



**Α.Τ.Ε.Ι. Θεσσαλονίκης**  
**Τμήμα Ηλεκτρονικών Μηχανικών**

**ΑΛΕΞΑΝΔΡΕΙΟ ΤΕΧΝΟΛΟΓΙΚΟ-ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ ΘΕΣΣΑΛΟΝΙΚΗΣ**  
**ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ**  
**ΤΜΗΜΑ ΗΛΕΚΤΡΟΝΙΚΩΝ ΜΗΧΑΝΙΚΩΝ ΤΕ**

**ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ**

**Υλοποίηση διαδραστικού κύβου φωτοδιόδων 4x4x4**

**Interactive 3D LED CUBE**



**ΤΣΟΜΠΑΝΙΚΟΛΙΔΗΣ ΧΑΡΑΛΑΜΠΟΣ**

**ΚΑΣ:513180**

**«17152M»**

**Επιβλέπων Καθηγητής:**  
**Παπακώστας Δημήτριος**

**Θεσσαλονίκη, Μάιος 2018**



## Πρόλογος

Η παρούσα Πτυχιακή εργασία με τίτλο «Κατασκευή διαδραστικού ηλεκτρονικού κύβου φωτοδιόδων» υλοποιήθηκε στα πλαίσια της ολοκλήρωσης των υποχρεώσεων, για την λήψη του πτυχίου μου από το Α.Τ.Ε.Ι. Θεσσαλονίκης, Σχολή Τεχνολογικών Εφαρμογών, Τμήμα Ηλεκτρονικών Μηχανικών. Η ανάθεση της έγινε στις 16 Οκτώβριου του 2017 και ολοκληρώθηκε στις 10 Μαΐου του 2018. Θα ήθελα να εκφράσω τις ευχαριστίες μου στον επιβλέποντα καθηγητή μου, κ. Παπακώστα Δημήτριο, που συνέβαλε στη βελτίωση της εργασίας.

Στα πλαίσια αυτής της πτυχιακής εργασίας πραγματοποιήθηκε η υλοποίηση ενός ηλεκτρονικού κύβου φωτοδιόδων (led), ο οποίος ρυθμίστηκε ώστε να συλλέγει πληροφορίες από έναν αισθητήρα θερμοκρασίας, λειτουργώντας ουσιαστικά ως μια πρωτότυπη μορφή θερμομέτρου. Αυτή τη στιγμή, υπάρχει μεγάλη ποικιλία κύβων φωτοδιόδων για διαφημιστικούς σκοπούς, όμως ελάχιστες είναι οι περιπτώσεις, όπου χρησιμοποιούνται σε συνδυασμό με αισθητήρες.



# Περιεχόμενα

Περίληψη.....	7
Abstract.....	8
Κεφάλαιο 1° Εισαγωγή.....	9
1.1 Εισαγωγή.....	10
1.2 Κύβος φωτοδιόδων: Τι είναι.....	10
1.3 Κύβος φωτοδιόδων: Λειτουργία.....	10
1.4 Δίοδος εκπομπής φωτός.....	10
1.5 Ανατομία και λειτουργία κύβου LED.....	11
1.6 Αισθητήρας Θερμοκρασίας.....	15
1.7 Τα σημαντικότερα στοιχεία της κατασκευής.....	15
Κεφάλαιο 2° Υλικά κατασκευής.....	17
2.1 Μικροελεγκτής PIC18F4550.....	18
2.2 Αναλογικός αισθητήρας θερμοκρασίας ακριβείας LM35.....	19
2.3 Δίοδοι εκπομπής φωτός-LED.....	20
2.4 Block διάγραμμα κατασκευής.....	22
Κεφάλαιο 3° Κατασκευή.....	25
3.1 Κατασκευή του κύβου φωτοδιόδων.....	26
3.2 Σχηματικό διάγραμμα κύβου.....	30
3.3 Προγραμματισμός μικροελεγκτή.....	31
3.4 Τελική κατασκευή κύβου σε πλακέτα.....	32
Κεφάλαιο 4° Λογισμικό.....	35
4.1 Τι είναι ο μικροελεγκτής.....	36
4.2 Στάδιο πρώτο: Σύνθεση κώδικα και μετατροπή σε γλώσσα μηχανής.....	36
4.3 Στάδιο δεύτερο: Τοποθέτηση κώδικα στη μνήμη του μικροελεγκτή.....	37
4.4 Διάγραμμα ροής κύβου.....	38
4.5 Επεξήγηση ADC (Analog to Digital Converter).....	43
Κεφάλαιο 5° Συμπεράσματα.....	45
5.1 Συμπεράσματα.....	46
5.2 Παρόμοιες κατασκευές LED κύβων.....	46
Παραρτήματα.....	51
Παράρτημα Α: Υλικά κατασκευής.....	52

