στοιχεία που φιλοξενεί, χωρίς καμία αλλοίωση για όση ώρα τροφοδοτείται με ρεύμα. Σε αυτές τις μνήμες χρησιμοποιούνται ειδικοί διακόπτες, που μπορεί να είναι είτε ανοικτοί είτε κλειστοί. Ο τρόπος που είναι κατασκευασμένη η μνήμη SRAM, μοιάζει με αυτόν που έχουν και οι



επεξεργαστές δηλαδή πολλά ολοκληρωμένα κυκλώματα, που έχουν τοποθετηθεί σε μία πλακέτα πυριτίου. Για να αποθηκεύσουμε το καθένα bit σε αυτή την μνήμη, πρέπει να έχουμε από 4 μέχρι 6 τρανζίστορ. Το μέγεθος της SRAM είναι μεγαλύτερο από αυτό της DRAM, όπου στην δεύτερη το κάθε bit χρειάζεται μόνο ένα πυκνωτή.

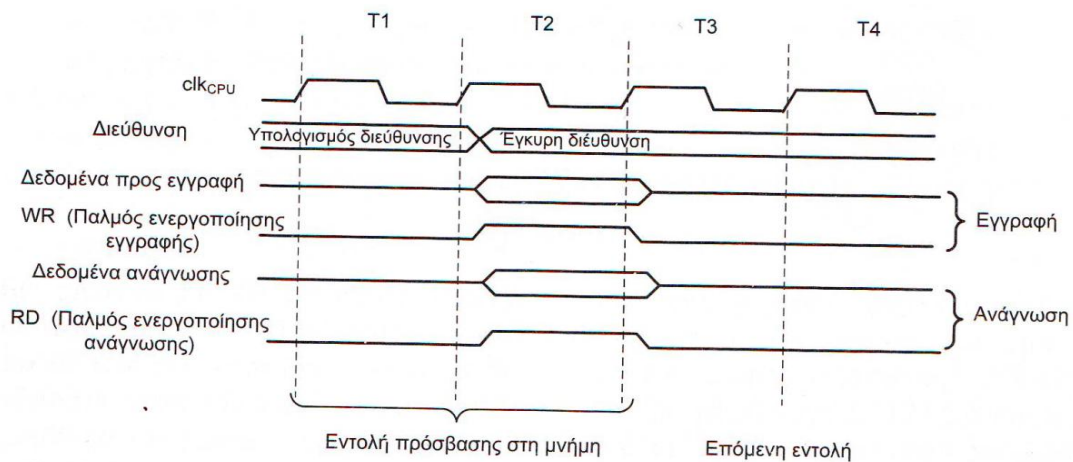
Η Δυναμική RAM είναι ένας τύπος RAM, που μπορεί να διατηρήσει τα περιεχόμενά της αναλλοίωτα, μόνο αν γίνεται συνεχής αναζωογόνηση από ένα ειδικό κύκλωμα που ονομάζεται refresh circuit. Αυτό το κύκλωμα διαβάζει με μεγάλη συχνότητα τα περιεχόμενα που έχει ο κάθε πυκνωτής. Αν δεν υπήρχε αυτό το κύκλωμα αναζωογόνησης, τότε τα περιεχόμενα της μνήμης δεν θα σώζονταν ακόμα και αν τροφοδοτούσαμε τον ηλεκτρονικό υπολογιστή με ρεύμα. Η DRAM ονομάζεται δυναμική RAM εξαιτίας του κυκλώματος αναζωογόνησης.

#### **1.10.4.1 ΕΞΩΤΕΡΙΚΗ ΜΝΗΜΗ SRAM**

Η εξωτερική μνήμη χρησιμοποιείται μόνο στους μεγαλύτερους επεξεργαστές της σειράς AVR όπως επίσης και σε εφαρμογές υψηλών απαιτήσεων. Ο χρόνος πρόσβασης στην εξωτερική μνήμη έχει διάρκεια 3 περιόδους χρονισμού, όμως σε κάποιες περιπτώσεις μπορεί να φτάσει και τις 4. Οι επεξεργαστές που έχουν θύρες πρόσβασης σε εξωτερικό δίαυλο δεδομένων και διευθύνσεων, μπορεί να χρησιμοποιηθεί οποιαδήποτε ποσότητα εξωτερικής μνήμης SRAM.

#### **1.10.4.2 ΕΣΩΤΕΡΙΚΗ ΜΝΗΜΗ SRAM**

Σε περιπτώσεις κατά τις οποίες τα δεδομένα μας δεν μπορούν να αποθηκευτούν στους καταχωρητές τότε η χρήση αυτής της μνήμης είναι αναγκαία. Είναι μία μνήμη η οποία δεν είναι απευθείας προσβάσιμη από την κεντρική μονάδα επεξεργασίας σε αντίθεση με τους καταχωρητές. Το μέγεθός της είναι από 128 bytes έως 4096 bytes. Για την πρόσβαση σε θέσεις μνήμης χρησιμοποιούμε έναν καταχωρητή για προσωρινή αποθήκευση. Επί προσθέτως, η συγκεκριμένη μνήμη χρησιμοποιείται τόσο για την αποθήκευση μεταβλητών όσο και ως στοίβα του συστήματος επειδή εάν εμφανιστεί μία διακοπή θα αποθηκευτούν εκεί οι πληροφορίες μας. Επίσης, θα έχουμε την δυνατότητα να επαναφέρουμε αυτές τις πληροφορίες. Από την άλλη πλευρά παρατηρούμε πως αυτή η μνήμη είναι πιο αργή σε σχέση με τους καταχωρητές. Αυτό το γεγονός οφείλεται στο ότι ο χρόνος πρόσβασης στη μνήμη απαιτεί δύο περιόδους του σήματος χρονισμού. Στην πρώτη περίοδο εκτελούνται οι απαιτούμενες λειτουργίες των καταχωρητών για τον καθορισμό της διεύθυνσης. Κατά την διάρκεια της δεύτερης περιόδου έχουμε την πρόσβαση στη μνήμη (ανάγνωση ή εγγραφή). Στο ακόλουθο σχήμα απεικονίζονται οι φάσεις ανάγνωσης- εγγραφής:



**ΣΧΗΜΑ 1.10:** Φάσεις ανάγνωσης- εγγραφής στο εσωτερικό της μνήμης SRAM.

Η σημαντικότερη χρήση αυτής της μνήμης είναι η λειτουργία της στοίβας. Εκεί, μπορούμε να αποθηκεύσουμε (push) δεδομένα και έπειτα να τα ανακτήσουμε (pop). Τα δεδομένα μπορεί να είναι το περιεχόμενο ενός καταχωρητή ή ακόμη και διάφορες τιμές.

### 1.10.5 ΚΑΤΑΧΩΡΗΤΕΣ

Οι καταχωρητές χαρακτηρίζονται ως μικρά σημεία της μνήμης που βρίσκονται μέσα στον επεξεργαστή οι οποίοι χρησιμοποιούνται για να φιλοξενούν δεδομένα τα οποία υπόκεινται επεξεργασία εκείνη την στιγμή. Επιπλέον, μπορεί να ασκούν κάποια συγκεκριμένη λειτουργία ή να έχουν και γενική χρήση από τους προγραμματιστές. Κάποιοι από τους καταχωρητές μπορεί να έχουν κάποια συγκεκριμένη λειτουργία. Έτσι, έχουμε:

- ⊙ Συσσωρευτής: Είναι ο πιο σημαντικός καταχωρητής του μικροεπεξεργαστή. Κρατάει τα δεδομένα για διαχείριση και μία από τις δύο λέξεις της λειτουργίας της αριθμητικής λογικής μονάδας. Το αποτέλεσμα της λειτουργίας της ALU τοποθετείται στον συσσωρευτή.
- ⊙ Καταχωρητής εντολών: Οι εντολές ενός προγράμματος βρίσκονται στην κεντρική μνήμη και ο μικροεπεξεργαστής τις ανακαλεί μία-μία από την μνήμη στον καταχωρητή εντολών για να τις εκτελέσει. Ο καταχωρητής εντολών είναι ένας ειδικός καταχωρητής ο οποίος περιέχει πάντα τον κωδικό της εντολής που πρόκειται να εκτελεστεί.
- ⊙ Απαριθμητής προγράμματος: στόχος αυτού του μετρητή είναι να παρακολουθεί ποια εντολή εκτελείται και ποια θα εκτελεστεί αργότερα. Κάθε φορά που ο μικροεπεξεργαστής φέρνει μία εντολή, το περιεχόμενο του απαριθμητή

προγράμματος αυξάνεται κατά ένα έτσι ώστε ο καταχωρητής αυτός να περιέχει την διεύθυνση της επόμενης εντολής που θα εκτελεστεί.

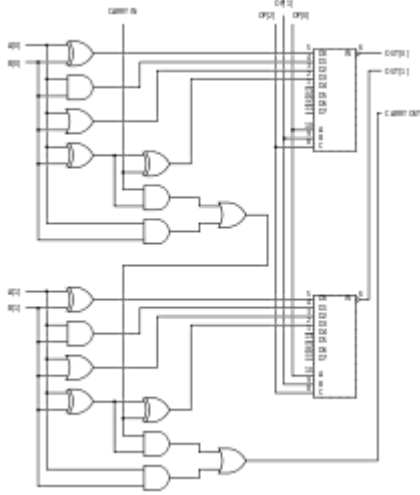
Οι καταχωρητές είναι η πιο γρήγορη μνήμη και σε αυτούς γίνονται οι περισσότερες λειτουργίες. Όσο πιο πολλούς έχει ο επεξεργαστής τόσο μεγαλύτερη ευκολία έχουμε για να γράψουμε καλύτερο κώδικα. Οι περισσότεροι μικροελεγκτές της οικογένειας AVR, διαθέτουν 32 καταχωρητές γενικής χρήσης που συμβολίζονται με τα σύμβολα R0 έως R31.

#### **1.10.6 ΑΡΙΘΜΗΤΙΚΗ ΚΑΙ ΛΟΓΙΚΗ ΜΟΝΑΔΑ**

Η αριθμητική και λογική μονάδα είναι ένα θεμελιώδες στοιχείο της κεντρικής μονάδας ενός ηλεκτρονικού υπολογιστή. Είναι ένα ψηφιακό κύκλωμα το οποίο πραγματοποιεί αριθμητικούς και λογικούς υπολογισμούς. Η αποθήκευση των αποτελεσμάτων γίνεται σε έναν συγκεκριμένο καταχωρητή που υποδεικνύεται από την ίδια την εντολή. Επεξεργάζεται τα δεδομένα που της παρέχουν τα άλλα στοιχεία του συστήματος του υπολογιστή, δηλαδή η μονάδα ελέγχου, η μνήμη, οι καταχωρητές, η λειτουργία I/O πριν βγάλουν τα αποτελέσματα. Μία αριθμητική και λογική μονάδα φορτώνει τα δεδομένα από τους καταχωρητές εισόδου μία εξωτερική μονάδα λείει στην ALU ποιες πράξεις να κάνει στα δεδομένα και στο τέλος ειδικοί μηχανισμοί μετακινούν τα δεδομένα από τους καταχωρητές στη μνήμη.

Οι περισσότερες αριθμητικές και λογικές μονάδες μπορούν να εκτελέσουν τις παρακάτω πράξεις:

- ✓ Ακέραιες αριθμητικές ( πρόσθεση, αφαίρεση, μερικές φορές πολλαπλασιασμό και διαίρεση, αν και η υλοποίησή τους είναι ακριβή).
- ✓ Λογικές (και, ή, όχι)
- ✓ Μετατόπιση περιστροφής μίας λέξης τόσο όσο ενός προκαθορισμένου αριθμού bits προς τα δεξιά ή τα αριστερά (μπορεί να υλοποιηθεί σαν πολλαπλασιασμός ή διαίρεση με το 2).



**ΣΧΗΜΑ 1.11:** Ένα παράδειγμα αριθμητικής και λογικής μονάδας των 2 bit που εκτελεί τις λογικές πράξεις AND, OR, XOR.

## **ΚΕΦΑΛΑΙΟ 2: ΣΧΕΤΙΚΑ ΜΕ ΤΟ ARDUINO**

## 2.1 ΓΕΝΙΚΕΣ ΠΛΗΡΟΦΟΡΙΕΣ

Το 2003 στην πόλη Ίβρεα της Ιταλίας ένας φοιτητής ανέπτυξε το wiring project. Αυτό το project ήταν ένα απλό σύστημα το οποίο βασίζεται σε ένα μικροελεγκτή που είναι εύκολος για ανάπτυξη εφαρμογών. Το Arduino ήταν ένα εργαλείο, έτσι ώστε να χρησιμοποιηθεί από άτομα τα οποία δεν είχαν κάποια σχέση με την τεχνολογία. Επιπλέον, είναι μία πλατφόρμα ανοιχτού κώδικα. Αυτή η πλατφόρμα περιέχει μία απλή πλακέτα μικροελεγκτή καθώς επίσης και ένα περιβάλλον ανάπτυξης προγραμμάτων για τον προγραμματισμό του μικροελεγκτή. Χάρη στο εύκολο λογισμικό του, είναι ο "εγκέφαλος" χιλιάδων εργασιών, από μία απλή καθημερινή εφαρμογή έως ένα περίπλοκο σύστημα. Το συμπέρασμα αυτό μας το δίνει η ίδια η πλατφόρμα αφού χρησιμοποιεί τους μικροελεγκτές Atmel AVR οι οποίοι έχουν διάφορες εκδόσεις διαθέτοντας περισσότερα ή λιγότερα χαρακτηριστικά. Η γλώσσα προγραμματισμού που διαθέτει έτσι ώστε να προγραμματίσουμε τον μικροελεγκτή είναι η Wiring C η οποία είναι μία παραλλαγή της γλώσσας C++ όπου με την προσθήκη βιβλιοθηκών εμφανίζονται περισσότερες δυνατότητες.

## 2.2 ΠΛΕΟΝΕΚΤΗΜΑΤΑ ΤΟΥ ARDUINO

Εξαιτίας της εύκολης και προσβάσιμης εμπειρίας της χρήσης του, το Arduino χρησιμοποιείται για ποικίλες εφαρμογές. Υπάρχουν πολλοί μικροεπεξεργαστές και μικροελεγκτές που παρέχουν στο χρήστη ακριβώς αυτά που αναζητά. Σε σχέση με τις άλλες πλατφόρμες διαφέρει σε πολλούς τομείς. Αρχικά, παρατηρούμε πως με το Arduino η ανάπτυξη της εφαρμογής μας γίνεται απλή και από την άλλη πλευρά ανακαλύπτουμε πως έχει και άλλα πολλά πλεονεκτήματα τα οποία αναφέρονται παρακάτω:

**Λογισμικό ανοιχτού κώδικα:** Όλα τα προγράμματα του Arduino είπαμε και πιο πάνω ότι είναι ανοιχτού κώδικα. Πρακτικά, αυτό σημαίνει πως ο οποιοσδήποτε σχετικός και μη σχετικός με την τεχνολογία μπορεί να βρει εύκολα κάποια τμήματα του κώδικα, να τα μελετήσει και να τα προσαρμόσει στις ανάγκες της εκάστοτε εφαρμογής που θέλει να υλοποιήσει. Επί προσθέτως, μέσα από την προσθήκη βιβλιοθηκών C++ μπορούμε να προσθέσουμε και άλλες δυνατότητες στο περιβάλλον ανάπτυξης. Οι προχωρημένοι χρήστες μπορούν να γράψουν τον δικό τους κώδικα και να είναι διαθέσιμος μέσα από τις βιβλιοθήκες ενώ από την άλλη πλευρά οι αρχάριοι χρήστες μπορούν να προσθέσουν επεκτάσεις χωρίς να γνωρίζουν τις λεπτομέρειες του προγραμματισμού που κρύβεται από πίσω.

**Υλικό ανοιχτού κώδικα:** Η πλατφόρμα Arduino βασίζεται στον μικροελεγκτή της Atmel και είναι ο ATmega328. Όποιος ασχολείται με αυτή την πλατφόρμα έχει την δυνατότητα να εξελίξει αυτή την πλακέτα και να συνδέει πάνω της διάφορα εξαρτήματα που του είναι χρήσιμα για την εφαρμογή του.

**Συμβατότητα με λειτουργικά συστήματα:** Το λογισμικό που περιέχεται στην αναπτυξιακή πλατφόρμα Arduino μπορεί να εκτελείται σε διάφορα περιβάλλοντα όπως π.χ. Windows, Linux και Macintosh. Όπου την ίδια στιγμή, άλλες αναπτυξιακές πλατφόρμες λειτουργούν μόνο σε περιβάλλον Windows.

