

Γενικό Λύκειο (ΓΕ.Λ.)

Α' Λυκείου

«Εφαρμογές Πληροφορικής»



[Πηγή](#)

Ανδρέας Πλαγεράς

Περιεχόμενα μαθήματος

1. Ενότητα 1: Υλικό-Λογισμικό και Εφαρμογές

- Κεφάλαιο 1: Υλικό Υπολογιστών
- Κεφάλαιο 2: Λογισμικό
- Κεφάλαιο 3: Εφαρμογές Υπολογιστών Άνθρωπος
- Κεφάλαιο 4: Κοινωνικές Επιπτώσεις

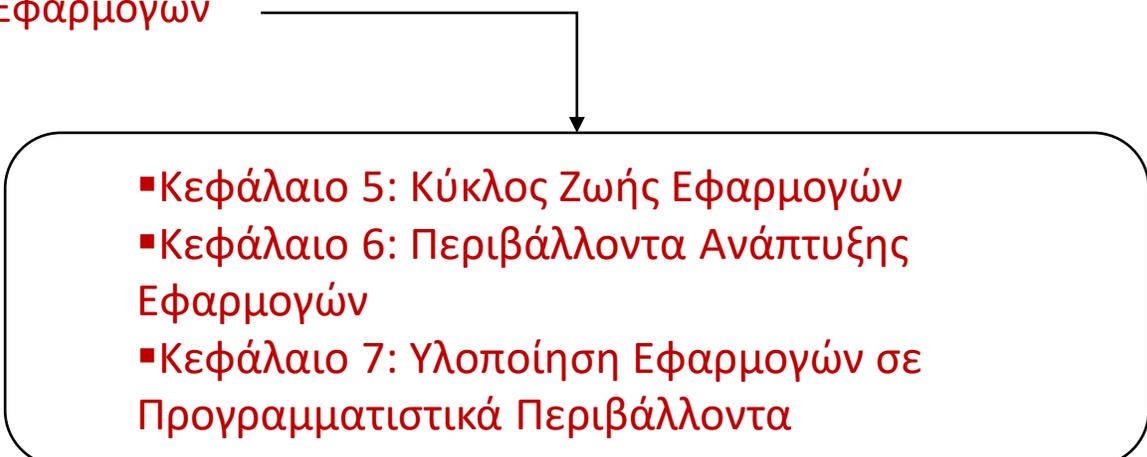
2. Ενότητα 2: Προγραμματιστικά Περιβάλλοντα – Δημιουργία Εφαρμογών

3. Ενότητα 3: Επικοινωνία και Διαδίκτυο

- Κεφάλαιο 8: Δίκτυα Υπολογιστών
- Κεφάλαιο 9: Διαδίκτυο, Web 2.0 και Web X.0.
- Κεφάλαιο 10: Υπηρεσίες και Εφαρμογές Διαδικτύου
- Κεφάλαιο 11: Εισαγωγή στην HTML
- Κεφάλαιο 12: Η Μάθηση στο Διαδίκτυο

4. Ενότητα 4: Συνεργασία και Ασφάλεια στο Διαδίκτυο

- Κεφάλαιο 13: Εφαρμογές Νέφους
- Κεφάλαιο 14: Τηλεργασία – Ασύγχρονη και Σύγχρονη Συνεργασία από Απόσταση
- Κεφάλαιο 15: Κοινωνικά Δίκτυα
- Κεφάλαιο 16: Ασφάλεια και Προστασία στο Διαδίκτυο

- 
- Κεφάλαιο 5: Κύκλος Ζωής Εφαρμογών
 - Κεφάλαιο 6: Περιβάλλοντα Ανάπτυξης Εφαρμογών
 - Κεφάλαιο 7: Υλοποίηση Εφαρμογών σε Προγραμματιστικά Περιβάλλοντα

Ανάπτυξη Εφαρμογών και Κύκλος Ζωής Εφαρμογών



Τι κοινό έχουν όλες οι γλώσσες?

Μία γλώσσα προσδιορίζεται από το αλφάβητο της, το λεξιλόγιο της, τη γραμματική της και τέλος, τη σημασιολογία της.