<b>Παράρτημα Β: Κώδικας.....</b>	<b>53</b>
<b>Αναφορές.....</b>	<b>71</b>

## Περίληψη

Η παρούσα πτυχιακή εργασία στοχεύει στην υλοποίηση ενός διαδραστικού-ηλεκτρονικού κύβου φωτοδιόδων 4x4x4. Οι φωτοδιόδοι έχουν κολληθεί έτσι ώστε να έχουν το σχήμα ενός κύβου, σε τρισδιάστατη μορφή. Οι εξήντα τέσσερις φωτοδιόδοι ελέγχονται από έναν μικροελεγκτή, ο οποίος με την σειρά του συλλέγει δεδομένα από έναν αισθητήρα θερμοκρασίας.

Αρχικά, επιλέχτηκε ο συγκεκριμένος μικροελεγκτής, προκειμένου να καλύψει τις ανάγκες αυτής της εργασίας. Έπειτα ο αισθητήρας θερμοκρασίας και οι φωτοδιόδοι, που μαζί με τα κατάλληλα υλικά, δημιούργησαν την παρούσα κατασκευή. Στη συνέχεια, σχεδιάστηκε και κατασκευάστηκε το ηλεκτρονικό κύκλωμα. Τέλος, η ανάπτυξη του λογισμικού σε γλώσσα προγραμματισμού C, ολοκλήρωσε την κατασκευή, δίνοντας έτσι επικοινωνία στα σημαντικότερα εξαρτήματα (μικροελεγκτής, θερμοστάτης, κύβος φωτοδιόδων).

Η εργασία αυτή, διαφοροποιείται από μία απλή κατασκευή ενός κύβου απεικόνισης, καθώς χρησιμοποιεί πραγματικά δεδομένα θερμοκρασίας για την λειτουργία του, συνδυάζοντας έτσι και την εμφάνιση αλλά και την πρακτικότητα.

Συνοπτικά, το πρώτο κεφάλαιο ξεκινά με μια εισαγωγή στις έννοιες και στον τρόπο λειτουργίας του ηλεκτρονικού κύβου. Στη συνέχεια ακολουθεί το δεύτερο κεφάλαιο το οποίο αναφέρεται στα βασικά υλικά της εργασίας. Στο τρίτο κεφάλαιο παρατίθεται το σχηματικό διάγραμμα του κύβου που πραγματοποιήθηκε σε πρόγραμμα προσομοίωσης ηλεκτρονικών εξαρτημάτων και γίνεται ανάλυση των περιεχομένων του, καθώς επίσης και η διαδικασία υλοποίησης του. Το επόμενο κεφάλαιο, αφορά τον προγραμματισμό της κατασκευής, το λογισμικό που χρησιμοποιήθηκε καθώς επίσης και επεξήγηση λειτουργίας ADC(Analog-Digital Converter) των μικροελεγκτών. Στο τελευταίο κεφάλαιο βρίσκονται τα συμπεράσματα και ορισμένα παραδείγματα που αποτέλεσαν πηγή έμπνευσης για την συγκεκριμένη πτυχιακή εργασία. Επιπλέον, παρατίθενται τα υλικά της κατασκευής στο παράρτημα Α και στο παράρτημα Β ο κώδικας σύμφωνα με τον οποίο λειτουργεί ο κύβος.

## Abstract

This diploma thesis aims at implementing of an interactive electronic 4x4x4 photodiode cube. The photodiodes have been soldered to the shape of a cube in three-dimensional form. The sixty-four LEDs are controlled by a microcontroller, which collects data from a temperature sensor.