**Απλό περιβάλλον προγραμματισμού:** Το περιβάλλον προγραμματισμού του Arduino είναι εύκολο και ευρύ αφού μπορεί να χρησιμοποιηθεί από έμπειρους και άπειρους χρήστες.

**Φθηνό κόστος:** Αν συγκρίνουμε άλλες αναπτυξιακές πλατφόρμες με αυτή του Arduino θα παρατηρήσουμε ότι η τελευταία είναι η πιο φθηνή λύση.

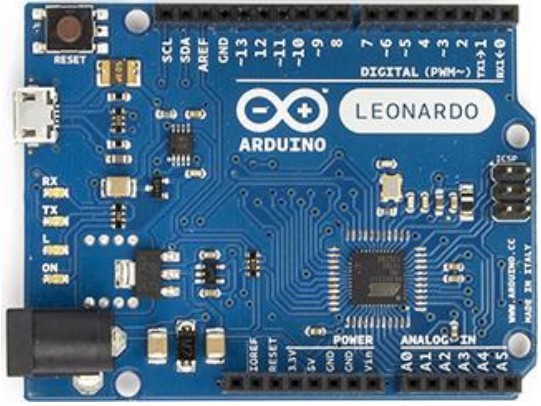
## 2.3 Εκδόσεις αναπτυξιακής πλατφόρμας Arduino

Υπάρχουν πολλές εκδόσεις και παραλλαγές του Arduino. Κάθε έκδοση υποστηρίζει συγκεκριμένο τύπο μικροελεγκτή και διαθέτει ένα σύνολο ακροδεκτών για είσοδο και έξοδο. Αυτοί οι ακροδέκτες χωρίζονται σε ακροδέκτες εισόδου/εξόδου και σε ακροδέκτες που έχουν σχέση με την τροφοδοσία. Η πρώτη κατηγορία είναι η ανταλλαγή σημάτων με εξωτερικά συστήματα και συσκευές. Η δεύτερη κατηγορία είναι οι ακροδέκτες που έχουν σχέση με την τροφοδοσία. Το Arduino διαβάζει και αναλογικά σήματα όπως επίσης και ψηφιακά. Για παράδειγμα, όταν κάποιος αισθητήρας θερμοκρασίας μας δίνει αναλογική ή ψηφιακή έξοδο θα πρέπει να τον συνδέσουμε με το Arduino για να συλλέξουμε τις τιμές αυτές. Στις εξόδους έχουμε μόνο ψηφιακά σήματα.



Επιπλέον, η τροφοδοσία που μπορεί να προσφέρει σε εξωτερικές συσκευές φτάνει μέχρι τα 5V ενώ στις εξόδους το μέγιστο ρεύμα δεν ξεπερνάει τα 30-40 mA. Αυτός ο περιορισμός, μας είναι πολύ σημαντικός ιδιαίτερα στην περίπτωση που θέλουμε να ελέγξουμε ένα κινητήρα επειδή απαιτείται περισσότερο ρεύμα κατά την εκκίνησή του. Από την άλλη πλευρά η σύνδεσή του με θύρα USB μας επιτρέπει την σύνδεσή του με τον ηλεκτρονικό υπολογιστή ενώ ταυτόχρονα παρέχει και την τροφοδοσία των 5V για την λειτουργία του. Αφού το προγραμματίσουμε μέσω του υπολογιστή, το Arduino μπορεί να λειτουργήσει και μόνο του με την προϋπόθεση ότι τροφοδοτείται από κάποια μπαταρία.

Στην συνέχεια, μπορούμε να δούμε κάποια από τα γενικά χαρακτηριστικά των εκδόσεων του Arduino:

Microcontroller	ATmega32u4
Operation Voltage	5v
Input Voltage	7-12V
Digital I/O pins	20
PWM Channels	7
Analog In channel	12
Dc current i/o pin	40mA
Dc current per 3.3v	50mA
Flash memory	32Kb
SRAM	2.5KB Atmega32u4
EEPROM	1KB



**EIKONA 2.1:** Arduino Leonardo

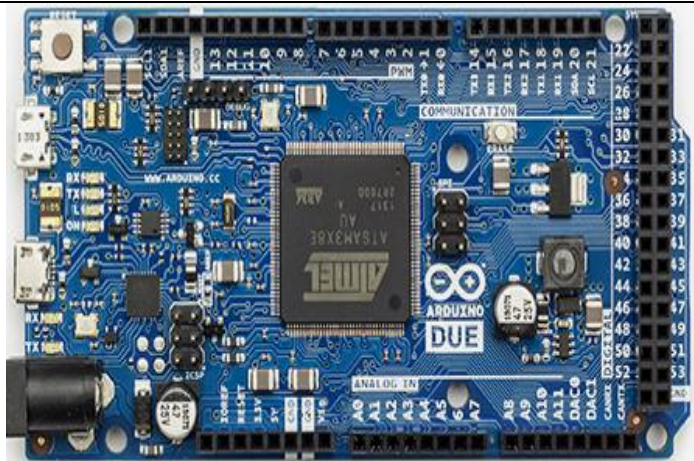
	Atmega32u4	
Clock speed	16MHz	
Microcontroller	ATmega328	
Operation Voltage	5v	
Input Voltage	7-12V	
Digital I/O pins	14	
PWM Channels	6	
Analog In channel	6	
Dc current i/o pin	40mA	
Dc current per 3.3v	50mA	
Flash memory	32Kb	
SRAM	2KB	
EEPROM	1KB	
Clock speed	16MHz	
Microcontroller	ATmega2560	
Operation Voltage	5v	
Input Voltage	7-12V	
Digital I/O pins	54	
PWM Channels	15	
Analog In channel	16	
Dc current i/o pin	40mA	
Dc current per 3.3v	50mA	
Flash memory	256Kb	
SRAM	8KB	
EEPROM	4KB	
Clock speed	16MHz	
Microcontroller	AT91SAM3X8E	
Operation Voltage	3.3V	
Input Voltage	7-12V	
Digital I/O pins	54	
PWM Channels	12	

**EIKONA 2.2:** Arduino Uno Rev3

**EIKONA 2.3:** Arduino Mega 2560 Rev3

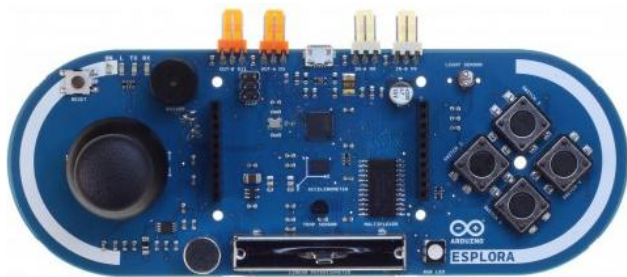


Analog In channel	12
Dc current i/o pin	130mA
Dc current per 3.3v	800mA
Flash memory	512Kb
SRAM	96KB (64+32)
EEPROM	1KB Atmega32u4
Clock speed	84MHz



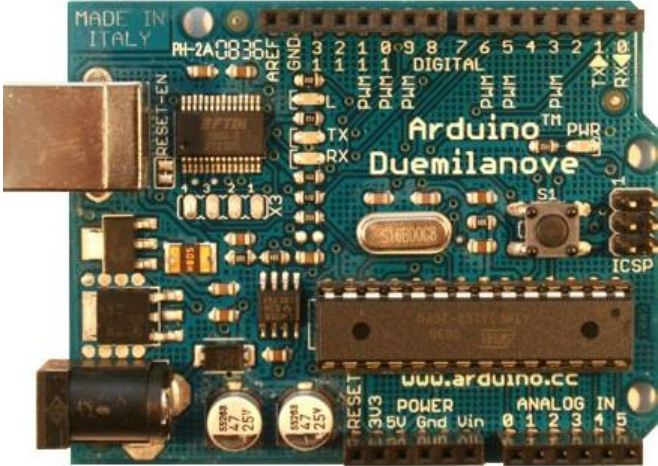

**EIKONA 2.4:** Arduino DUE

Microcontroller	ATmega32u4
Operation Voltage	5V
Input Voltage	
Digital I/O pins	
PWM Channels	
Analog In channel	
Dc current i/o pin	
Dc current per 3.3v	
Flash memory	32Kb
SRAM	2.5KB
EEPROM	1KB Atmega32u4
Clock speed	16MHz



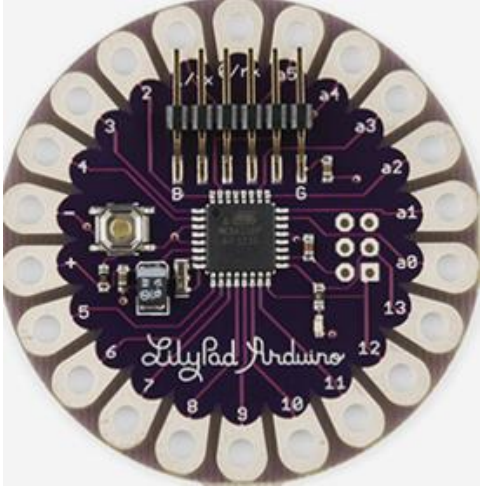
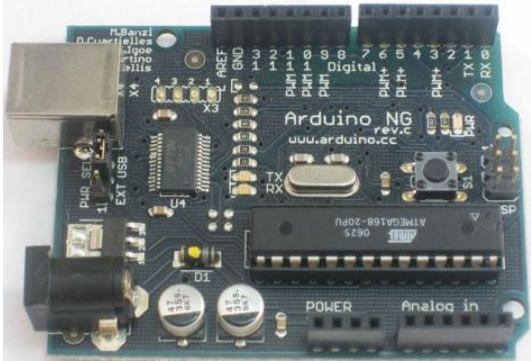
**EIKONA 2.5:** Arduino Esplora

Microcontroller	ATmega168
Operation Voltage	5V
Input Voltage	7-12
Digital I/O pins	14
PWM Channels	6
Analog In channel	6
Dc current i/o pin	40mA
Dc current per 3.3v	50mA

Flash memory	32Kb ή 16Kb	
SRAM	1Kb ή 2Kb	
EEPROM	512by ή 1kb	
Clock speed	16MHz	
Microcontroller	Atmega168 Atmega328	
Operation Voltage	5V	
Input Voltage	7-12V	
Digital I/O pins	14	
PWM Channels	6	
Analog In channel	8	
Dc current i/o pin	40mA	
Dc current per 3.3v		
Flash memory	16Kb ή 32Kb	
SRAM	1Kb ή 2Kb	
EEPROM	1KB ή 512b	
Clock speed	16MHz	
Microcontroller	ATmega328	
Operation Voltage	2.7-5.5V	
Input Voltage	2.7-5.5V	
Digital I/O pins	14	
PWM Channels	6	
Analog In channel	6	
Dc current i/o pin	40mA	
Dc current per 3.3v		

EIKONA 2.6: Arduino Duemilanove

EIKONA 2.7: Arduino Nano

Flash memory	16Kb	
SRAM	1Kb	
EEPROM	512bytes	
Clock speed	8 MHz	
Microcontroller	ATmega32u4	
Operation Voltage	5V	
Input Voltage	5-9V	
Digital I/O pins	14	
PWM Channels	3	
Analog In channel	6	
Dc current i/o pin		
Dc current per 3.3v		
Flash memory	32Kb	
SRAM	2.5KB	
EEPROM	1KB	
Clock speed	16MHz	

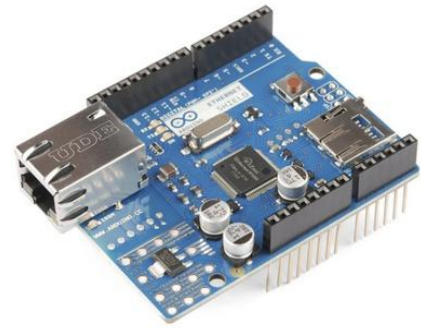
**EIKONA 2.8:** Arduino LilyPad

**EIKONA 2.9:** Arduino NG

## 2.4 Arduino Shield

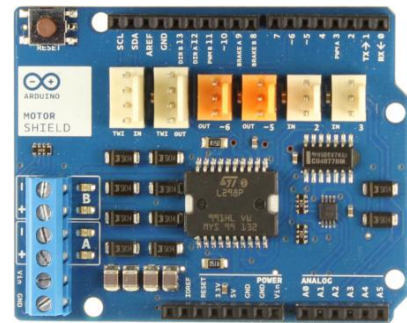
Τα Arduino Shields είναι boards τα οποία χρησιμοποιούνται για να επεκτείνουν τα κυκλώματα και συνδέονται με το Arduino. Τα πιο γνωστά και χρήσιμα είναι για τον έλεγχο κινητήρων, διασύνδεσης ενσύρματα και ασύρματα με το διαδίκτυο, GPS καθώς και οθονών LCD.

Πιο συγκεκριμένα το Ethernet shield αποτελείται από μια υποδοχή για κάρτα microSD όπου χρησιμοποιείται για αποθήκευση και ανάγνωση αρχείων. Συνδέεται με ένα καλώδιο RJ45 για τροφοδοσία (Power Over Ethernet) καθώς και σύνδεση με το διαδίκτυο.



**EIKONA 2.10:** Ethernet shield

Το Motor Shield μπορεί να ελέγξει τουλάχιστον δύο κινητήρες συνεχούς ρεύματος, έλεγχο φορά περιστροφής ταχύτητας. Δηλαδή ελέγχει επαγωγικά φορτία Ρελέ ή κινητήρες DC



**EIKONA 2.11:** Motor shield

Το GSM shield διασυνδέεται με το διαδίκτυο και μπορεί να λάβει μηνύματα ή να στείλει ή να δεχτεί φωνητικές εντολές.



**EIKONA 2.12:** GSM shield

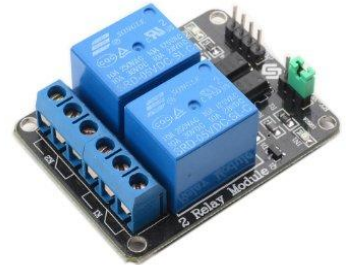


Το *wifi shield* επιτρέπει την ασύρματη δικτύωση με το διαδίκτυο χρησιμοποιώντας, τα πρότυπα 802.11g και 802.11b



**ΕΙΚΟΝΑ 2.13:** wifi shield

Το *Relay shield* επιτρέπει τον έλεγχο της μέσης τάσης, υπάρχουν με 1,2,4,8,16 και 32 κανάλια που είναι εξοπλισμένα με relay. Επιτρέπει να περάσουν 10 A ή 250VAC ή 28Vdc



**ΕΙΚΟΝΑ 2.14:** Relay shield

## 2.5 ΑΡΧΙΤΕΚΤΟΝΙΚΗ ARDUINO UNO

### 2.5.1 ΤΕΧΝΙΚΑ ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ

Στην εφαρμογή που θα υλοποιήσουμε, θα χρησιμοποιήσουμε το μοντέλο του Arduino Uno. Οι λόγοι που θα χρησιμοποιήσουμε αυτή την έκδοση είναι η χαμηλή κατανάλωση ρεύματος αλλά και το χαμηλό κόστος. Παρακάτω, εμφανίζονται τα τεχνικά χαρακτηριστικά του:

- ✓ Μικροελεγκτής: ATMEGA 328
- ✓ Τάση λειτουργίας: 5V
- ✓ Τάση εισόδου: 7-12V
- ✓ Όρια τάσης εισόδου: 6-20V
- ✓ Αναλογικοί ακροδέκτες εισόδου: 6
- ✓ Ψηφιακοί ακροδέκτες εισόδου-εξόδου: 14, 6 PWM έξοδοι

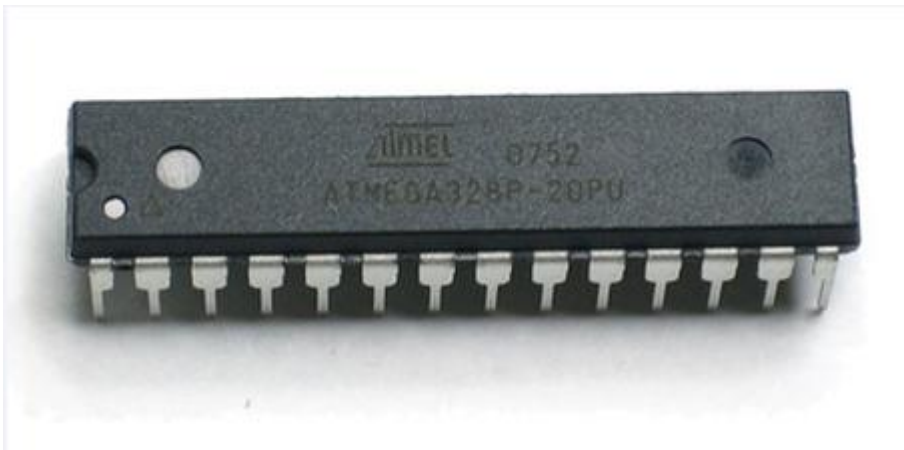
## 2.5.2 ΜΝΗΜΗ

Το Arduino βασίζεται στον ATmega328, έναν 8-bit RISC μικροελεγκτή, ο οποίος χρονίζεται στα 16MHz. Ο ATmega328 διαθέτει ενσωματωμένη μνήμη τριών τύπων :

2kb μνήμης SRAM που είναι ωφέλιμη μνήμη που μπορούν να χρησιμοποιήσουν τα προγράμματά σας για να αποθηκεύουν μεταβλητές, πίνακες κλπ. Κατά το runtime. Όπως και σε έναν υπολογιστή, αυτή η μνήμη χάνει τα δεδομένα της όταν η παροχή ρεύματος στο Arduino σταματήσει ή αν γίνει reset.