- **Αλφάβητο:**
 - ✓ είναι το **σύνολο των χαρακτήρων** που χρησιμοποιούνται από τη γλώσσα
- **Λεξιλόγιο:**
 - ✓ είναι το **σύνολο των λέξεων** που αναγνωρίζει η γλώσσα και έχουν συγκεκριμένη και μοναδική σημασία
 - ✓ Στις γλώσσες προγραμματισμού το λεξιλόγιο είναι πολύ περιορισμένο (μερικές δεκάδες λέξεις), ώστε να μπορούμε να το μάθουμε εύκολα.
- **Συντακτικό:**
 - ✓ είναι το **σύνολο των κανόνων** που πρέπει να ακολουθούμε, για να συνδέουμε λέξεις σε προτάσεις
 - ✓ Σε μια γλώσσα προγραμματισμού η σύνδεση λέξεων δημιουργεί ολοκληρωμένες εντολές προς τον υπολογιστή.
 - ✓ Αν δεν ακολουθήσουμε αυστηρά το συντακτικό μιας γλώσσας, είναι αδύνατο για τον υπολογιστή να καταλάβει ποια εντολή του δίνουμε.
- **Σημασιολογία:**
 - ✓ είναι η **σημασία** των λέξεων και προτάσεων.

Περιβάλλοντα Ανάπτυξης Εφαρμογών (2)

- Κάθε υπολογιστής μπορεί να κατανοήσει και να εκτελέσει **εντολές** που είναι διατυπωμένες με έναν καθορισμένο τρόπο, ο οποίος έχει σχέση με τον επεξεργαστή του.
- Οι γλώσσες προγραμματισμού, στις οποίες γράφονται οι εντολές αυτές, ονομάζονται **γλώσσες μηχανής**, και αποτελούνται από μια ακολουθία **δυναδικών ψηφίων (0 και 1)**.
- Για τη διευκόλυνση της εργασίας του προγραμματισμού δημιουργήθηκαν οι **συμβολικές γλώσσες ή γλώσσες χαμηλού επιπέδου**, όπου οι εντολές είναι συντομογραφίες λέξεων της Αγγλικής γλώσσας.
- Υπάρχει εξάρτηση μεταξύ γλώσσας και αρχιτεκτονικής υπολογιστή.
- Ένα πρόγραμμα γραμμένο σε συμβολική γλώσσα, για να εκτελεστεί από έναν υπολογιστή, πρέπει να μεταφραστεί στη γλώσσα μηχανής του με ένα ειδικό πρόγραμμα που ονομάζεται **συμβολομεταφραστής**.

Γλώσσες Προγραμματισμού

- Φυσικές γλώσσες
- Γλώσσες Προγραμματισμού
 - Γλώσσες Μηχανής
 - Γλώσσες Χαμηλού Επιπέδου ή Συμβολικές
 - Γλώσσες Υψηλού Επιπέδου

```
10101000 00001010
10001100 00000001
00111100
01010001 00000001
01000011 00000001
11000000 11111010
10001100 00000000
11111111
```

```
INDEX=$01
SUM=$02
LDA #10
STA INDEX
CLA
LOOP ADD INDEX
      DEC INDEX
      BNE LOOP
STA SUM
BRK
```

```
sum = 0
FOR index = 1 TO 10
    sum = sum + index
NEXT index
END
```


Γλώσσα Μηχανής

- Ο υπολογιστής είναι μια ψηφιακή μηχανή που «αντιλαμβάνεται» μόνο δύο καταστάσεις (περνάει ή δεν περνάει ρεύμα), στις οποίες αντιστοιχούν τα δυαδικά ψηφία **0** και **1**.
- **Γλώσσα μηχανής** είναι η γλώσσα που χρησιμοποιεί ως **αλφάβητο** μόνο το **0** και το **1**.
- Δηλαδή στη γλώσσα μηχανής όλες οι εντολές, τα δεδομένα και τα αποτελέσματα εκφράζονται ως μια ακολουθία από 0 και 1, με βάση κάποιο κώδικα αντιστοιχίας (π.χ. [κωδικοποίηση ASCII](#)).

[πηγή ΒΙΚΙΠΑΙΔΕΙΑ]

- «**Η εκμάθηση της γλώσσας μηχανής είναι αρκετά δύσκολη**».
- Η γλώσσα μηχανής δεν είναι ίδια σε όλους τους υπολογιστές, αλλά εξαρτάται από τον τύπο (αρχιτεκτονική) του υπολογιστή.