Initially, this microcontroller was selected to meet the needs of this project. Then the temperature sensor and the photodiodes, which together with the appropriate materials, created the present construction. Then, the electronic circuit was designed and built. Finally, the software development in C + programming language completed the construction, providing communication to the most important components (microcontroller, thermostat, photodiode cube).

This paper differs from a simple construction of a 3d led cube, as it uses actual temperature data to operate, combining both appearance and practicality.

In brief, the first chapter begins with an introduction to the basic principles of functionality of the electronic cube. After that there is the second chapter that mentions the basic materials of this build. At the third chapter there is a form diagram of the cube which materialized in a simulation program of electronic components as well as its analysis and its building process. The next chapter refers to the programming of this build and the software that used. Also there is an explanation of the ADC function of the microcontroller. In the last chapter, there are two attached files. The first one shows the materials that were used and the second one is the code according to which the cube works.



# Κεφάλαιο 1<sup>ο</sup>

## Εισαγωγή

## 1.1 Εισαγωγή

Αυτή η πτυχιακή εργασία στοχεύει στην ανάλυση και καταγραφή όλων των βημάτων που απαιτούνται για την υλοποίηση ενός διαδραστικού-ηλεκτρονικού 3D κύβου φωτοδιόδων.

Η κατασκευή αυτή έχει διαστάσεις 9x9x9 cm και μορφή ενός γεωμετρικού κύβου που αποτελείται από εξήντα τέσσερις φωτοδιόδους, διατεταγμένες σε τέσσερις ορόφους και κάθε όροφος περιέχει από δεκαέξι led.

## 1.2 Κύβος φωτοδιόδων: Τι είναι

Ένας τέτοιος κύβος θυμίζει αρκετά τις σημερινές οθόνες LED, όπου ανάβοντας ή κρατώντας σβηστά κάποια led( pixel), απεικονίζεται αυτό που επιθυμεί ο χρήστης. Ο κύβος όμως διαθέτει μία επιπλέον διάσταση, δημιουργώντας έτσι την ψευδαίσθηση απεικόνισης τριών διαστάσεων. Αυτό επιτυγχάνεται, με την κατάλληλη τοποθέτηση των led σε συγκεκριμένη απόσταση και σε γεωμετρικό σχήμα κύβου δίνοντας έτσι πρόσβαση στο εσωτερικό του κύβου, σε αντίθεση με την οθόνες led, όπου στόχος είναι η τοποθέτηση των pixel όσο πιο κοντά γίνεται για καλύτερη ανάλυση.