1kb μνήμη EEPROM η οποία μπορεί να χρησιμοποιηθεί για εγγραφή/ανάγνωση δεδομένων ανά byte από τα προγράμματά μας κατά το runtime. Σε αντίθεση με την SRAM, η EEPROM δεν χάνει τα περιεχόμενά της με την απώλεια τροφοδοσίας ή reset οπότε είναι το ανάλογο του σκληρού δίσκου.

32Kb μνήμης FLASH, από τα οποία τα 2kb χρησιμοποιούνται από το Firmware του Arduino που έχει εγκαταστήσει ήδη ο κατασκευαστής του. Το Firmware ή Boot loader είναι αναγκαίο για την εγκατάσταση των δικών μας προγραμμάτων στον μικροελεγκτή μέσω της θύρας USB. Τα υπόλοιπα 30Kb της μνήμης Flash χρησιμοποιούνται για την αποθήκευση αυτών ακριβώς των προγραμμάτων, αφού πρώτα μεταγλωττιστούν στον υπολογιστή μας. Η μνήμη Flash και η EEPROM δεν χάνουν τα περιεχόμενά τους με την απώλεια τροφοδοσίας ή reset. Επίσης ενώ η μνήμη Flash υπό κανονικές συνθήκες δεν προορίζονται για χρήση Runtime μέσα από τα προγράμματά μας λόγω της μικρής συνολικής μνήμης που είναι διαθέσιμα σε αυτή (2kb SRAM και 1kb EEPROM), έχει σχεδιαστεί μια βιβλιοθήκη που επιτρέπει την χρήση όσου χώρου περισσεύει. Δηλαδή 30kb μείον το χώρο που πιάνει το πρόγραμμά μας.



**EIKONA 2.15:** ATMEGA328, 8-bit μικροελεγκτής.

### 2.5.3 ΤΡΟΦΟΔΟΣΙΑ

Η τροφοδοσία του Arduino Uno μπορεί να γίνει είτε με ρεύμα είτε μέσα από τον υπολογιστή και πιο συγκεκριμένα μέσω της σύνδεσης USB. Επιπλέον, μπορεί να τροφοδοτηθεί και από εξωτερική τροφοδοσία μέσω της υποδοχής των 2,1 mm που βρίσκεται στην αριστερή γωνία. Για να αποφύγουμε τα προβλήματα, η εξωτερική τροφοδοσία θα πρέπει να είναι από 7-12V. Στην συνέχεια εμφανίζονται αναλυτικά οι ακροδέκτες τροφοδοσίας:

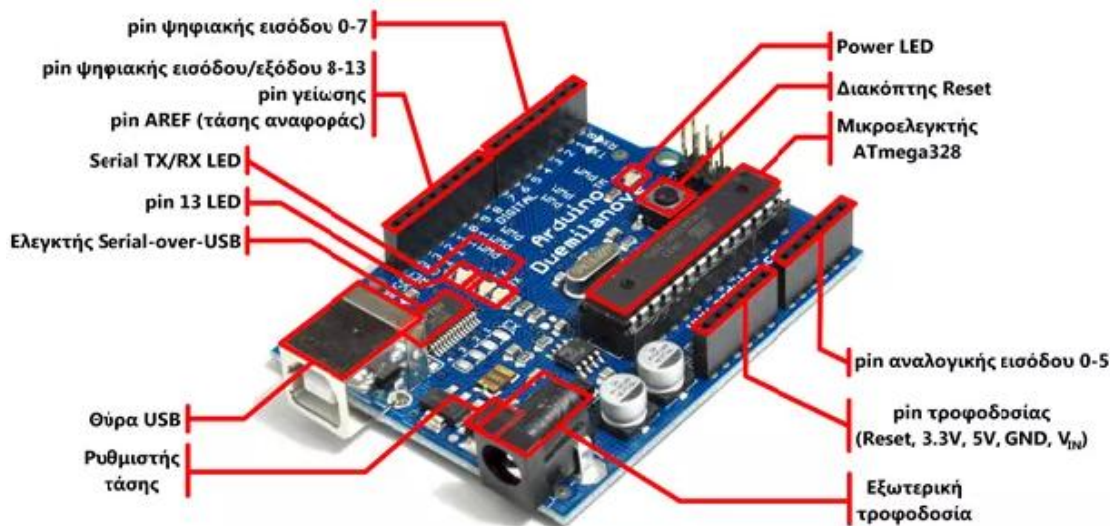
- Vin: Η τάση εισόδου της πλακέτας όταν χρησιμοποιεί εξωτερική πηγή. Η τροφοδοσία της τάσης γίνεται μέσα από αυτόν τον ακροδέκτη.
- 5V: Η τάση που χρησιμοποιείται από τα διάφορα μέρη της πλακέτας και το μικροελεγκτή είναι τα 5V. Αυτή η τάση, την οποία δίνει αυτός ο ακροδέκτης είναι είτε η τάση 5V που δίνει η σύνδεση με τον υπολογιστή μέσω της θύρας USB είτε με ρυθμισμένη τάση που δίνεται μέσω του ακροδέκτη Vin.
- 3.3 V: Αυτή η τάση παράγεται από το ολοκληρωμένο FTDI. Το όριο του ρεύματος είναι τα 50mA.
- GRD: Είσοδοι γειώσεως.



**ΕΙΚΟΝΑ 2.16:** Είσοδοι-έξοδοι της τροφοδοσίας.

### 2.5.4 ΕΙΣΟΔΟΙ-ΕΞΟΔΟΙ

Αρχικά το Arduino διαθέτει Serial interface. Ο μικροελεγκτής ATmega υποστηρίζει σειριακή επικοινωνία, την οποία προωθεί μέσα από έναν ελεγκτή Serial-over-usb ώστε να συνδέεται εύκολα με τον υπολογιστή μέσω usb. Η σύνδεση αυτή χρησιμοποιείται για την μεταφορά του προγράμματος που σχεδιάζουμε στον υπολογιστή, στο Arduino αλλά μπορούμε να δούμε και το πρόγραμμα σε πραγματικό χρόνο καθώς παρέχεται και αυτή η δυνατότητα.



**ΕΙΚΟΝΑ 2.17:** Ακροδέκτες.

Όπως φαίνεται στην εικόνα από πάνω βρίσκονται 14 pin αριθμημένα από 0 έως 13 που είναι αναλογικές ή ψηφιακές εισοδοι και έξοδοι. Λειτουργούν με παροχή στα 5V ή 40mA.

Όταν χρησιμοποιείται σαν ψηφιακή έξοδος τότε στο πρόγραμμά μας είμαστε σε κατάσταση HIGH ή LOW. Με βάση την κατάσταση αυτήν το arduino ξέρει αν θα δώσει ρεύμα στο συγκεκριμένο Pin ή όχι. Έτσι αν συνδεθεί εξωτερική συσκευή μέσω κατάλληλης εντολής μπορούμε διαβάσουμε την κατάστασή του. Ωστόσο κάποια pin από αυτά έχουν και άλλες λειτουργίες :

Τα pin 0 και 1 λειτουργούν ως RX και XT της σειριακής όταν το πρόγραμμά μας ενεργοποιεί τη σειριακή θύρα. Όταν το πρόγραμμα στέλνει τα δεδομένα στην σειριακή αυτά στέλνονται στην USB μέσω του Serial-over-usb αλλά και στον pin 0 για να διαβάσει και μια δεύτερη συσκευή. Αν πραγματοποιήσουμε αυτή τη μέθοδο τότε χάνουμε 2 εισόδους.

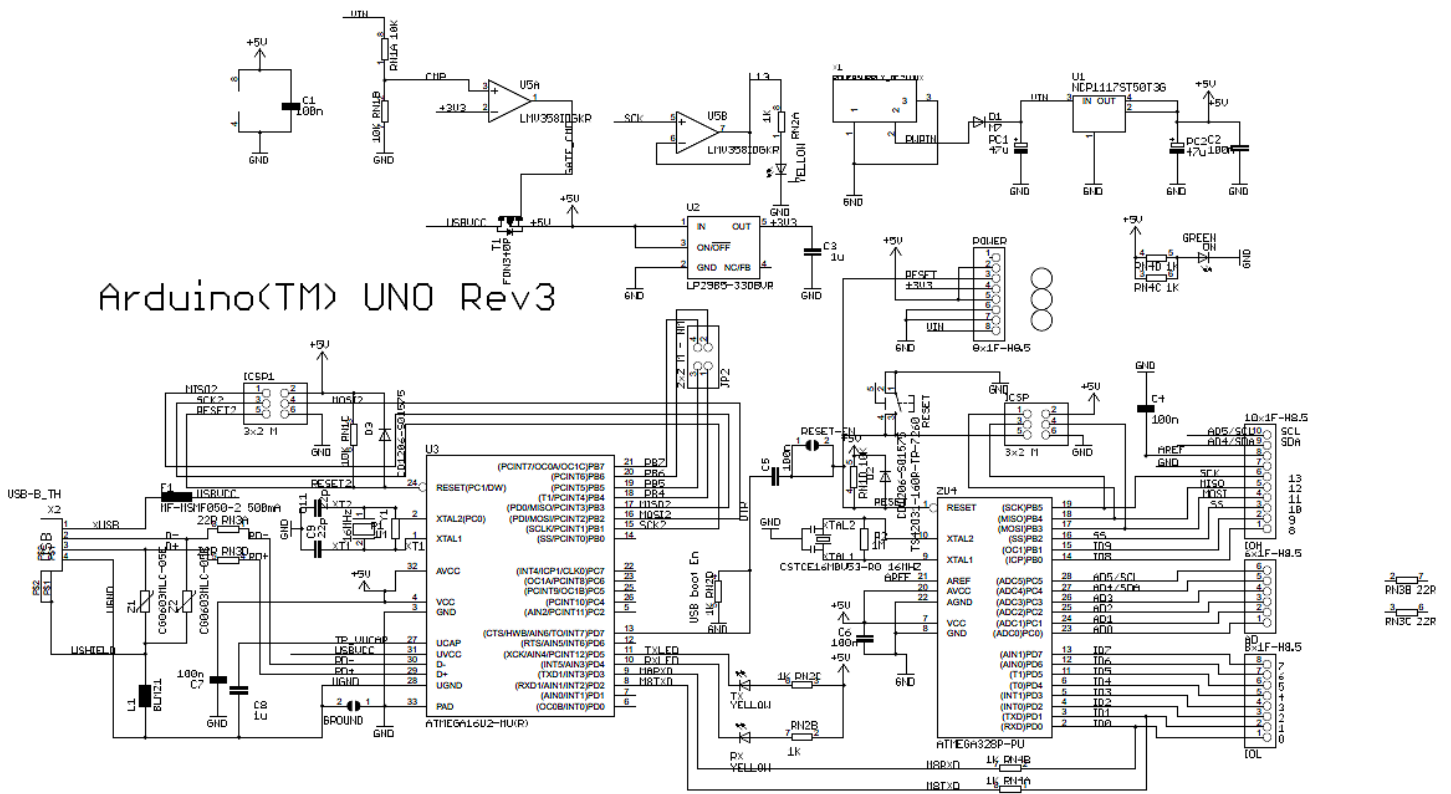
Τα pin 2,3 λειτουργούν και ως εξωτερικοί διακόπτες. Δηλαδή μπορούμε να τα χρησιμοποιήσουμε μέσα από το πρόγραμμά μας να κάνουν συγκεκριμένες αλλαγές, η κανονική ροή σταματάει και εκτελείται η συνάρτηση. Οι εξωτερικοί διακόπτες είναι χρήσιμοι όταν απαιτείται συγχρονισμός μεγάλης ακρίβειας.

Τα pin 3,5,6,9,10,11 λειτουργούν ως αναλογικές έξοδοι με PWM. Ουσιαστικά μέσω του συστήματος αυτού δεν μπορούμε να ρυθμίσουμε την τάση αναλογικά αλλά τον παλμό που δίνει και εναλλάσσεται με μεγάλη συχνότητα και για ίδιους χρόνους μεταξύ των τιμών 0V και 5V

Στην άλλη πλευρά βρίσκουμε μια σειρά με αριθμημένα 0-6 Pin τα οποία λειτουργούν ως αναλογικές εισοδοι κάνοντας χρήση του ADC που ενσωματώνει ο μικροελεγκτής. Μέσα από το πρόγραμμα διαβάζουμε τιμές μέχρι το 1023 όταν η τάση είναι 5V.



Πιο συγκεκριμένα, το Arduino πέραν από την πλακέτα που βλέπουμε εμείς έχει μέσα του αναλογικά και ψηφιακά κυκλώματα για να έρθει στην τελική μορφή της πλακέτας. Συνοπτικά περιέχει τελεστικούς ενισχυτές, μια usb θύρα, Atmega16u2-μικροελεγκτή και Atmega328p-μικροελεγκτή που είναι ο επεξεργαστής και οι μνήμες που περιέχει η πλακέτα. Επίσης υπάρχουν αρκετά ολοκληρωμένα πάνω του. Μερικά από αυτά είναι NCP117ST50T3G, RN4C, CSTCE16M0N53-R0, NF-HSMF050-2. Τέλος, υπάρχουν και κάποιες υποδοχές που μας δίνουν πολύ απλά τις ψηφιακές και αναλογικές εισόδους-εξόδους κάποια από αυτά είναι 10x1F-H8.5, 6x1F-H8.5, Bx1F-H8.5. Στο παρακάτω σχήμα βλέπουμε αναλυτικά το εσωτερικό του.



ΣΧΗΜΑ 2.18: Σχηματικό διάγραμμα αρχιτεκτονικής Arduino.

## 2.6 ARDUINO IDE

Το περιβάλλον ανάπτυξης Arduino εμπεριέχει ένα πρόγραμμα επεξεργασίας κειμένου, για την σύνταξη του κώδικα, ένα πεδίο στο οποίο εμφανίζονται μηνύματα, ένας πίνακας κειμένου και μία γραμμή εργαλείων που έχει την μορφή κουμπιών. Η σύνδεση γίνεται με το υλικό μέρος του arduino (hardware) για να φορτώσει τα προγράμματα και να γίνει η επικοινωνία. Ο κώδικας που περιέχεται στο Arduino ονομάζεται sketch. Εν συνεχεία παρουσιάζονται τα εργαλεία του περιβάλλοντος ανάπτυξης:



**Verify:** Ελέγχει τα συντακτικά λάθη του κώδικα.



**Upload:** Κάνει την μεταγλώττιση του κώδικα και το φορτώνει στο Arduino. Αν η σύνταξη δεν είναι σωστή δεν μπορεί να φορτωθεί.