Συστήματα Αρίθμησης

Δεκαδικό	Δυαδικό	Οκταδικό	Δεκαεξαδικό
0	00	000	0000
1	01	001	0001
2	10	010	0010
3	11	011	0011
4	100	100	0100
5	101	101	0101
6	110	110	0110
7	111	111	0111
8	1000		1000
9	1001		1001
10	1010		A (1010)
11	1011		B (1011)
12	1100		C (1100)
13	1101		D (1101)
14	1110		E (1110)
15	1111		F (1111)

Μονάδες Μέτρησης της Μνήμης του Η/Υ

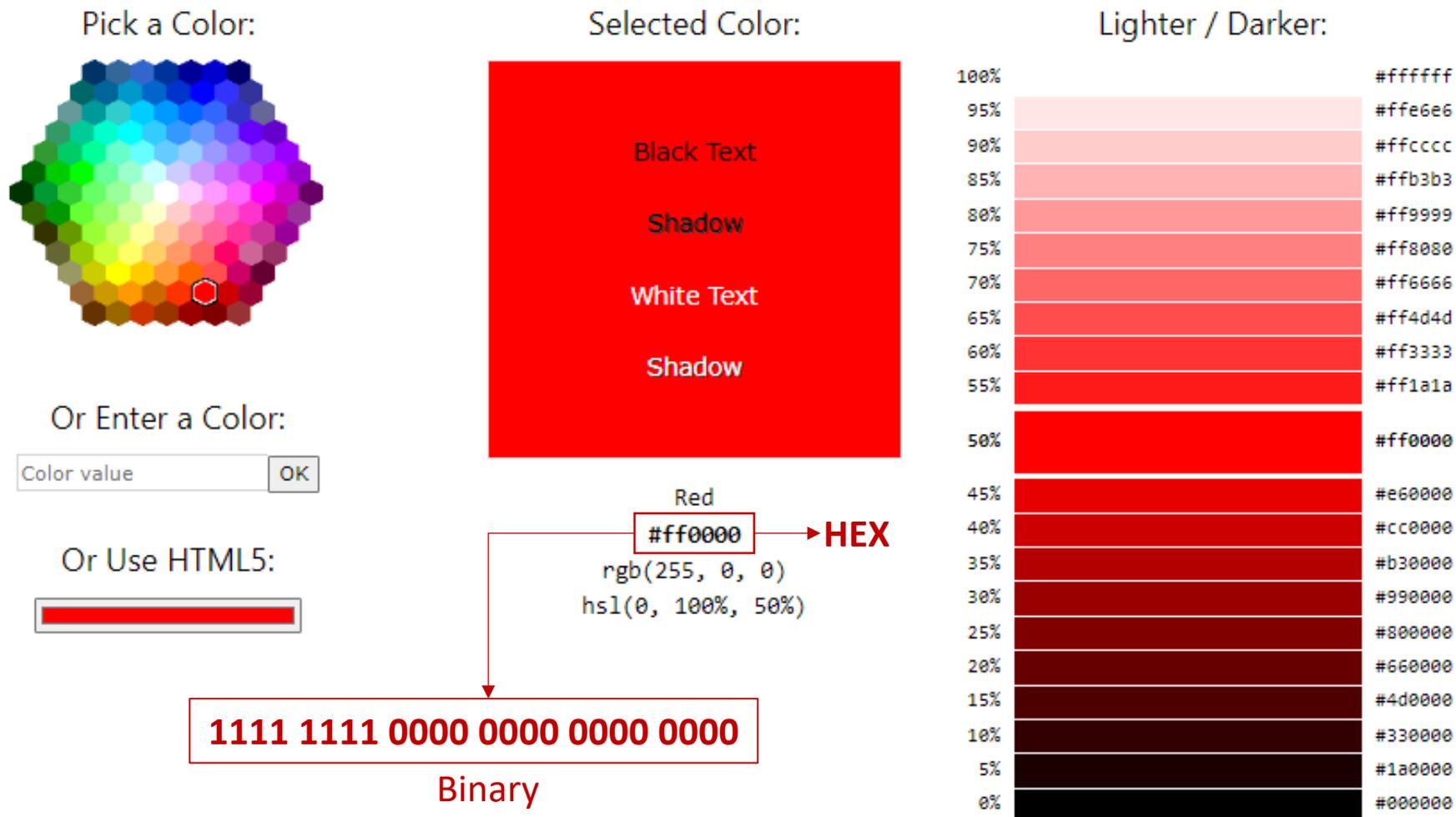
1 Bit	= Binary Digit
8 Bits	= 1 Byte
1024 Bytes	= 1 KB [Kilo Byte]
1024 KB	= 1 MB [Mega Byte]
1024 MB	= 1 GB [Giga Byte]
1024 GB	= 1 TB [Terra Byte]
1024 TB	= 1 PB [Peta Byte]
1024 PB	= 1 EB [Exa Byte]
1024 EB	= 1 ZB [Zetta Byte]
1024 ZB	= 1 YB [Yotta Byte]
1024 YB	= 1 Bronto Byte
1024 Brontobyte	= 1 Geop Byte