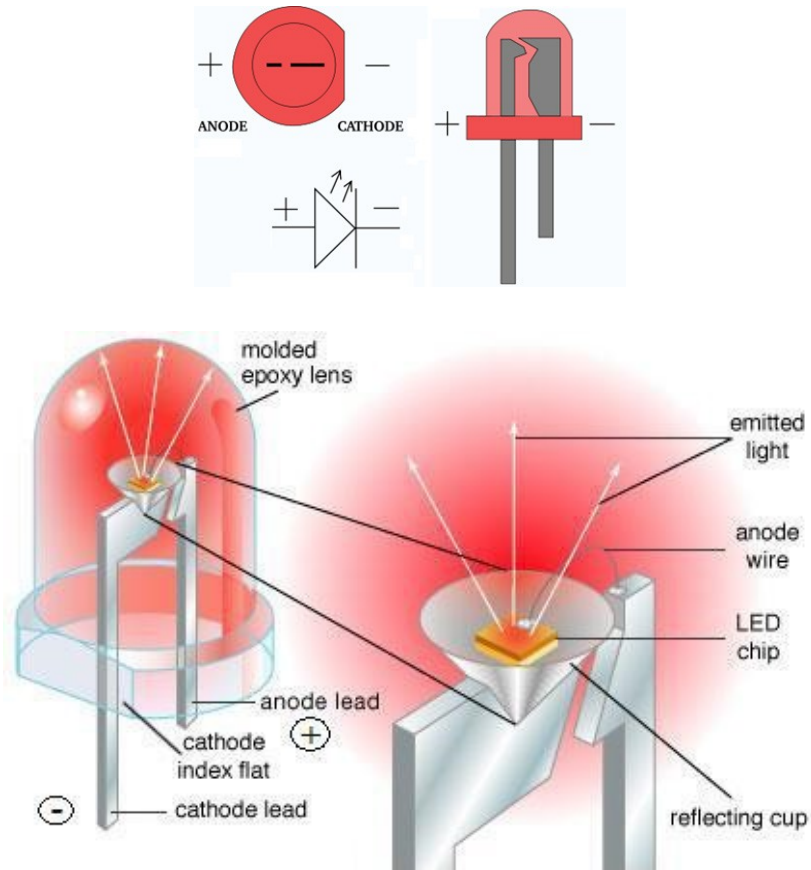
## 1.3 Κύβος φωτοδιόδων: Λειτουργία

Ο κύβος 4x4x4 αποτελείται από εξήντα τέσσερις φωτοδιόδους. Ο έλεγχος κάθε led ξεχωριστά ενός τέτοιου κύβου, και πόσο μάλλον κάποιου πολύ μεγαλύτερου, θα ήταν αδύνατο αφού θα χρειαζόμασταν έναν μικροελεγκτή με τουλάχιστον εξήντα τέσσερις πόρτες εισόδου-εξόδου. Αντ' αυτού, οι κύβοι led, βασίζονται σε ένα οπτικό φαινόμενο που ονομάζεται Persistence of Vision (POV), γνωστό και ως αισθητήρια μνήμη. Στην ουσία, εάν αναβοσβήσεις πολύ γρήγορα ένα led, η εικόνα του αναμμένου led θα παραμείνει στον αμφιβληστροειδή χιτώνα σου για λίγη ώρα αφότου έχει σβήσει. Σε συνδυασμό με μια μέθοδο πολυπλεξίας, όπου κάνοντας τις κατάλληλες συνδέσεις, χρειαζόμαστε τελικά μόνο είκοσι εισόδους-εξόδους ενός μικροεπεξεργαστή για τον έλεγχο κάθε ενός led του κύβου.[1]

## 1.4 Δίοδος εκπομπής φωτός

Δίοδος εκπομπής φωτός (LED, Light Emitting Diode) , ονομάζεται ένας ημιαγωγός που όταν τροφοδοτείται με τάση κατά τη φορά ορθής πόλωσης, εκπέμπει φωτεινή

ακτινοβολία στενού φάσματος. Ανάλογο με την χημική σύσταση του υλικού εκχέεται χρώμα φωτός, που μπορεί να είναι υπεριώδες, ορατό ή υπέρυθρο. Στην ουσία, όταν περνάει ρεύμα από την άνοδο προς την κάθοδο εκπέμπει ένα φως σε κάποιο μήκος κύματος και έτσι έχουμε τα διάφορα χρώματα. [2]