**New:** Δημιουργεί ένα καινούριο sketch.



**Open:** Δείχνει ένα μενού με όλα τα sketch. Ενεργοποιώντας κάποιο από αυτά, θα ανοίξει αυτόματα στο τρέχον παράθυρο.

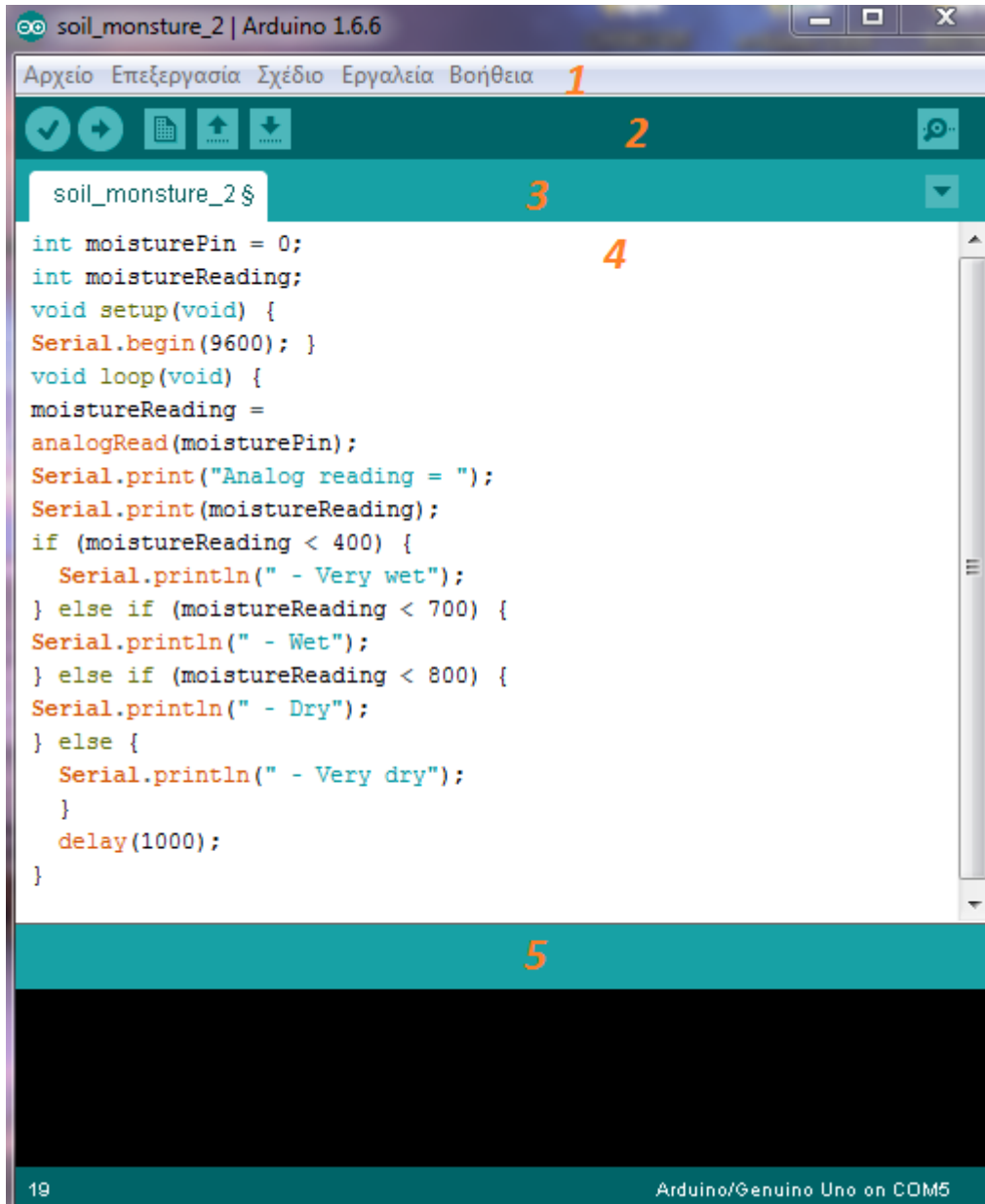


**Save:** Αποθήκευση ενός sketch.



**Serial monitor:** Άνοιγμα της σειριακής οθόνης. Μέσα από αυτή γίνεται η παρακολούθηση της ανταλλαγής δεδομένων που γίνεται στην σειριακή θύρα.

Στο επόμενο σχήμα απεικονίζεται ένα ολοκληρωμένο περιβάλλον ανάπτυξης,



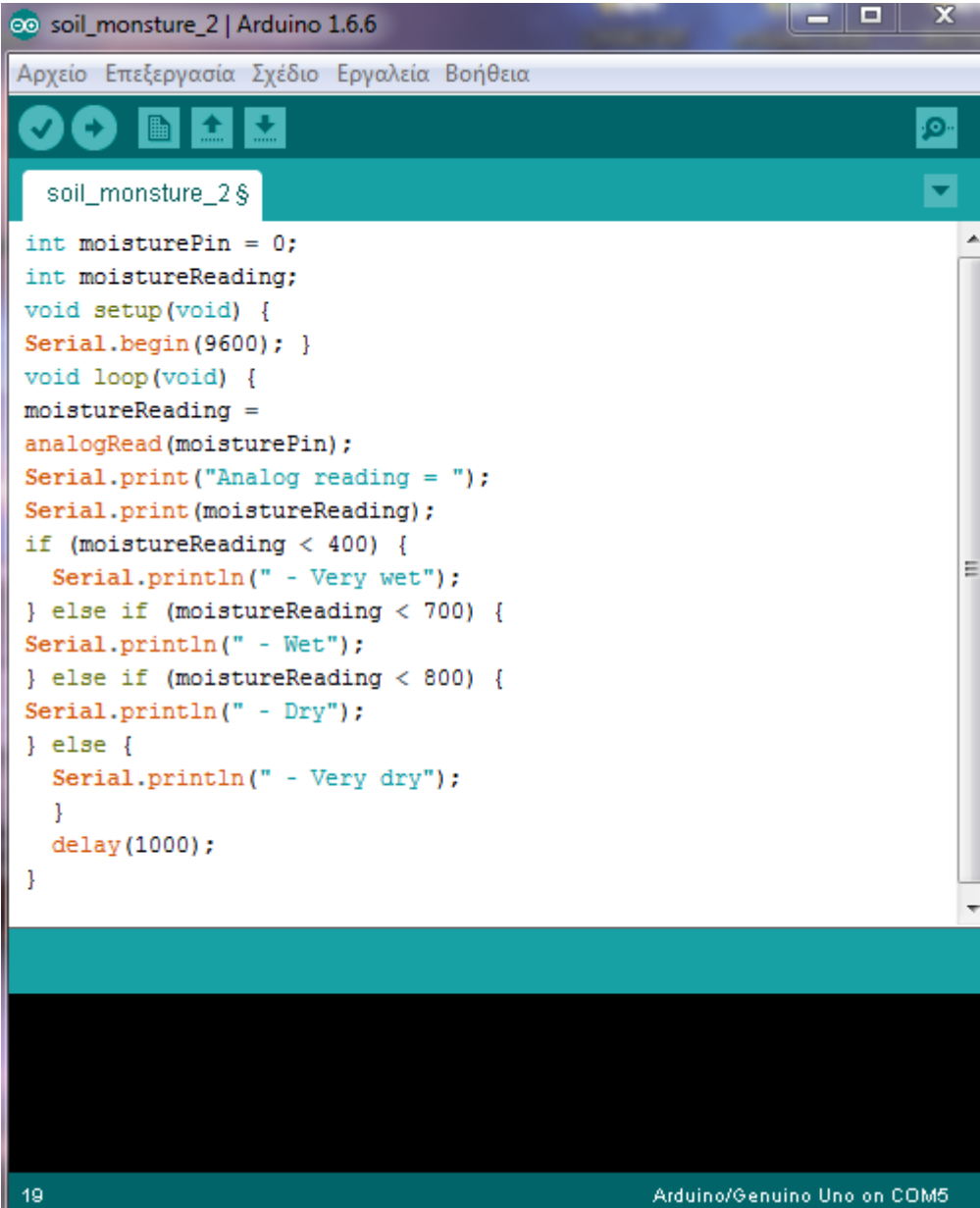
**ΕΙΚΟΝΑ 2.19:** Ολοκληρωμένο περιβάλλον ανάπτυξης.

Όπου :

- 1 είναι το μενού.
- 2 είναι η εργαλειοθήκη
- 3 είναι οι καρτέλες
- 4 είναι η δημιουργία αρχείου
- 5 είναι η κονσόλα μηνύματος

Το Arduino IDE παρέχει:

- Ένα πρακτικό περιβάλλον για την γραφή των προγραμμάτων με συντακτική χρωματική σήμανση.
- Βιβλιοθήκες για προέκταση, που βρίσκονται εύκολα στο διαδίκτυο.
- Compiler (μεταγλωττιστής) για να γίνεται η μεταγλώττιση των sketch.
- Serial monitor (σειριακή οθόνη) που παρακολουθεί τις επικοινωνίες της σειριακής USB, στέλνει αλφαριθμητικά στο Arduino μέσα από αυτή και είναι πολύ χρήσιμο για να αποφευχθούν λάθη των sketch.



```
soil_monsture_2 $
int moisturePin = 0;
int moistureReading;
void setup(void) {
  Serial.begin(9600); }
void loop(void) {
  moistureReading =
  analogRead(moisturePin);
  Serial.print("Analog reading = ");
  Serial.print(moistureReading);
  if (moistureReading < 400) {
    Serial.println(" - Very wet");
  } else if (moistureReading < 700) {
    Serial.println(" - Wet");
  } else if (moistureReading < 800) {
    Serial.println(" - Dry");
  } else {
    Serial.println(" - Very dry");
  }
  delay(1000);
}
```

**ΕΙΚΟΝΑ 2.20:** Πρόγραμμα.

## 2.6.1 ΔΟΜΗ ΚΑΙ ΓΛΩΣΣΑ ΠΡΟΓΡΑΜΜΑΤΟΣ

Η γλώσσα που έχει το Arduino βασίζεται στην γλώσσα wiring μία παραλλαγή της C/C++ για μικροελεγκτές αρχιτεκτονικής AVR όπως ο ATMEGA και υποστηρίζει όλες τις κύριες δομές της C, καθώς επίσης και μερικά χαρακτηριστικά της C++. Για μεταγλωττιστής χρησιμοποιείται ο AVR gcc και ως βιβλιοθήκη η AVR libc. Στη γλώσσα συγγραφής του κώδικα του Arduino, ο οποίος βασίζεται στην C, μπορούν να χρησιμοποιηθούν οι ίδιες βασικές εντολές και συναρτήσεις με την ίδια σύνταξη τους ίδιους τύπους δεδομένων και τους ίδιους τελεστές όπως και στην γλώσσα C. Πέρα από αυτές υπάρχουν και κάποιες ειδικές εντολές, συναρτήσεις και σταθερές που βοηθούν στην διαχείριση του hardware του Arduino.

Τα προγράμματα που περιέχονται στο Arduino διακρίνονται σε τρία κομμάτια:

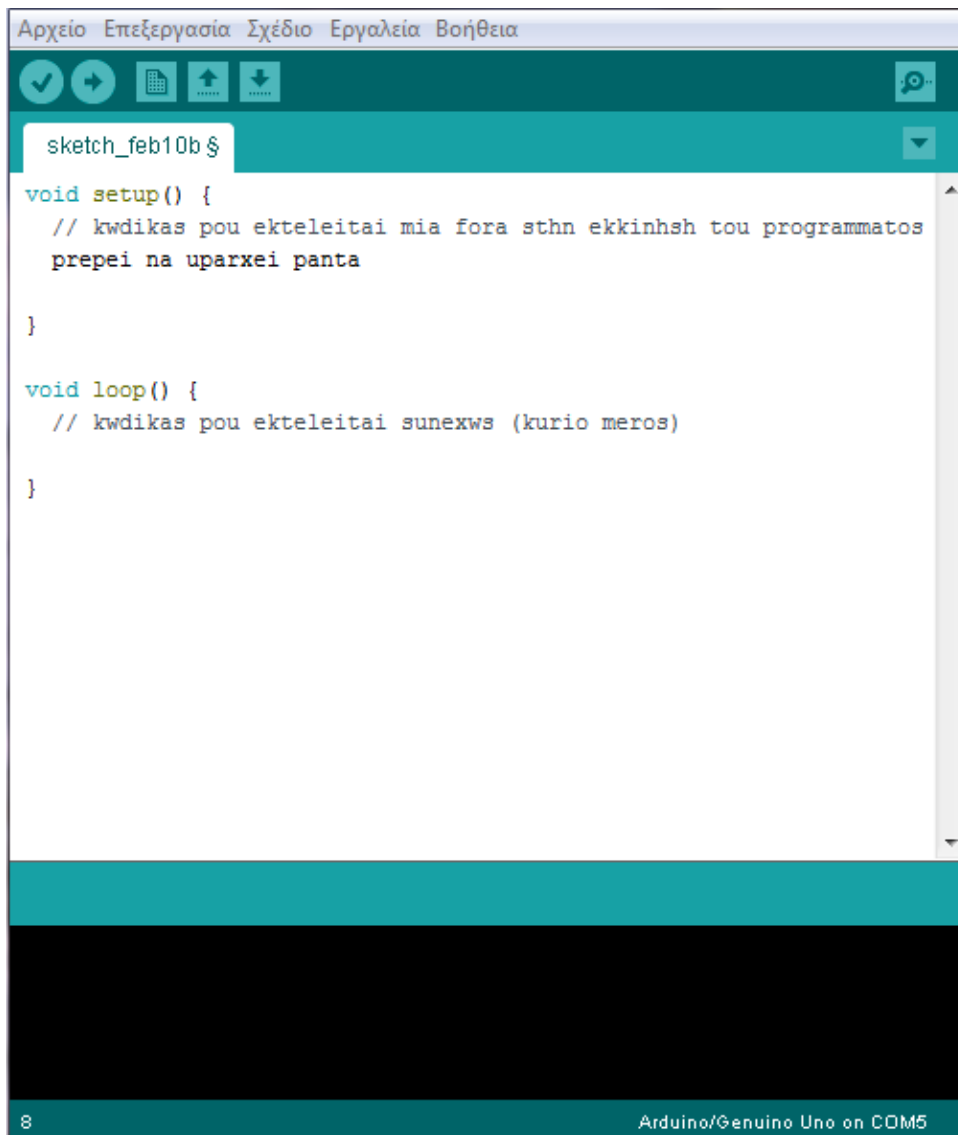
- ☆ Δομή (structure)
- ☆ Τιμές (values)
- ☆ Συναρτήσεις (functions)

Στην παρακάτω εικόνα βλέπουμε την αρχική σελίδα του Arduino καθώς και τις δυο εντολές που εμφανίζονται αυτόματα όταν ανοίγουμε ένα αρχείο.

Η εντολή `void setup()` είναι ο κώδικας που εκτελείται μια φορά στην εκκίνηση του προγράμματος. Υπάρχει πάντα.

Η εντολή `void loop()` κώδικας που εκτελείται συνέχεια είναι το κύριο μέρος.

Τέλος αξίζει να αναφέρουμε ότι το Arduino IDE δεν αναγνωρίζει τα ελληνικά στα σχόλια για αυτό παραθέτουμε με την εντολή `//` (σχόλιο)



**ΕΙΚΟΝΑ 2.21:** Αρχική σελίδα Arduino ide

Παρακάτω, θα αναλύσουμε τις πιο σημαντικές δομές και λειτουργίες έτσι ώστε να μας βοηθήσουν να γράψουμε ένα πρόγραμμα:

### **Δομή**

setup(), loop()

### **Δομές ελέγχου ροής**

**if** Δομή ελέγχου μίας συνθήκης

**if..else** Δομή πολλαπλών συνθηκών

**do..while** Δομή επανάληψης

**for** Δομή επανάληψης

**switch case** δομή ελέγχου περιπτώσεων

**break** Εντολή διακοπής μιας επαναληπτικής δομής

**return** Εντολή επιστροφής από μία συνάρτηση.

**go to** Εντολή μετάβασης σε κάποιο σημείο του κώδικα

**continue** Εντολή παράλειψης της τρέχουσας επανάληψης.

### **Τύποι δεδομένων**

**boolean** Λογική δυαδική τιμή.

**char** Προσημασμένος χαρακτήρας 8 ψηφίων.

**(unsigned)char** Μη προσημασμένος χαρακτήρας 8 ψηφίων.

**byte** Μη προσημασμένος χαρακτήρας 8 ψηφίων.

**int** Προσημασμένος ακέραιος 16 ψηφίων.

**(unsigned)int** Μη προσημασμένος ακέραιος 16 ψηφίων.

**word** Μη προσημασμένος ακέραιος 16 ψηφίων.

**long** Προσημασμένος ακέραιος 32 ψηφίων.

**(unsigned)long** Μη προσημασμένος ακέραιος 32 ψηφίων.

**float, double** Αριθμός κινητής υποδιαστολής απλής ακρίβειας.

**string** Αντικείμενο αλφαριθμητικού με χρήσιμες μεθόδους.

### **Δυαδικοί τελεστές.**

**&, |** Δυαδική σύζευξη – διάζευξη.

**<<, >>** Δυαδική αριστερή – δεξιά ολίσθηση.

**^** Δυαδική αποκλειστική διάζευξη.

**~** Δυαδική άρνηση.

### **Τελεστές αύξησης και μείωσης.**

**++, --** Αύξηση – μείωση κατά μία ακέραια μονάδα

### **Σύνθετοι τελεστές.**

**+=, -=, \*=, /=, %=** σύνθετοι αριθμητικοί τελεστές.

**&=, |=, ^=, ~=, <<=, >>=** Σύνθετοι δυαδικοί τελεστές.

### **Αριθμητικοί τελεστές.**

**=, +, -, \*, /** Τελεστής εκχώρησης, πρόσθεσης, αφαίρεσης, διαίρεσης.

**%** Υπόλοιπο ακέραια διαίρεσης.

<, >, <=, >= Μικρότερο, Μεγαλύτερο, Μικρότερο ή ίσο, Μεγαλύτερο ή ίσο.  
==, != Ισότητα, Ανισότητα.