Geop Byte is the Highest Memory.

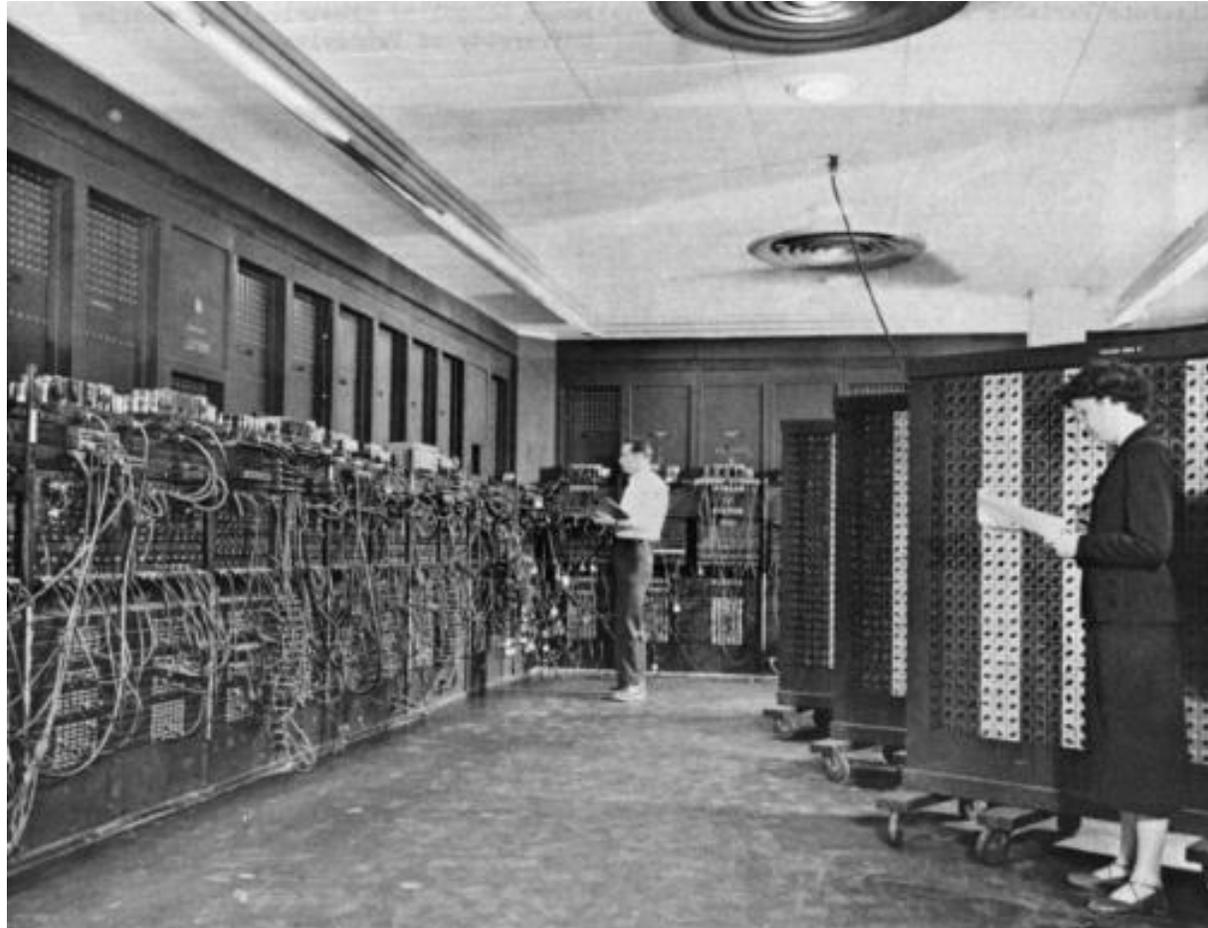
[Google Data Center](#)