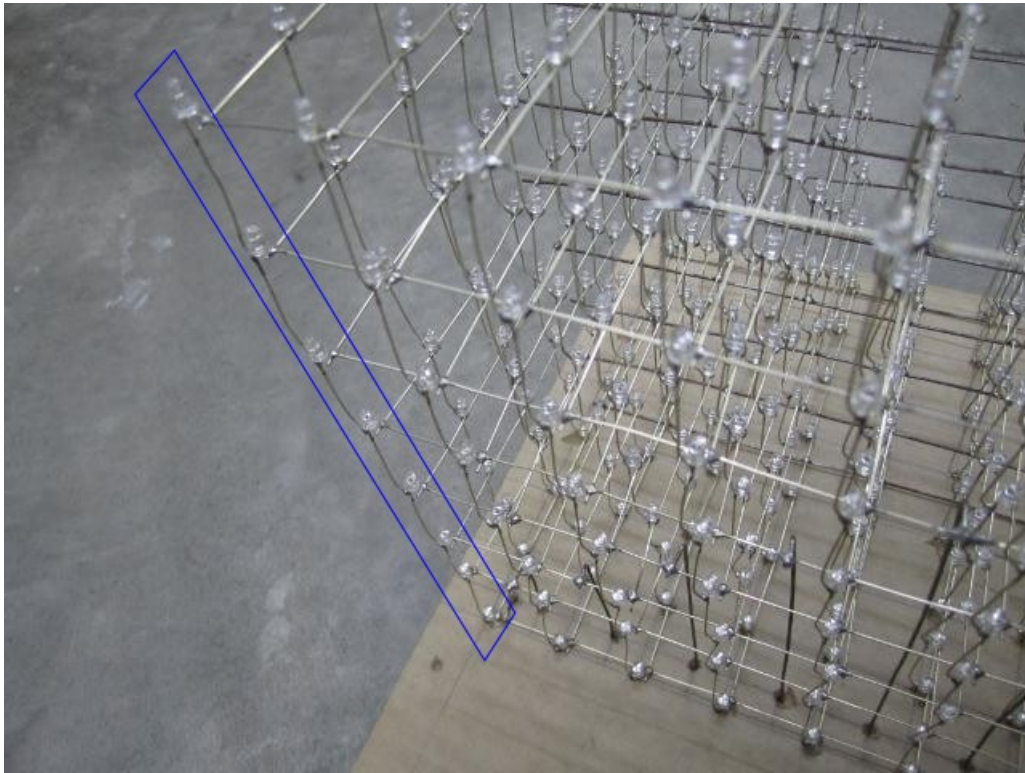


**Εικόνα 1-2: Δομή διόδου εκπομπής φωτός [2]**

### **1.5 Ανατομία και λειτουργία κύβου LED**

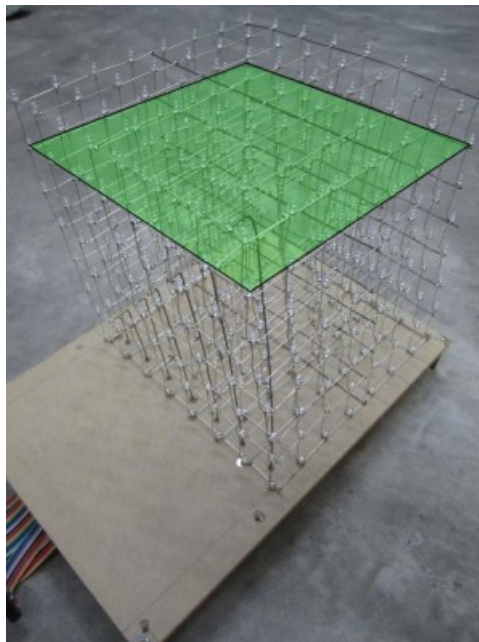
Αρχικά για να κατανοήσουμε τον κύβο αρκεί να έχουμε στο μυαλό μας τη λειτουργία του led, που για να εκπέμπει φως αρκεί να περάσει ρεύμα από την άνοδο προς την κάθοδο.

Ανατομία κύβου: στήλες και επίπεδα εικόνα 3,4 .[3]



**Εικόνα 3: Στήλες κύβου**

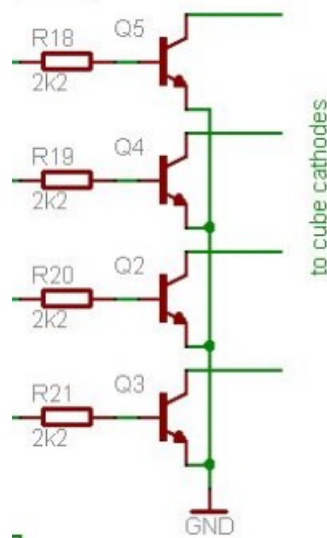
Οι led στήλες του κύβου έχουν κοινές ανόδους, όπως φαίνεται στην εικόνα, δημιουργώντας έτσι δεκαέξι ξεχωριστές στήλες.