&&, ||, ! Λογική σύζευξη, διάζευξη, άρνηση.

### **Σταθερές**

**HIGH – LOW** Τιμή υψηλής - χαμηλής στάθμης για μία επαφή εισόδου.

**INPUT – OUTPUT** ορισμό μίας επαφής ως είσοδο – έξοδο.

**True-false** Λογικό επίπεδο αλήθειας - ψεύδους σε μία συνθήκη.

**A0,...,A5** Σταθερές για τις αναλογικές επαφές εισόδου

### **Συναρτήσεις εξόδου-εισόδου**

**pinMode()** Ορίζει μια επαφή ως είσοδο ή έξοδο.

### **Συναρτήσεις αναλογικής εισόδου και εξόδου**

**analogReference()** Ορίζει την τάση αναλογικής εισόδου.

**analogRead()** Διαβάζει από μία αναλογική επαφή εισόδου.

**analogWrite()** Γράφει PWM σήματα σε μία επαφή εξόδου.

### **Συναρτήσεις ψηφιακής εισόδου και εξόδου**

**digitalWrite()** Γράφει μία ψηφιακή επαφή εξόδου.

**digitalRead()** Διαβάζει μία ψηφιακή επαφή εισόδου.

### **Συναρτήσεις χρόνου**

**delay** Παύση προγράμματος

### **Προηγμένες συναρτήσεις εισόδου και εξόδου.**

**tone()**, Παράγει ένα τετραγωνικό σήμα ορισμένης συχνότητας.

**noTone()** Διακόπτει την παραγωγή τετραγωνικών σημάτων.

### **Μαθηματικές και τριγωνομετρικές συναρτήσεις.**

**max(), min()** Βρίσκει τον μεγαλύτερο - μικρότερο ανάμεσα σε δύο αριθμούς.

**abs()** Επιστρέφει την απόλυτη τιμή ενός αριθμού..

**map()** Πραγματοποιεί γραφική παράσταση

**pow()** Επιστρέφει το αποτέλεσμα μίας δύναμης.

**sqrt()** Επιστρέφει την ρίζα ενός αριθμού.

**sin(), cos(), tan()** Υπολογίζει το ημίτονο, συνημίτονο, εφαπτομένη ενός αριθμού.

### **Συναρτήσεις ψευδοτυχαίων αριθμών.**

**random()** Δίδεται ένας νέος αριθμός από την γεννήτρια..

### **Υποστήριξη σειριακής επικοινωνίας.**

**Serial** Αντικείμενο σειριακής επικοινωνίας με χρήσιμες μεθόδους



## 2.6.2 ΒΙΒΛΙΟΘΗΚΕΣ

Οι βιβλιοθήκες μας χρησιμεύουν στην ευκολότερη επικοινωνία με το Arduino ώστε να καταλάβει απευθείας τη χρησιμοποιούμε για παράδειγμα ethernet shield, relay και τα λοιπά. Αξίζει να αναφέρουμε ότι οι βιβλιοθήκες περιέχουν τον κώδικα οδήγησης της πλακέτας που χρησιμοποιούμε για επικοινωνία. Οι βιβλιοθήκες δηλώνονται στην αρχή του προγράμματος με το σύμβολο # αυτό έχει σαν αποτέλεσμα να κατάλαβαίνει το Arduino τι προσθέσαμε πάνω του ώστε να το προγραμματίσουμε εύκολα.  
Κάποιες βιβλιοθήκες είναι:

### **Καθιερωμένες βιβλιοθήκες:**

Ethernet: Σύνδεση με το διαδίκτυο χρησιμοποιώντας την πλακέτα (ενσύρματα)

Wifi : Σύνδεση με το διαδίκτυο χρησιμοποιώντας την πλακέτα (ασύρματα)

GSM: Σύνδεση σε δίκτυο κινητής τηλεφωνίας με την πλακέτα

EEPROM: Εγγραφή και ανάγνωση της μνήμης

SPI: Επικοινωνία με δίαυλο SPI

SD: Εγγραφή και ανάγνωση κάρτας μνήμης

WIRE: Αποστολή δεδομένων εγγραφή μέσω δικτύων (TWI/I2C)

TFT: Για οθόνη TFT προβολή κειμένου.

SERVO: Για έλεγχο σερβοκινητήρων

Firmata: Επικοινωνία με υπολογιστή με σειριακό πρωτόκολλο

NewSoftSerial: Βελτιωμένη βιβλιοθήκη της SoftwareSerial

SoftwareSerial: Σειριακή επικοινωνία σε ακίδες.

### **Επικοινωνίας (δικτύωση-πρωτόκολλα):**

Messenger: Επεξεργασία κειμένου από τον υπολογιστή

Ps2keyboard: Διαβάζει πληκτρολόγιο τύπου PS2

SSerial2mobile: Αποστολή κειμένου μέσω κινητού τηλεφώνου

x10: Αποστολή σημάτων μέσω AC ρεύματος

### **Συχνότητα παραγωγής ήχου :**

Tone: Παράγει κύματα ήχου

### **Χρονοδιάγραμμα:**

datatime: τρέχουσα ημερομηνία και ώρα.

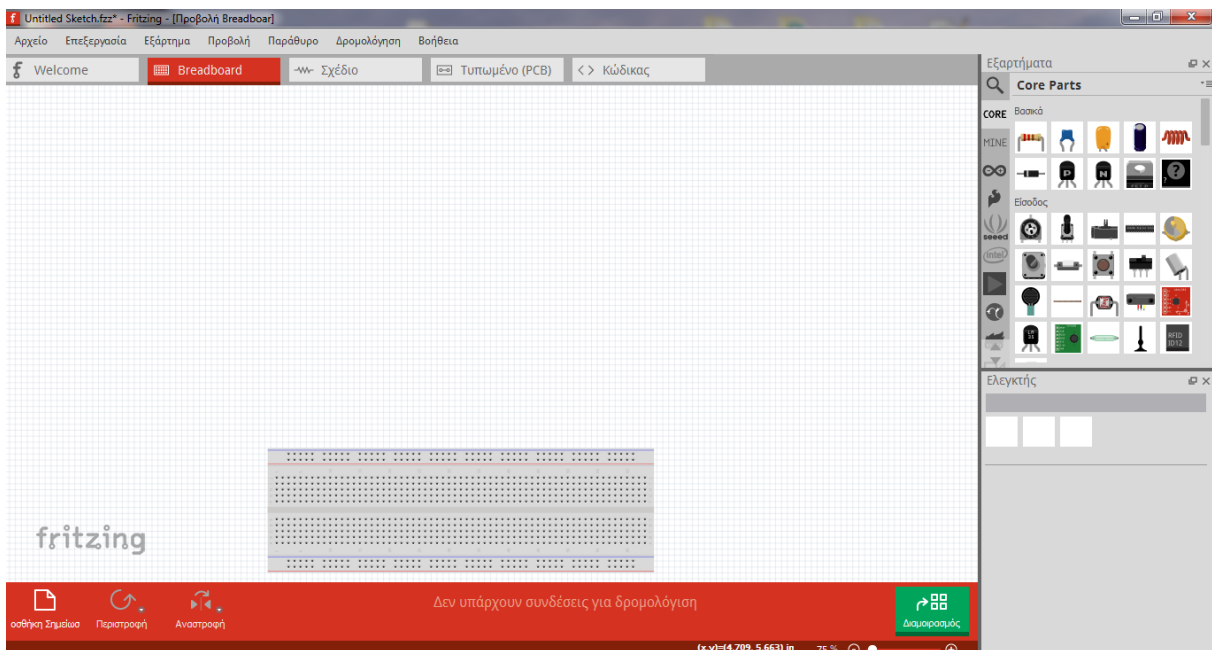
### **Οδήγηση LED:**

Ledcontrol: έλεγχο led ή 7segment display

Leddisplay: πολλαπλή οδήγηση led

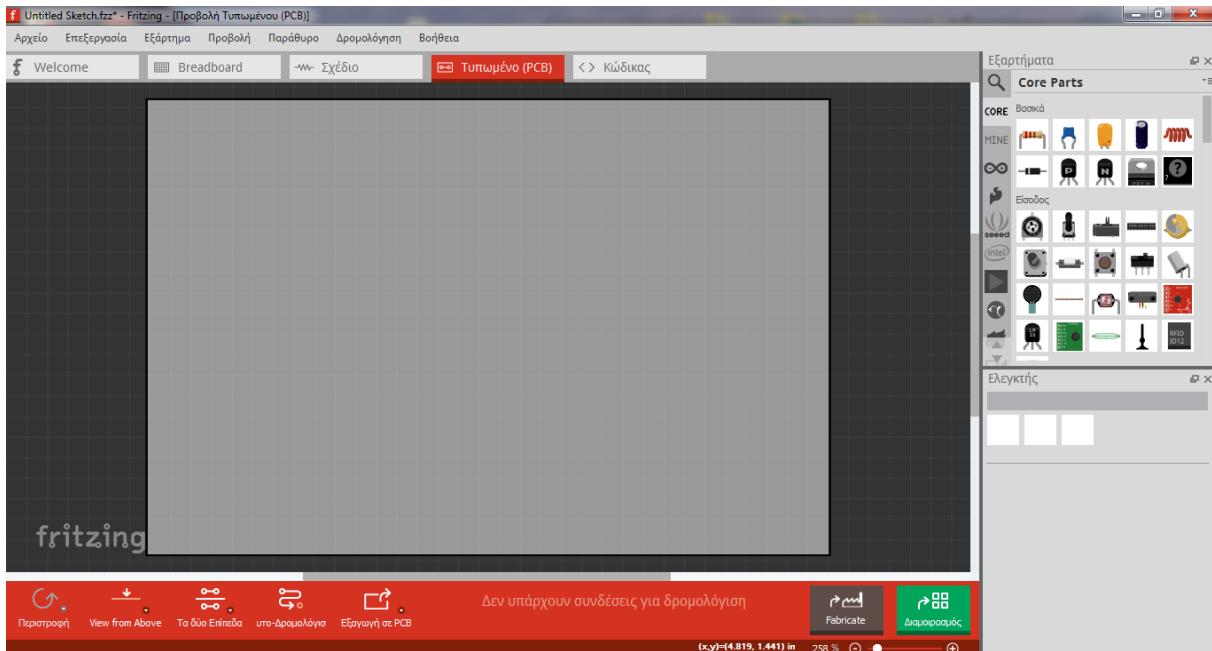
## 2.7 Fritzing

Το fritzing είναι ένα εργαλείο προγραμματισμού με εύκολη επικοινωνία προς το χρήστη. Η ιδιαιτερότητα που έχει είναι ότι μπορεί να φτιάξει το κύκλωμα εικονικά με γραφικά στοιχεία. Μπορεί επίσης να φτιάξει το τυπωμένο κύκλωμα (PCB) καθώς να γράψει τον κώδικα από την αρχή και να τον φορτώσει πάνω στο Arduino. Ουσιαστικά είναι τρία ξεχωριστά πράγματα σε ένα. Όπως βλέπουμε στις εικόνες έχουμε την αρχική σελίδα για να δημιουργηθεί εύκολα το κύκλωμα σε σχέδιο (breadboard):



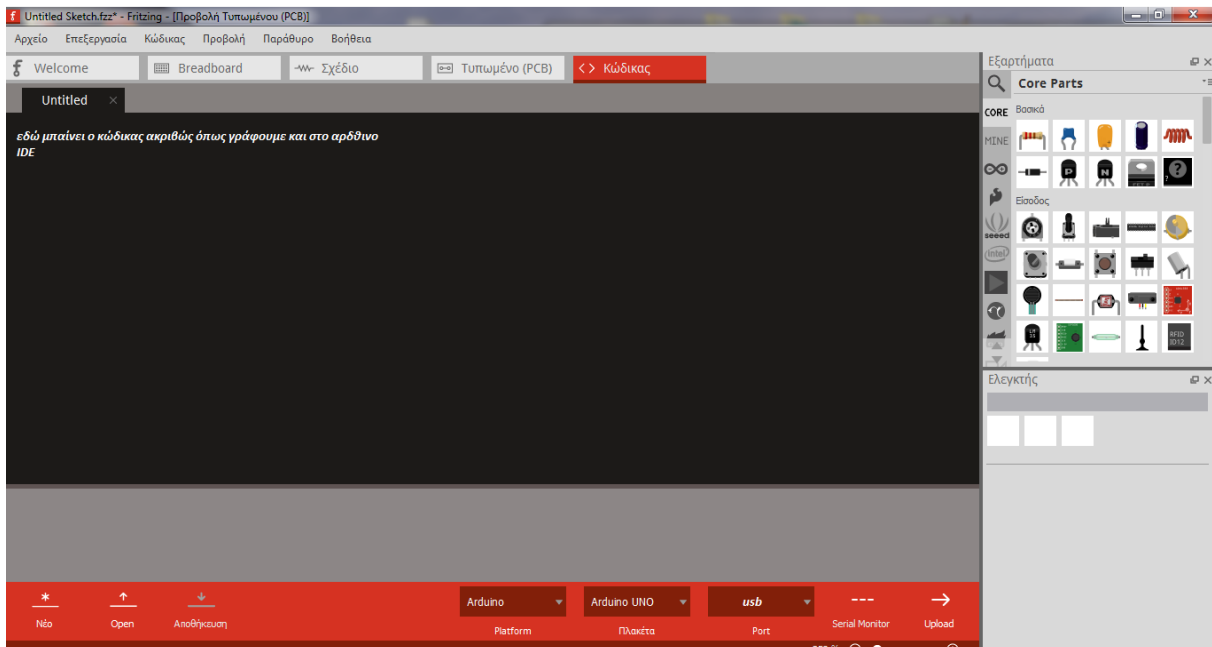
**EIKONA 2.22:** Αρχική σελίδα για breadboard.

Η αρχική σελίδα του τυπωμένου (PCB):



**ΕΙΚΟΝΑ 2.23:** Αρχική σελίδα τυπωμένου PCB

Και για τον κώδικα:



**ΕΙΚΟΝΑ 2.24:** Αρχική σελίδα για τον κώδικα

Και στις τρεις περιπτώσεις αλλάζει το φόντο της αρχικής σελίδας . Στην δεξιά πλευρά υπάρχουν τα εξαρτήματα που χρησιμοποιούμε για τα περισσότερα κυκλώματα. Αν χρειαστούμε κάτι που δεν υπάρχει τότε μπορούμε να το κατεβάσουμε και να το εγκαταστήσουμε πολύ εύκολα.

Επίσης η τοποθέτηση γίνεται πολύ εύκολη απλά σέρνοντας τα εξαρτήματα από την δεξιά μεριά στην αριστερή και ενώνοντας με τα καλώδια. Τέλος αξίζει να αναφερθεί ότι είναι ανοιχτού κώδικα και δωρεάν.