Η αντίληψη του υπολογιστή

Παράδειγμα:



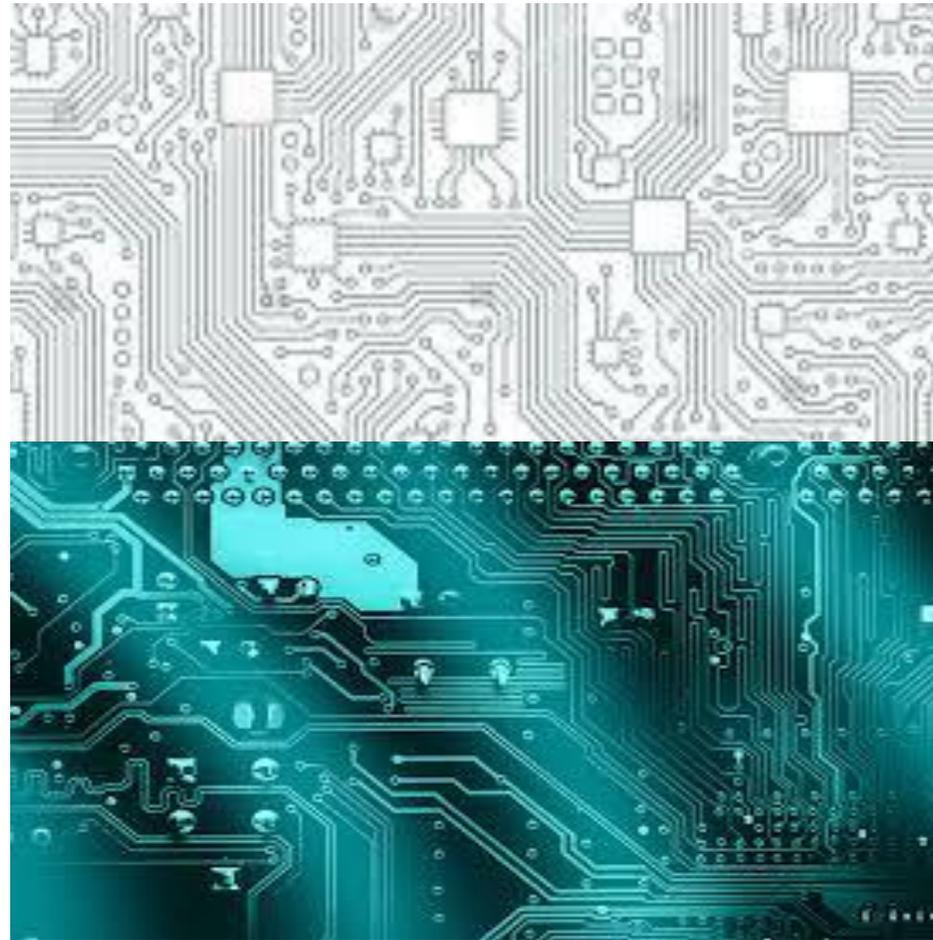
1946: ENIAC