**Εικόνα 4: Επίπεδα κύβου**

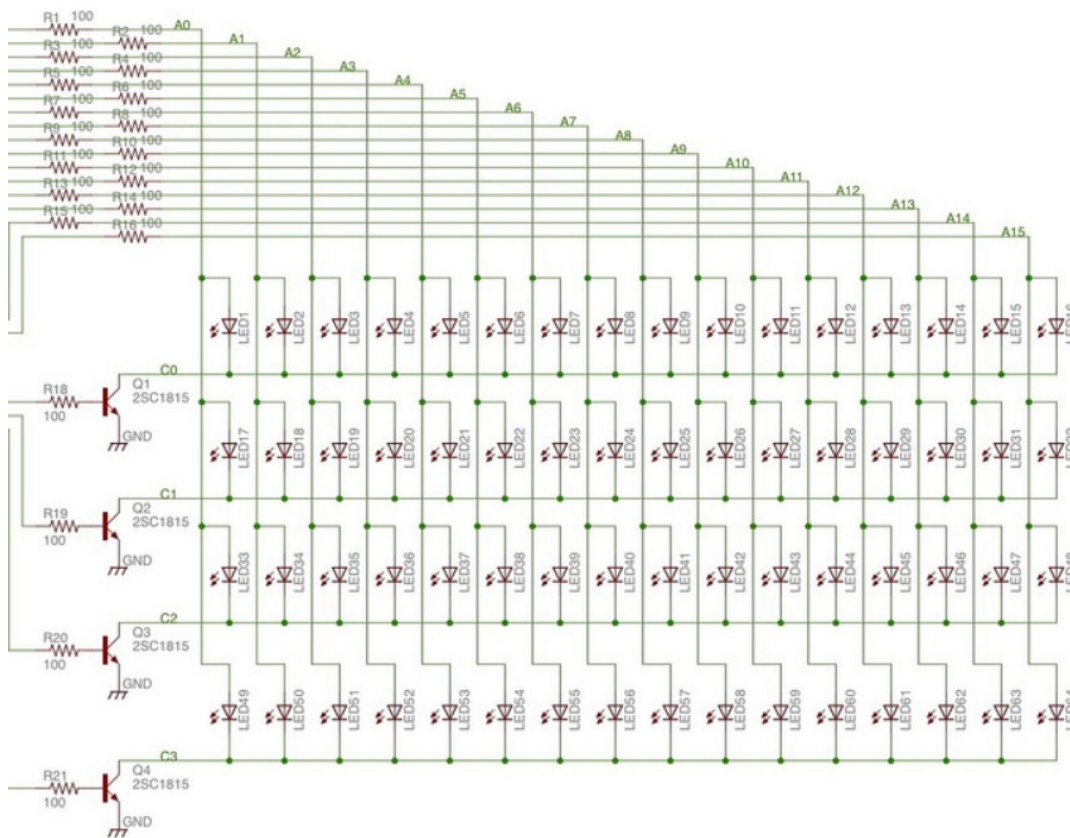
Το κάθε led επίπεδο του κύβου έχει κοινή κάθοδο, όπως φαίνεται στην εικόνα, δημιουργώντας έτσι τέσσερα ξεχωριστά επίπεδα.

Κάθε επίπεδο του κύβου, και κατά συνέπεια κάθε κάθοδος του κύβου, συνδέεται με ένα τρανζίστορ, συνολικά τέσσερα τρανζίστορ για τους τέσσερις ορόφους. Τα τρανζίστορ αυτά με την βοήθεια του μικροελεγκτή ενεργοποιούν ή απενεργοποιούν την ροή του ρεύματος από τα επίπεδα, επιτρέποντας έτσι στα led να εκπέμψουν φως ή όχι.



**Εικόνα 5: Τρανζίστορ που ελέγχουν τις καθόδους**

Στην εικόνα 6 βλέπουμε το τελικό εικονικό σχεδιασμό του κύβου που πραγματοποιήθηκε σε πρόγραμμα προσομοίωσης. Με αυτό το σχεδιασμό μπορούμε να ελέγξουμε κάθε ένα από τα led του κύβου ξεχωριστά, χρησιμοποιώντας μόνο 20 πύλες του μικροελεγκτή.