## **ΚΕΦΑΛΑΙΟ 3: ΕΡΓΑΛΕΙΑ ΠΟΥ ΘΑ ΧΡΗΣΙΜΟΠΟΙΗΣΟΥΜΕ ΓΙΑ ΤΗΝ ΕΦΑΡΜΟΓΗ ΜΑΣ**

### 3.1 ΠΕΡΙΓΡΑΦΗ ΤΟΥ ΣΥΣΤΗΜΑΤΟΣ

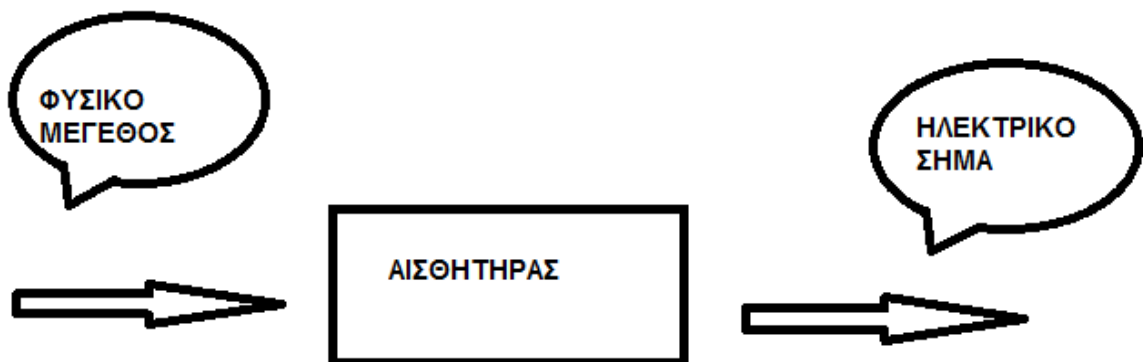
Η εφαρμογή η οποία θέλουμε να φτιάξουμε, είναι η δημιουργία ενός αυτόνομου συστήματος που θα μπορεί να ποτίζει τα φυτά χωρίς να κάνουμε τίποτα. Αυτά που θα χρειαστούμε έτσι ώστε να λειτουργεί η εφαρμογή είναι η πλατφόρμα Arduino Uno r3, μία αντλία νερού, δύο channel relay, ένας αισθητήρας υγρασίας χώματος, ένας αισθητήρας θερμοκρασίας και ένα shield για τον έλεγχο του νερού. Πιο γενικά, μπορούμε να τα κατηγοριοποιήσουμε στα εξής:

- ⊕ Μικροελεγκτής Arduino Uno
- ⊕ Αισθητήρες περιβάλλοντος
- ⊕ Ελεγχόμενες συσκευές
- ⊕ Κατασκευή μακέτας για την υλοποίηση του συστήματος

### 3.2 ΠΕΡΙΓΡΑΦΗ ΤΩΝ ΑΙΣΘΗΤΗΡΩΝ

Ο αισθητήρας (sensor) είναι μία διάταξη που χρησιμοποιείται για την μέτρηση κάποιου μεγέθους. Μετατρέπει το φυσικό μέγεθος δηλαδή το μετρούμενο μέγεθός μας σε ένα ηλεκτρικό σήμα. Εδώ, να αναφέρουμε ότι με τον όρο ηλεκτρικό σήμα εξόδου ενός αισθητήρα, εννοούμε είτε την τάση (όταν ο αισθητήρας μετατρέπει το μετρούμενο μέγεθος σε τάση) είτε το ρεύμα (όταν ο αισθητήρας μετατρέπει το μετρούμενο μέγεθος σε ρεύμα).

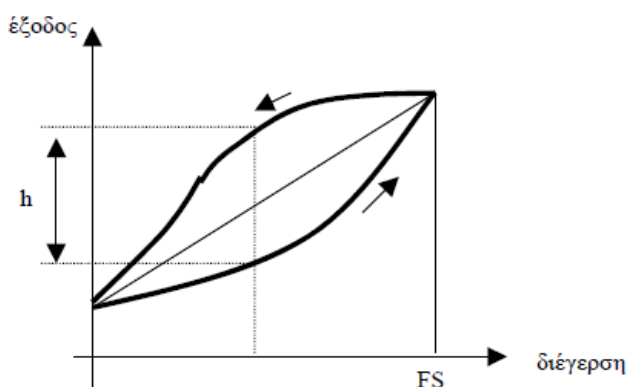
Κάποια παραδείγματα φυσικών μεγεθών που μετρούνται με αισθητήρες είναι η θερμοκρασία, η υγρασία, η τάση, το ρεύμα και άλλα. Συχνά οι αισθητήρες δεν δίνουν στην έξοδό τους το κατάλληλο ηλεκτρικό σήμα. Έτσι απαιτείται η χρήση ενός επιπρόσθετου ηλεκτρονικού κυκλώματος, το οποίο να λαμβάνει την έξοδο του αισθητήρα και να την μετατρέπει σε κατάλληλο ηλεκτρικό σήμα. Αυτό το κύκλωμα ονομάζεται κύκλωμα ρύθμισης σήματος (signal conditioning circuit), κύκλωμα ελέγχου (control circuit) ή εξωτερική μονάδα (external module).



**ΣΧΗΜΑ 3.1:** Σχηματική διάταξη ενός αισθητήρα.

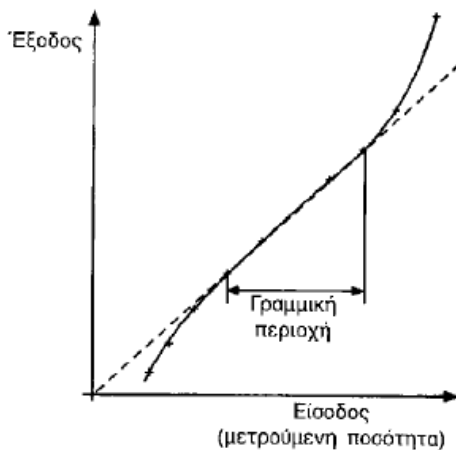
## ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ ΑΙΣΘΗΤΗΡΩΝ

- ✓ Ακρίβεια(accuracy): Η ακρίβεια ενός αισθητήρα είναι το εύρος του αναμενόμενου σφάλματος μεταξύ της μετρούμενης τιμής και της πραγματικής τιμής. Για να προσδιορίσουμε την ακρίβεια, λαμβάνουμε υπόψη όλες τις δυνατές αιτίες σφαλμάτων. Συνήθως η ακρίβεια εκφράζεται σαν ποσοστό επί του δυνατού εύρους τιμών εξόδου του αισθητήρα.
- ✓ Εύρος τιμών (range): Το εύρος των τιμών ενός αισθητήρα είναι το διάστημα τιμών, μέσα στο οποίο μπορεί να μεταβάλλεται η είσοδος του αισθητήρα (φυσικό μέγεθος), έτσι ώστε ο αισθητήρας να μας δίνει σίγουρα αποτελέσματα.
- ✓ Ευαισθησία (sensitivity): Η ευαισθησία ενός αισθητήρα, εκφράζει το κατά πόσο μεταβάλλεται η έξοδος του αισθητήρα ανά μονάδα μεταβολής της εισόδου του.
- ✓ Στατικό σφάλμα (static error): Το στατικό σφάλμα ενός αισθητήρα είναι η απόκλιση ανάμεσα στην μετρούμενη και την πραγματική τιμή μετά από σταθεροποίηση. Οφείλεται σε διάφορες αιτίες και είναι συνήθως διαφορετικά για κάθε μετρούμενη τιμή.
- ✓ Σφάλμα υστέρησης( hysteresis): Ένας αισθητήρας μπορεί να δώσει μία διαφορετική μέτρηση για την ίδια τιμή κάποιας μετρούμενης ποσότητας, ανάλογα με την μορφή της μεταβολής της εισόδου του αισθητήρα. Η απόκλιση που έχουν οι διαφορετικές μετρήσεις είναι το σφάλμα υστέρησης.



**ΣΧΗΜΑ 3.2:** Χαρακτηριστική υστέρησης.

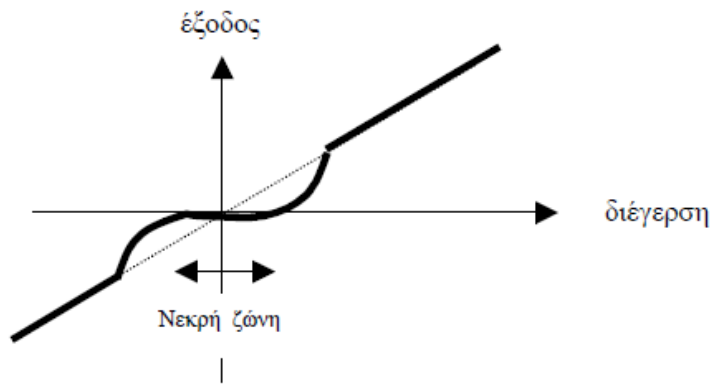
- ✓ Διακριτική ικανότητα (resolution): Η διακριτική ικανότητα του αισθητήρα εκφράζει τη μικρότερη δυνατή μεταβολή της εισόδου του αισθητήρα που μπορεί να μετρηθεί. Όσο μικρότερη είναι αυτή η μεταβολή τόσο μεγαλύτερη είναι η διακριτική ικανότητα του αισθητήρα.
- ✓ Γραμμικότητα (linearity): Είναι ο βαθμός στον οποίο η γραφική παράσταση της εξόδου ως προς την είσοδο ενός αισθητήρα συγκλίνει προς μία ευθεία γραμμή. Ένας αισθητήρας μπορεί να είναι γραμμικός σε μία περιοχή τιμών εισόδου και η γραμμικότητα δίνεται ως ποσοστό επί του εύρους λειτουργίας.



**ΣΧΗΜΑ 3.3:** Γραφική παράσταση εξόδου ως προς είσοδο του αισθητήρα.

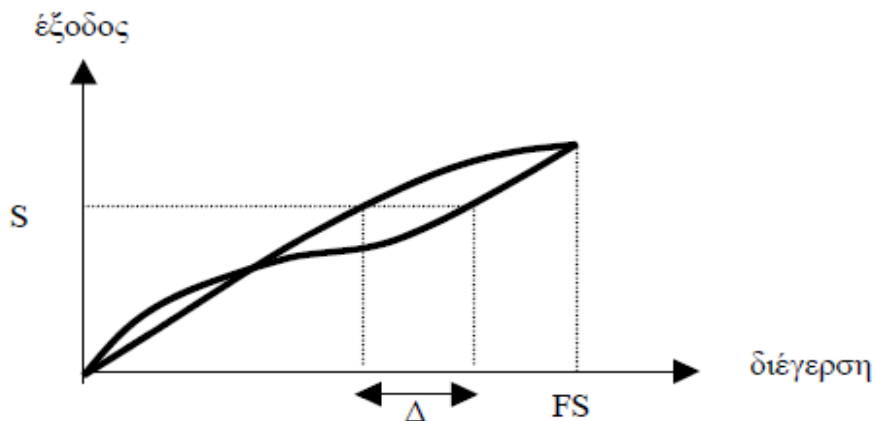
- ✓ Νεκρή ζώνη (dead zone): Όταν η είσοδος ενός αισθητήρα αυξάνεται αφού ξεκινάει από μικρές τιμές, υπάρχει περίπτωση να μην γίνεται αντιληπτή η μεταβολή από τον αισθητήρα, μέχρι η τιμή της εισόδου να ξεπεράσει ένα κατώφλι. Η τιμή του κατωφλίου πέρα από το οποίο γίνεται αντιληπτή η μεταβολή της εισόδου από τον αισθητήρα ονομάζεται νεκρή ζώνη.





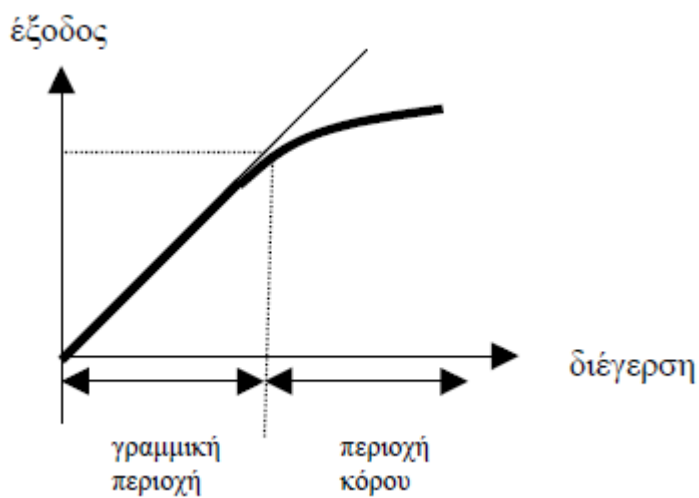
**ΣΧΗΜΑ 3.4:** Χαρακτηριστική νεκρής ζώνης.

- ✓ Επαναληψιμότητα (repeatability): Οι μεταβολές της εισόδου του αισθητήρα έχουν επίδραση στην τιμή της μέτρησης, όταν για παράδειγμα ο αισθητήρας παρουσιάζει υστέρηση. Έτσι, μπορεί να εμφανιστούν αποκλίσεις μεταξύ των επαναλαμβανόμενων μετρήσεων της ίδιας τιμής της εισόδου του αισθητήρα. Το σφάλμα που εμφανίζουν οι διαφορετικές μετρήσεις της ίδιας τιμής της εισόδου του αισθητήρα χαρακτηρίζει την επαναληψιμότητα του αισθητήρα και εκφράζεται σαν ποσοστό επί του συνολικού εύρους τιμών. Όσο μικρότερο είναι αυτό το ποσοστό τόσο καλύτερη είναι η επαναληψιμότητα του αισθητήρα.



**ΣΧΗΜΑ 3.5:** Χαρακτηριστική σφάλματος επαναληψιμότητας.

- ✓ Κορεσμός(saturation): Σχεδόν όλα τα αισθητήρια έχουν όρια λειτουργίας. Έτσι για κάθε ένα αισθητήριο το οποίο έχει γραμμική συμπεριφορά, υπάρχει κάποια τιμή διέγερσης πέρα από την οποία το αισθητήριο δεν έχει την ίδια απόκριση. Στην περιοχή κορεσμού ή (κόρου), η περαιτέρω αύξηση της διέγερσης δεν ισοδυναμεί με ανάλογη αύξηση του σήματος εξόδου. Η αύξηση θα είναι μικρότερη και τελικά μηδενική.



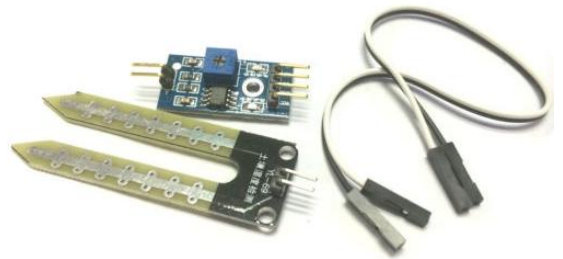
**ΣΧΗΜΑ 3.6:** Χαρακτηριστική κόρου.

### 3.3 ΠΕΡΙΓΡΑΦΗ ΑΙΣΘΗΤΗΡΩΝ ΠΕΡΙΒΑΛΛΟΝΤΟΣ

Χρησιμοποιούμε αναλογικούς και ψηφιακούς αισθητήρες όπου τους συνδέουμε στις εισόδους του Arduino. Η επιλογή των αισθητήρων έγινε με βάση την λειτουργία του Arduino χωρίς δεύτερη εξωτερική πηγή παρά μόνο την τάση 9V είτε 5V όπου η μια δίνετε μέσω μπαταρίας και η άλλη από θύρα USB. Οι αισθητήρες επίσης δένουν πάνω στην πλακέτα προγραμματίζονται και δουλεύουν κανονικά. Τέλος, επιλέχθηκαν αισθητήρες με μικρό κόστος.

### 3.3.1 Αισθητήρας υγρασίας χώματος

Για να μπορέσουμε να μετρήσουμε την υγρασία από το χώμα χρησιμοποιούμε ένα πολύ απλό αισθητήρα με ενσωματωμένο αισθητήριο LM35 καθώς και την πλακέτα ώστε να μπορέσουμε να ελέγξουμε καλύτερα το αισθητήριο.