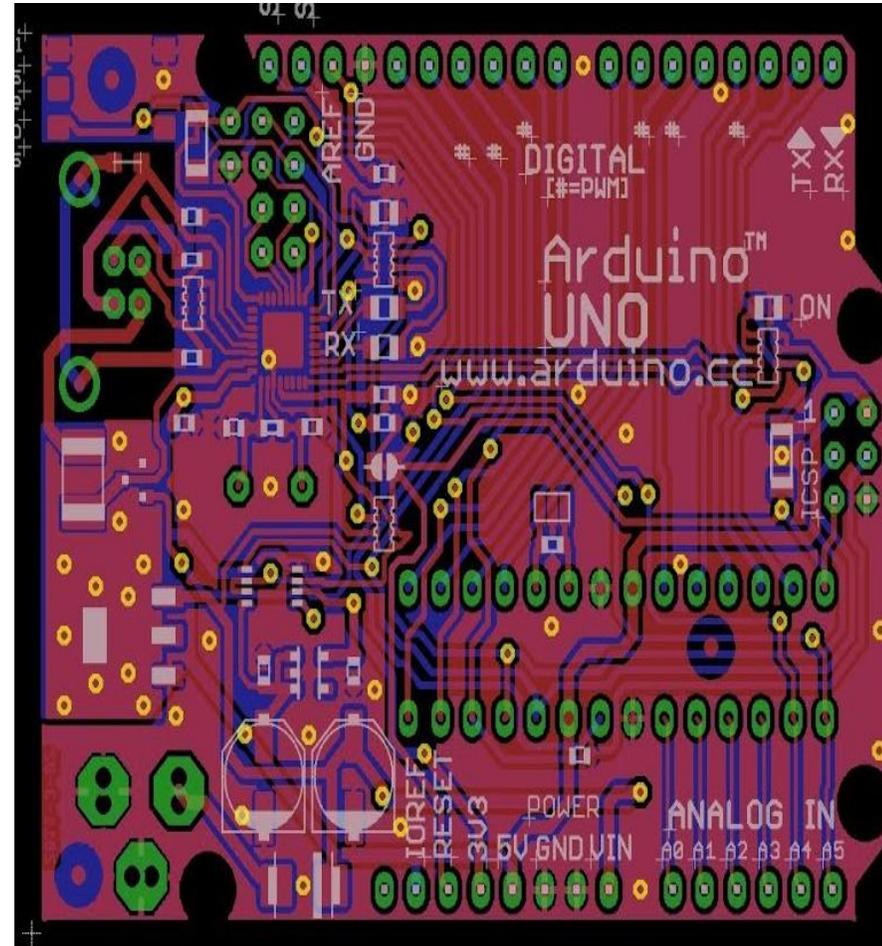
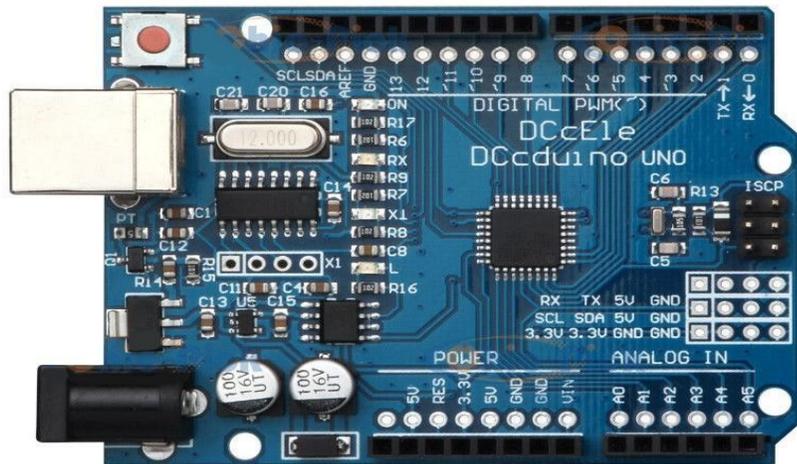
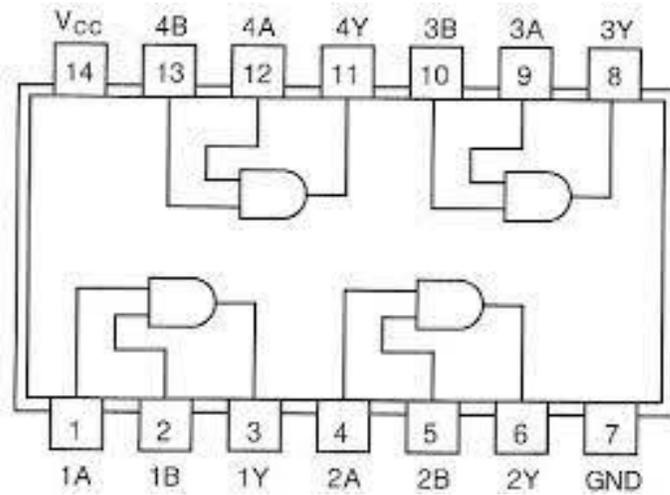
Λογικές Πύλες