**ΕΙΚΟΝΑ 3.7:** Αισθητήρας υγρασίας χώματος

**Γενικά χαρακτηριστικά του :**

VCC: 3.3V-5.5V

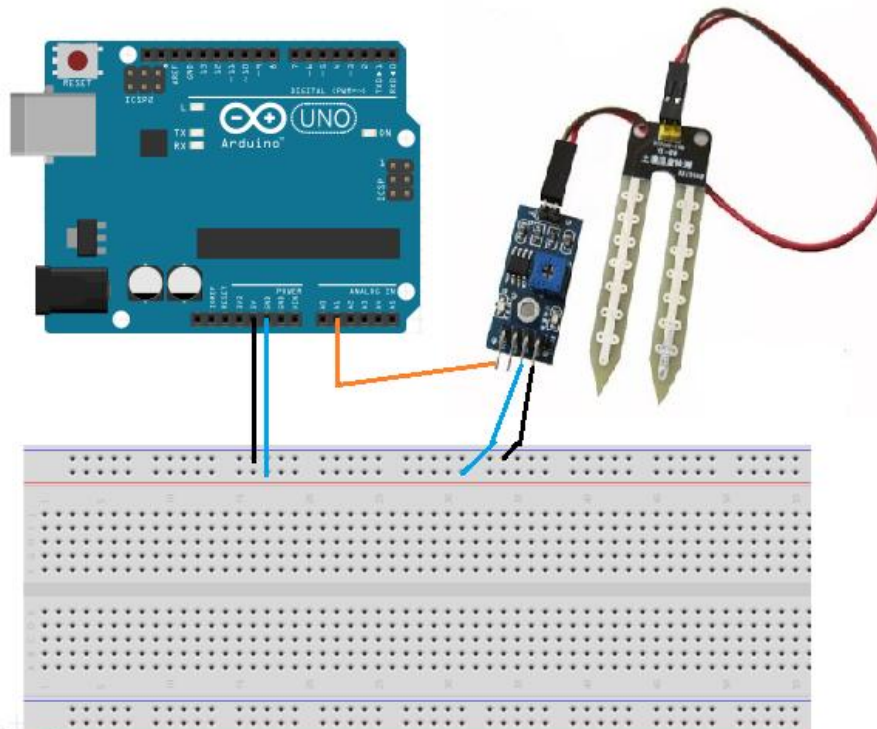
GND

DO: ψηφιακή έξοδο (0 ή 1)

AO: αναλογική έξοδος

**Ο προγραμματισμός του:**

Η σύνδεση γίνεται πολύ απλά συνδέοντας την τάση 5V,GND A1



fritzing

**ΣΧΗΜΑ 3.8:** Σύνδεση αισθητήρα υγρασίας χώματος

Ο προγραμματισμός του γίνεται με δυο τρόπους είτε με ποσοστό είτε αναλογικά.

Με ποσοστό γίνεται :

```
void setup() {  
  Serial.begin(9600);  
}  
void loop() {  
  int humidityRaw = analogRead(A0);  
  int humidityReal = map(humidityRaw, 1023, 0, 0, 100);  
  
  Serial.println(humidityReal);  
  delay(100);  
}
```

Είτε αναλογικά εάν θεωρήσουμε ότι :

Από 0-400 = πολύ βρεγμένο  
400-700 = βρεγμένο  
700-800 = ξερό  
800-1023 = πολύ ξερό

```
int ugrasiaPin = 0;  
int ugrasiaReading;  
void setup(void) {  
  
  Serial.begin(9600); }  
  
void loop(void)  
{  
  
  ugrasiaReading = analogRead(ugrasiaPin);  
  
  Serial.print("Υgrasia = ");  
  
  Serial.print(ugrasiaReading);  
  
  if (ugrasiaReading < 400) {  
    Serial.println(" polu vregmeno ");  
  
  }  
  else if (ugrasiaReading < 700) {  
    Serial.println(" vregmeno");  
  
  }  
  else if (ugrasiaReading < 800) {  
    Serial.println(" ksero ");  
  }  
  else {  
    Serial.println(" polu ksero ");  
  }  
}
```

```
}  
  delay(1000);  
}
```

### 3.3.2 Αισθητήρας βροχής

Ο αισθητήρας βροχής καταγράφει αν υπάρχει βροχή ή όχι. Μπορεί να χρησιμοποιηθεί σαν διακόπτης είτε σαν μετρητής βροχής.

Γενικά χαρακτηριστικά :

VCC: 3.3V-5.5V

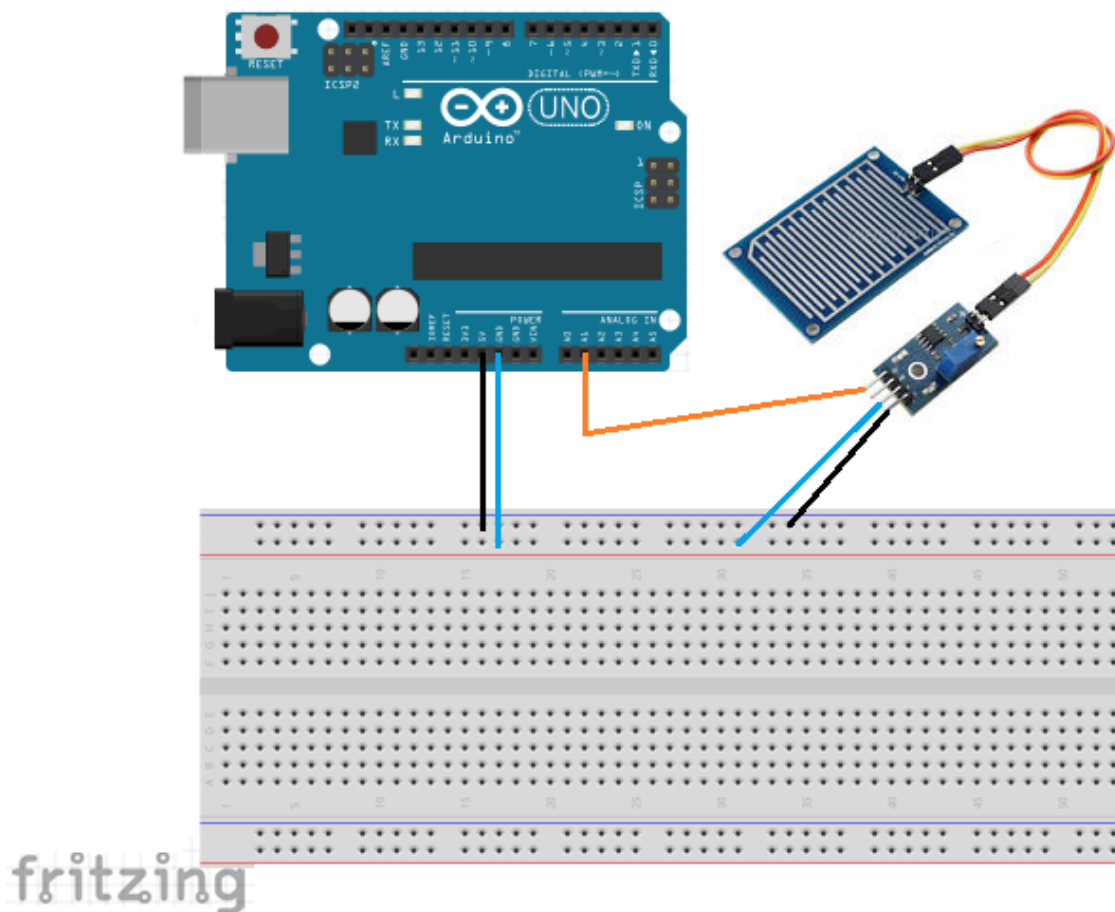
GND

DO: ψηφιακή έξοδο (0 ή 1)

AO: αναλογική έξοδος



**ΕΙΚΟΝΑ 3.9:** Αισθητήρας βροχής



**ΣΧΗΜΑ 3.10:**Σύνδεση αισθητήρα βροχής

## Ο προγραμματισμός του :

Η σύνδεση γίνεται πολύ απλά συνδέοντας την τάση 5V,GND A0

Ο προγραμματισμός του γίνεται με δυο τρόπους είτε με ποσοστό είτε αναλογικά.

Με ποσοστό γίνεται :

```
void setup() {  
  Serial.begin(9600);  
  
}  
  
void loop() {  
  int rainyRaw = analogRead(A0);  
  int rainyReal = map(rainyRaw, 1023, 0, 0, 100);  
  
  Serial.println(rainyReal);  
  delay(100);  
}
```

Είτε αναλογικά αν θεωρήσουμε ότι :

Από 0-500 = βροχή  
500-800 = ψιχαλίζει  
800-1023 = δεν βρέχει

```
int RainyPin = 0;  
int RainyReading;  
void setup(void) {  
  Serial.begin(9600);  
}  
void loop(void) {  
  
  RainyReading = analogRead(RainyPin);  
  
  Serial.print("vrexai = ");  
  
  Serial.print(RainyReading);  
  
  if(RainyReading < 500) {  
  
    Serial.println(" vrexai");  
  }  
  else if (RainyReading < 800) {  
  
    Serial.println(" psixalizei");  
  }  
  else {  
    Serial.println(" den vrexai");  
  }  
}
```

```
delay(1200);
```

### 3.3.3 Συσκευές ελέγχου

#### A) 2 Channel Relay :

Το ρελέ θα μας επιτρέψει να ελέγξουμε την ηλεκτροβάννα καθώς αντλία νερού.



**ΕΙΚΟΝΑ 3.11:** relay

#### Γενικά χαρακτηριστικά:

Τάση : AC250V, 10A ,  
DC30V , 10A

συμβατό με Arduino, 8051,avr, pic, dsp, Arm.

#### Ο προγραμματισμός του :

Η σύνδεση γίνεται πολύ απλά είτε από εξωτερική τροφοδοσία είτε από τα 5V του arduino.

```
void setup () {  
  pinMode (5, OUTPUT);  
}  
void loop(){  
  pinMode(5, OUTPUT);  
  delay(1000);  
  digitalWrite(5,LOW);  
  delay(1000);  
}
```

#### B) Έλεγχος ροής νερού

Τον έλεγχο νερού θα τον πραγματοποιήσουμε μέσω του αισθητήρα μας. Ο αισθητήρας έχει μια φτερωτή που όταν περνάει το νερό αυτή γυρνάει και έτσι βλέπει ότι υπάρχει νερό

**Χαρακτηριστικά :**

Μικρότερη τάση λειτουργίας : 4.5DC (5-24V)

Μέγιστο ρεύμα: 15mA (DC 5v)

Θερμοκρασία -25 εως 80 βαθμούς

Επιτρεπτή υγρασία 35-90%RH

Επιτρεπτή πίεση 2bar

Λειτουργία 1εως30L/min

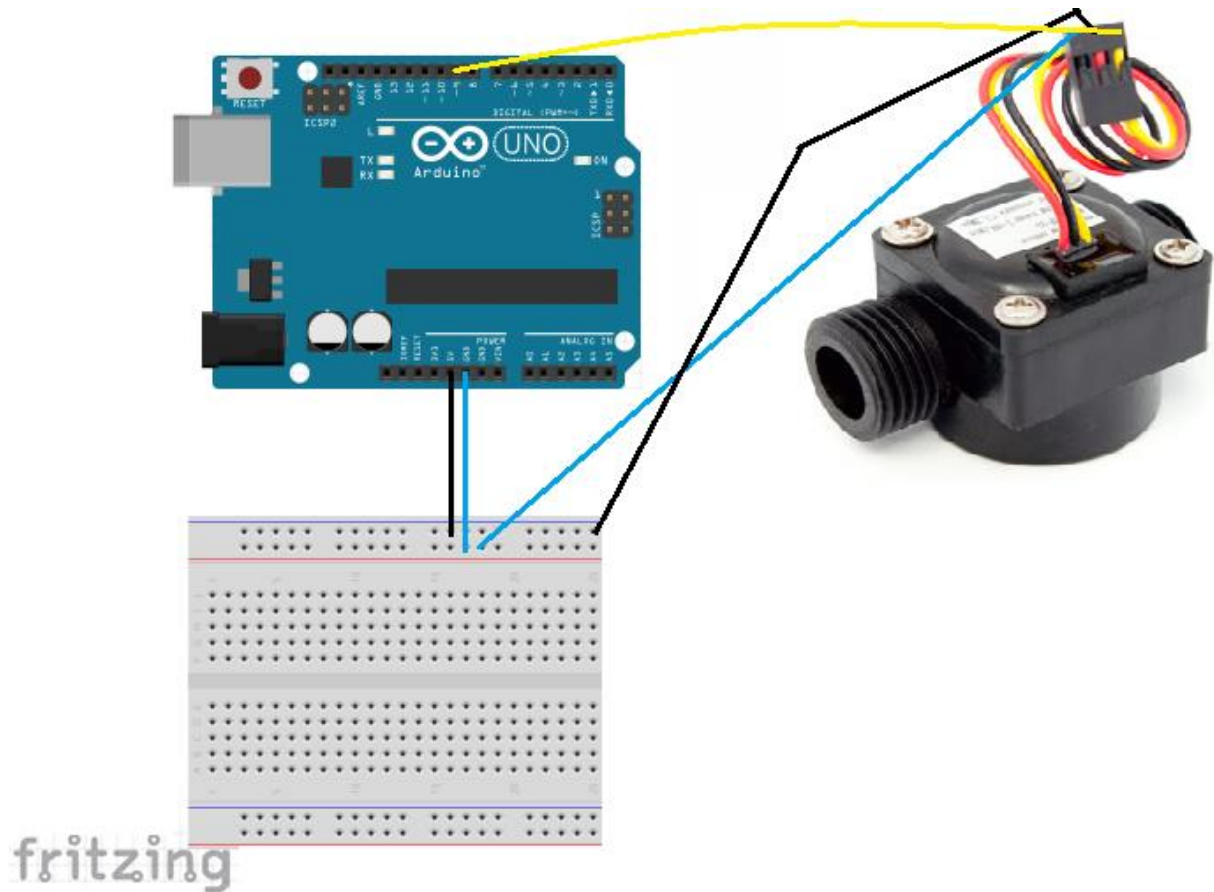


**ΕΙΚΟΝΑ 3.12:** Αισθητήρας ελέγχου ροής νερού

**Ο Προγραμματισμός:**

Συνδέουμε την τάση γείωση και βάζουμε μια ψηφιακή έξοδο.





**ΣΧΗΜΑ 3.13:** Σύνδεση αισθητήρα ελέγχου ροής νερού.

```

volatile int NbTopsFan;
int Calc;
int hallsensor =2;

Void rpm () {
NbTopsFan++;
{
pinMode(hallsensor, INPUT);
Serial.begin(9600);
attachInterrupt (0,rpm ,RISING);
}
}

void loop () {
NbTopsFan = 0;
sei ();
delay(1000);
cli();
Calc = (NbTopsFan * 60 / 7.5);
Serial.print (Calc , DEC);
Serial.print (" L/hour\r\n");
}

```

## Γ) Ηλεκτροβάνα

Με την ηλεκτροβάνα μπορούμε να καθορίσουμε πότε μπορεί να ανοίξει ή να κλείσει 0 ή 1 δηλαδή. Όμως σε περίπτωση που χρησιμοποιήσει ρελέ τότε η ηλεκτροβάνα είναι πλεονασμός καθώς μεγαλώνει το κύκλωμα και ο προγραμματισμός γίνεται πιο σύνθετος. Για να συνδεθεί η ηλεκτροβάνα χρειάζεται εξωτερική τροφοδοσία 12V. Για να γίνει ο έλεγχος της πρέπει να συνδεθεί σε ρελέ. Αναλυτικότερα συνδέουμε την γείωση στο breadboard και την τάση του NO του ρελέ και το κοινό του ρελέ το συνδέουμε επίσης στο breadboard. Από εκεί δίνουμε απευθείας τάση μέσω τροφοδοτικού συνεχής τάση 12V.

Όπως βλέπουμε στο σχήμα η ηλεκτροβάνα εσωτερικά αποτελείται από ένα ηλεκτρομαγνήτη, 2 πηνία, ένα ελατήριο καθώς και μια είσοδο ελέγχου. Δημιουργείται μαγνητικό πεδίο μέσω των πηνίων και ο μαγνήτης σπρώχνει το ελατήριο προς τα κάτω ή πάνω και έτσι αφήνεται ελεύθερο να περάσει το νερό από μέσα.



### Χαρακτηριστικά στοιχεία :

Θερμοκρασία λειτουργίας : 0-100°C

Τάση λειτουργίας : 12V

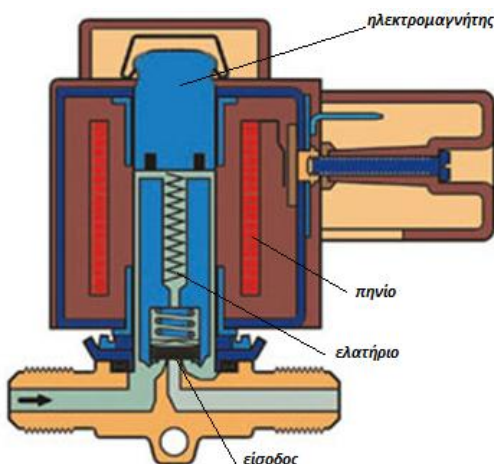
Μέγεθος σπειρώματος : 3/4"

Πίεση λειτουργίας : 0,00400m νερού

Μέγεθος εισόδου-εξόδου : 1/2

Περιβάλλον εργασίας : Νερό και λάδι

**ΕΙΚΟΝΑ 3.14:** Ηλεκτροβάνα



**ΕΙΚΟΝΑ 3.15** Εσωτερικό ηλεκτροβάνας

**ΚΕΦΑΛΑΙΟ 4:**  
**Περιγραφή συνδέσεων της εφαρμογής μας.**

## 4.1 Περιγραφή εφαρμογής

Η εφαρμογή που δημιουργήσαμε είχε στόχο να ελέγξουμε τον έλεγχο ροής του νερού και να τον στείλουμε να ποτίσει τα λουλούδια μας. Ο τρόπος που χρησιμοποιήσαμε ήταν να μετρήσουμε την υγρασία του χώματος και παράλληλα αν υπάρχει βροχή και μέσω της αντλίας νερού να στείλουμε νερό στα λουλούδια μας. Για να μπορέσει ο έλεγχος να είναι ορθός χρησιμοποιήσαμε τις πλακέτες που είχαν πάνω οι ασθητήρες μας (υγρασίας, βροχής). Ωστε να μην χρειαστεί να προσθέσουμε αντιστάσεις είτε τρανζίστορ για να μην δημιουργηθούν υψηλά ρεύματα αφού είμαστε καλυμένοι με τις πλακέτες. Χρησιμοποιήσαμε μικροελεγκτή Arduino υπο οπου προγραμματίστηκε κατάλληλα καθώς και ρελέ όπου οδήγησαμε την αντλία νερού.