Λογική Πύλη	Είσοδοι	Έξοδος	Συνάρτηση	Λογικό Διάγραμμα
Απομονωτής Buffer	A	Y	$Y=A$	 $Y=A$
Αντιστροφέας NOT	A	Y	$Y=\bar{A}$	 $Y=\bar{A}$
AND	A,B	Y	$Y=A \cdot B$	 $Y=A \cdot B$
OR	A,B	Y	$Y=A+B$	 $Y=A+B$
NAND	A,B	Y	$Y=\overline{A \cdot B}$	 $Y=\overline{A \cdot B}$
NOR	A,B	Y	$Y=\overline{A+B}$	 $Y=\overline{A+B}$
XOR	A,B	Y	$Y=A \cdot \bar{B} + \bar{A} \cdot B = A \oplus B$	 $Y=A \oplus B$
XNOR	A,B	Y	$Y=A \cdot B + \bar{A} \cdot \bar{B} = \overline{A \oplus B} = A \odot B$	 $Y=A \odot B$

Μητρική Πλακέτα και Λογικές Πύλες

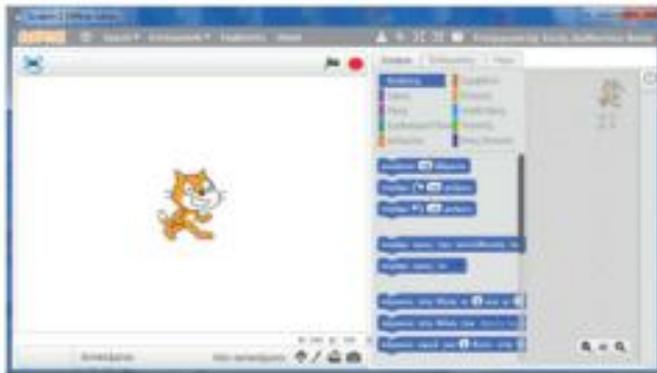


Arduino - Λογικές πύλες - Λογικά κυκλώματα



Ανάπτυξη Εφαρμογών

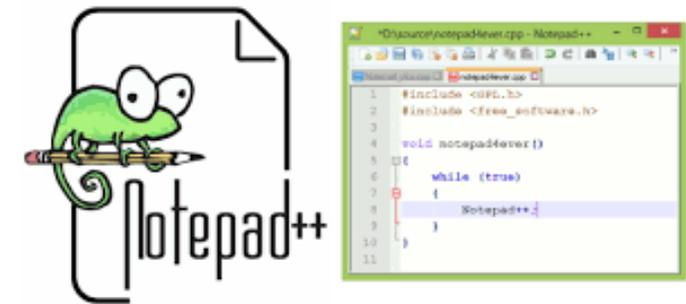
- Εργαλείο Σύνταξης Κώδικα (Code Editor)
- Μεταγλωττιστή (Compiler) και Διερμηνέας ή Συμβολομεταφραστή (Interpreter)
- Εργαλείο Εκσφαλμάτωσης (Debugger)
- Ενσωματωμένο Περιβάλλον Ανάπτυξης (Integrated Development Environment – IDE)



Scratch



Visual Studio Code



Notepad++

Στάδια Ανάπτυξης Εφαρμογών

1. Ανάλυση
2. Σχεδίαση
3. Υλοποίηση
4. Λειτουργία
5. Αξιολόγηση
6. Συντήρηση



Στάδια Ανάπτυξης Εφαρμογών

- Κατά την **ανάλυση** καθορίζουμε με ακρίβεια και σαφήνεια το τι θα κάνει το πρόγραμμά μας.
- Στη φάση της **σχεδίασης** καθορίζουμε πώς θα εκτελούνται οι οδηγίες του προγράμματος με λεπτομέρεια.
- Κατά την τρίτη φάση **υλοποιούμε** το πρόγραμμά μας. Μετατρέπουμε δηλαδή τα βήματα που περιγράψαμε στη φάση δύο σε πρόγραμμα, με χρήση μιας γλώσσας προγραμματισμού.
- Στην τέταρτη φάση, αυτή της **λειτουργίας**, ελέγχουμε εάν το πρόγραμμά μας κάνει αυτό που θέλαμε να κάνει. Αν υπάρχουν λάθη στον τρόπο που γράψαμε τις εντολές, τα εντοπίζουμε και τα διορθώνουμε.
- Κατά τη φάση της **αξιολόγησης**, αξιολογούμε το τελικό αποτέλεσμα της δουλειάς μας και αποφαινόμεστε εάν θα κρατήσουμε το πρόγραμμα ή θα το απορρίψουμε.
- Στην τελευταία φάση, αυτή της **συντήρησης**, περιλαμβάνεται η σωστή διατήρηση και ενημέρωση του προγράμματός μας σε βάθος χρόνου.

Ευχαριστώ για την προσοχή σας!