## 4.2 ΑΠΑΙΤΗΣΕΙΣ ΤΟΥ ΣΥΣΤΗΜΑΤΟΣ

Η εφαρμογή που πραγματοποιήσαμε έχει ως στόχο να αντικαταστήσει την ανθρώπινη παρουσία έτσι ώστε ο έλεγχος να γίνεται μέσα από την οθόνη του υπολογιστή. Αυτό γίνεται εφικτό μέσα από την εξέλιξη της τεχνολογίας και τις δυνατότητες που δίνονται και στους χρήστες που δεν έχουν άμεση σχέση με αυτή. Έτσι, μας δίνεται η δυνατότητα να δημιουργήσουμε τις κατάλληλες συνθήκες για να γίνεται το πότισμα τη στιγμή που πρέπει καθώς και να γίνεται ο έλεγχος ροής νερού και παράλληλα η εξοικονόμισή του. Από την άλλη πλευρά, βλέπουμε πως ο χρήστης μπορεί να επεμβαίνει στα αποτελέσματα μέσα από τις συσκευές ελέγχου. Άρα, το σύστημά μας καλύπτει τις παρακάτω απαιτήσεις :

- Εύχρηστο προς τον άνθρωπο: Το σύστημα που υλοποιήσαμε έχει σχεδιαστεί με σκοπό να είναι εύκολος ο προγραμματισμός του. Αυτό σημαίνει ότι μπορεί να το διαχειριστεί και κάποιος άπειρος χρήστης που δεν έχει σχέση με το αντικείμενο.
- Επίβλεψη των συσκευών ελέγχου: Ο χρήστης έχει την δυνατότητα να ελέγχει αυτές τις συσκευές καθώς επίσης μέσα από τον προγραμματισμό τους μπορεί είτε να τις ενεργοποιεί είτε να τις απενεργοποιεί.
- Αντικατάσταση του ανθρώπινου παράγοντα: Η συγκεκριμένη εφαρμογή έχει ως κύριο στόχο να απαλλάξει τον άνθρωπο από επιπλέον προβλήματα. Αυτό σημαίνει πως δεν χρειάζεται η παρουσία του έτσι ώστε να κάνει το πότισμα των λουλουδιών του ή ακόμα και του χωραφιού του. Η γρήγορη ανάπτυξη της τεχνολογίας μας έδωσε την δυνατότητα να μπορούμε να ελέγξουμε τα πάντα μέσα από τον ηλεκτρονικό υπολογιστή.

### 4.3 ΚΟΣΤΟΣ ΥΛΙΚΩΝ- ΚΑΤΑΣΚΕΥΗ ΜΑΚΕΤΑΣ

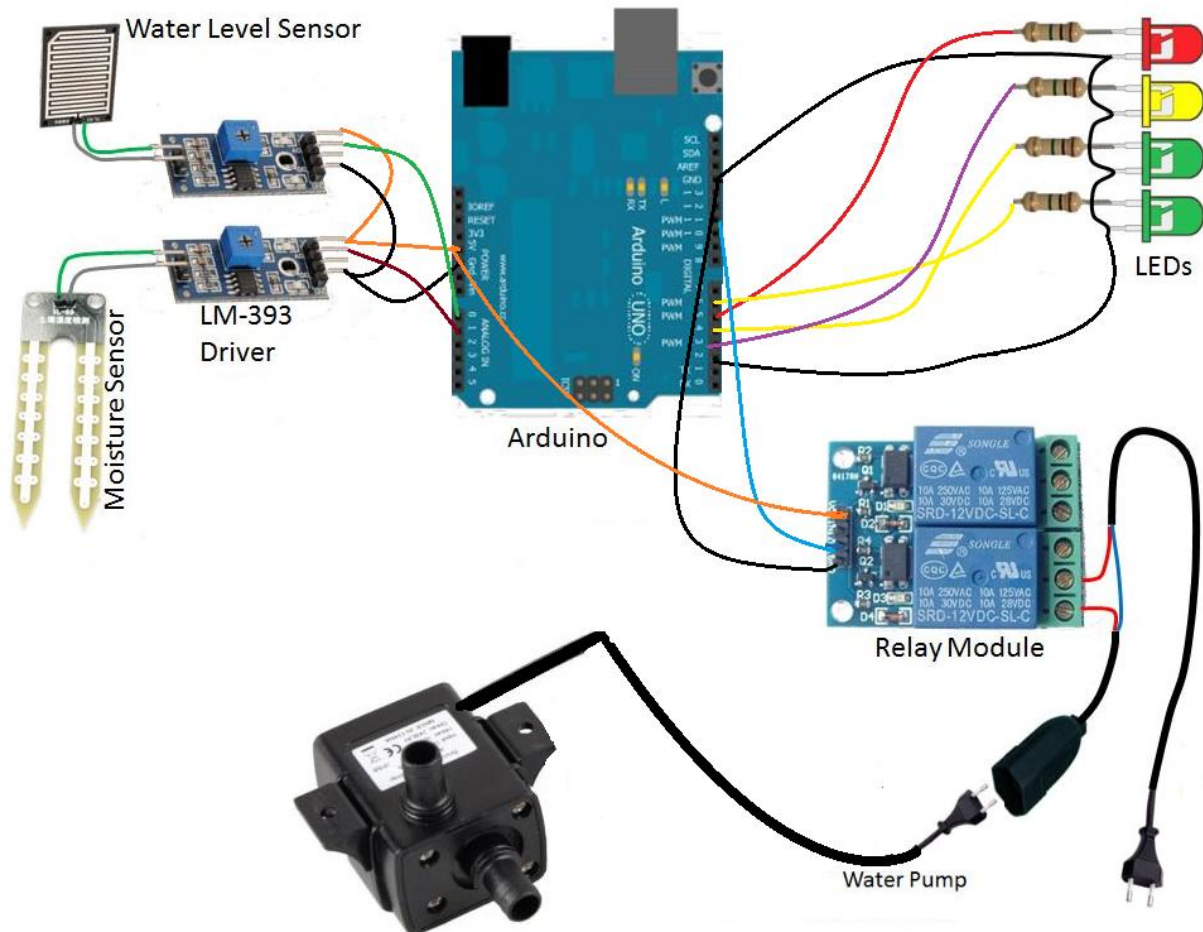
ΠΙΝΑΚΑΣ 4.1 : Υλικά και κόστος

ΥΛΙΚΑ	ΚΟΣΤΟΣ
Mj arduino uno r3	5e
Breadboard	2e
Αισθητήρας χρώματος	1e
Αισθητήρας βροχής	1,5e
Ηλεκτροβάνα	5e
Αντλία νερού	4.8e
Αισθητήρας ελέγχου ροής	12e
Μετασχηματιστής 220Vac σε 12Vdc	10e
Καλωδια-led-ασφάλειες-αντιστάσεις	9e
Διάφορα άλλα υλικά	3e
1 κανάλι ρελέ	1e
2 κανάλια ρελέ	3e
Υλικά μακέτας	12e
Εργατώρες	40ώρες
Συνολικό κόστος	69,3e

### 4.4 ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΚΑΙ ΣΧΕΔΙΑ ΓΙΑ ΤΗΝ ΥΛΟΠΟΙΗΣΗ ΤΟΥ ΣΥΣΤΗΜΑΤΟΣ

**ΣΧΗΜΑ 4.2:**  
Σύνδεση συστήματος και υλοποίησή του.





#### 4.5 ΜΕΛΜΟΝΤΙΚΕΣ ΑΛΛΑΓΕΣ ΤΟΥ ΣΥΣΤΗΜΑΤΟΣ ΜΑΣ

Το παρών σύστημα μπορεί να βελτιωθεί αισθητά. Καταρχήν, θα πρέπει να γίνεται όσο το δυνατόν ευκολότερο στην χρήση από τον άνθρωπο. Αυτό θα γίνει μέσω εφαρμογής κινητού τηλεφώνου. Η εφαρμογή θα έχει στόχο να μπορεί ο χρήστης να βλέπει τα φυτά του καθώς και το νερό που υπάρχει στο σύστημα του. Πιο συγκεκριμένα :

- Μέσω GSM shield μπορούμε να έχουμε επικοινωνία με την εφαρμογή στέλνοντας μας κάποιο μήνυμα είτε κάποιο email.
- Μέσω wifi shield εφόσον το σύστημα μας είναι στο μπαλκόνι είτε στον κήπο μας και υπάρχει ασύρματο δίκτυο μπορεί να συνδεθεί εύκολα.

- Μπορούμε να εγκαταστήσουμε μία οθόνη LCD και να δείχνει σε πραγματικό χρόνο τα αποτελέσματα από το σύστημα μας.
- Μπορούμε να τοποθετήσουμε ένα buzzer να μας ειδοποιεί εάν η υγρασία του χώματος είναι χαμηλή ακόμα και αν η στάθμη του νερού είναι χαμηλή.
- Ωστόσο μπορούμε να προσθέσουμε μια δεύτερη αντλία και έναν αισθητήρα υπερήχων ώστε να μετράει την δεξαμενή και να στέλνει νερό μέσω μιας ηλεκτροβάνας σε αυτήν όταν δεν έχει νερό.
- Μέσω ανανεώσιμων πηγών ενέργειας μπορούμε να τροφοδοτήσουμε πλήρως το σύστημα μας ώστε να μην χρειάζεται εξωτερική τροφοδοσία.
- Μπορούμε να εγκαταστήσουμε κάμερα από την οποία θα βλέπουμε σε πραγματικό χρόνο την ανάπτυξη του φυτού καθώς και η δυνατότητα της αποθήκευσης του ιστορικού του.
- Τέλος, μέσω πολυπλέκτη μπορούμε να εγκαταστήσουμε περισσότερους αισθητήρες υγρασίας χώματος καθώς και αισθητήρες βροχής ώστε να έχουμε τον απόλυτο έλεγχο των φυτών ώστε να γίνεται η εξοικονόμηση νερού και χρόνου.

#### 4.6 ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΕΦΑΡΜΟΓΗΣ

```
//
#####
#
#   Arxeio
#       Automated control system for flow
#       water based on module Arduino..ino
#   Plaketa:      Arduino UNO
#   Glwsses:     Wiring/C/Processing/Fritzing/Arduino
#
#   Antikeimeno:  Na potisoume !
#
#
#
#
#   Programmatistes:  Psyllas Dimititrios-Athanasios
#                   Dimitriou-Kommata Kwnstantina
#   Hmeromhnia:     24/02/16
#   Topos:          Patras ,Greece
#
#####
//
```

```

volatile int NbTopsFan;
int Calc;
int hallsensor = 2;

void rpm ()
{
  NbTopsFan++;
}
#define Relay1 10
void setup()
{Serial.begin(9600);
  pinMode(11,OUTPUT);
  pinMode(Relay1 ,OUTPUT);
  pinMode(hallsensor, INPUT);
  Serial.begin(9600);
  attachInterrupt(0, rpm, RISING);
}

void loop ()
{
  NbTopsFan = 0;
  sei();
  delay (1000);
  cli();
  Calc = (NbTopsFan * 60 / 7.5);

  Serial.print (Calc, DEC);
  Serial.print (" L/hour\r\n");
int a=analogRead(A0);//diavazei ygrasia
int b=analogRead(A1);// diavazei broxh
Serial.println("Soil moisture value= "); //tupwnei thn ugrasia
Serial.println(a);
delay(300);
Serial.println("rain intensity= "); // tupwnei thn vroxi
Serial.println(b);
delay(300);
if(a>500&&a<1023)
{digitalWrite(Relay1,0); // anoigei vana
} else
{digitalWrite(Relay1,1);} // kleinei vana
if(b>500)
{ digitalWrite(11,1); }
else
{ digitalWrite(11,0);}
}

```



## 4.7 ΣΥΜΠΕΡΑΣΜΑΤΑ

Η πτυχιακή εργασία που εκπονήθηκε στις παραπάνω παραγράφους, υλοποίησε ένα πρότυπο ενσωματωμένο σύστημα παρακολούθησης ελέγχου ροής νερού με αυτόματο πότισμα. Βάση αναγκών των απλών χρηστών αλλά και των διαχειριστών αναπτύχθηκαν και οι αντίστοιχες άνωθεν λειτουργίες.

Ο χρήστης έχει στόχο να ποτίσει τα φυτά του , να κάνει έλεγχο ροής του νερού καθώς και την εξοικονόμησή του. Αυτό έχει σαν αποτέλεσμα ο χρήστης να ποτίζει τα λουλούδια όταν πρέπει και όχι συνέχεια. Θέτοντας τις παραμέτρους μπορεί εύκολα ο χρήστης να έχει μια άποψη για τα λουλούδια αν είναι ποτισμένα ή όχι.

Η δημιουργία του συστήματος αυτού , έγινε με σκοπό την πλήρη αντικατάσταση της ανθρώπινης παρουσίας όσο είναι δυνατόν καθώς χρειάζεται και η ανθρώπινη παρουσία. Απώτερος σκοπός της εργασίας αυτής είναι ο χρήστης να μπορεί να ελέγχει τα φυτά του με καλύτερη επίβλεψη μειώνοντας τον χρόνο που απαιτείται να ποτίσει, τη φιλικότητα προς το περιβάλλον αλλά και στην εξοικονόμηση του νερού. Τέλος αν επιτευχθούν και οι μελλοντικές αλλαγές τότε η παρουσία του χρήστη θα του επιτρέπει να τα ελέγχει από το καναπέ του σπιτιού του είτε γυρνώντας από την δουλεία του!

## **ΒΙΒΛΙΟΓΡΑΦΙΑ**

### **ΔΙΑΔΥΚΤΙΟ:**

<https://www.arduino.cc/> (ανάκτηση Σεπτέμβριος 2015)  
<https://el.wikipedia.org/wiki/Arduino> (ανάκτηση Σεπτέμβριος 2015)  
<http://www.ladyada.net/learn/arduino/> (ανάκτηση Σεπτέμβριος 2015)  
<https://www.arduino.cc/en/Main/Software> (ανάκτηση Ιούνιος 2015) Arduino IDE  
<http://fritzing.org/home/> (ανάκτηση Σεπτέμβριος 2015) fritzing  
[http://arduinolab.weebly.com/pirhoomicrongammarhoalphamumualphatauiotasigmamuomicr  
onsigmaf-arduino.html](http://arduinolab.weebly.com/pirhoomicrongammarhoalphamumualphatauiotasigmamuomicr<br/>onsigmaf-arduino.html) (ανάκτηση Σεπτέμβριος 2015)  
<https://deltahacker.gr/arduino-intro/> (ανάκτηση Σεπτέμβριος 2015)

### **ΒΙΒΛΙΑ:**

Συστήματα μικροϋπολογιστών (τόμος 1) Μικροεπεξεργαστές 80x86 pentium και AMR  
Πεκμεστζή , Εκδόσεις Συμμετρία 2009  
Συστήματα μικροϋπολογιστών (τόμος 2) Μικροελεγκτές AVR και PIC , Πεκμεστζή , Εκδόσεις  
Συμμετρία 2009.  
Ανάπτυξη εφαρμογών με το Arduino , Παπάζογλου, Λιώνης εκδόσεις Τζιόλα 2015.  
Ηλεκτρικές μετρήσεις και αισθητήρες , Καλαιτζάκης, Κουτρούλης εκδόσεις Κλειδάριθμος  
2010