Εικόν

### α 6: Πλήρες διάγραμμα κύβου σχεδιασμένο σε πρόγραμμα προσομοίωσης

Η λειτουργία του κύβου είναι αρκετά απλή. Ο μικροελεγκτής έχει σαν έξοδο στις πόρτες του 5 Volts. Αυτή η τάση μπορεί να χρησιμοποιηθεί για την άνοδο του led, όχι όμως και για την κάθοδο. Έτσι, επιλέγουμε ένα τρανζίστορ, λειτουργώντας το ως διακόπτη. Αρχικά, ο μικροελεγκτής τροφοδοτεί με τάση, μέσω της αντίστασης, τη στήλη όπου βρίσκεται το επιθυμητό led. Ταυτόχρονα, δίνει τάση στο τρανζίστορ που ελέγχει το επίπεδο, μέσω της αντίστασης, ωθώντας το έτσι να άγει. Το αποτέλεσμα είναι το κύκλωμα να κλείσει και η φωτοδιόδος να ανάψει.

Για παράδειγμα, εάν θέλαμε να ανάψουμε το led 8, θα έπρεπε να στείλουμε υψηλό δυναμικό στην πόρτα που ελέγχει την στήλη A7 όπου βρίσκεται το συγκεκριμένο led, και ταυτόχρονα να στείλουμε υψηλό δυναμικό στη βάση του τρανζίστορ Q1, το επίπεδο που βρίσκεται το led, ώστε να άγει, να γειωθεί και τελικά να λάμψει το led. Αντίστοιχα, εάν

θέλαμε να ανάψουμε το led 55 θα έπρεπε να στείλουμε 5V στη στήλη A6 και στο τρανζίστορ Q3.

## 1.6 Αισθητήρας Θερμοκρασίας



Εικόνα 7: Αισθητήρας ακριβείας θερμοκρασίας[4]

Ο LM35, είναι ένας ακριβείας αισθητήρας θερμοκρασίας. Διαθέτει μια αναλογική έξοδο, ρυθμισμένη σε βαθμούς Κελσίου. Αποτελείται από τρία ποδαράκια, την τροφοδοσία, την γείωση και την έξοδο. Όπως φαίνεται και στο σχήμα για κάθε 10 mV εξόδου, έχουμε και ένα βαθμό Κελσίου.

## 1.7 Τα σημαντικότερα στοιχεία της κατασκευής

Ο μικροελεγκτής παίζει τον βασικότερο ρόλο στην κατασκευή, καθώς ελέγχει, με την βοήθεια του κατάλληλου λογισμικού – κώδικα και των δεδομένων του αισθητήρα , τις φωτοδιόδους.

Το επόμενο σημαντικό στοιχείο, είναι ο αισθητήρας θερμοκρασίας που στέλνει τις απαραίτητες, για την σωστή λειτουργία της κατασκευής, πληροφορίες στον μικροελεγκτή για να τις επεξεργαστεί και να δώσει το επιθυμητό αποτέλεσμα.

Τέλος, απαραίτητα στοιχεία αποτελούν οι δίοδοι φωτοεκπομπής, μιας και με την βοήθεια τους, γίνεται δυνατή η μετάδοση των πληροφοριών στον χρήστη.







## Κεφάλαιο 2<sup>ο</sup> Υλικά κατασκευής

## 2.1 Μικροελεγκτής PIC18F4550

Για την επιλογή του κατάλληλου μικροελεγκτή, δόθηκε βάση σε δύο απαραίτητα χαρακτηριστικά. Αρχικά, όπως έχει προαναφερθεί, λόγω του σχεδιασμού του κύβου, μπορούμε να τον ελέγξουμε πλήρως χρησιμοποιώντας μόνο είκοσι πόρτες εισόδου-εξόδου. Έπειτα, χρειαζόμαστε κάποιον ελεγκτή που να διαθέτει πόρτα ADC, Analog to Digital Converter, ώστε να μπορούμε να συλλέξουμε πληροφορίες από αναλογικό αισθητήρα όπως ο LM35. Έτσι, για τις ανάγκες της κατασκευής επιλέχθηκε ένας οικονομικός, εύχρηστος αλλά ισχυρός μικροελεγκτής, στην έκδοση με τα 40 pin, που μπορεί να καλύψει τις απαιτήσεις που τέθηκαν.

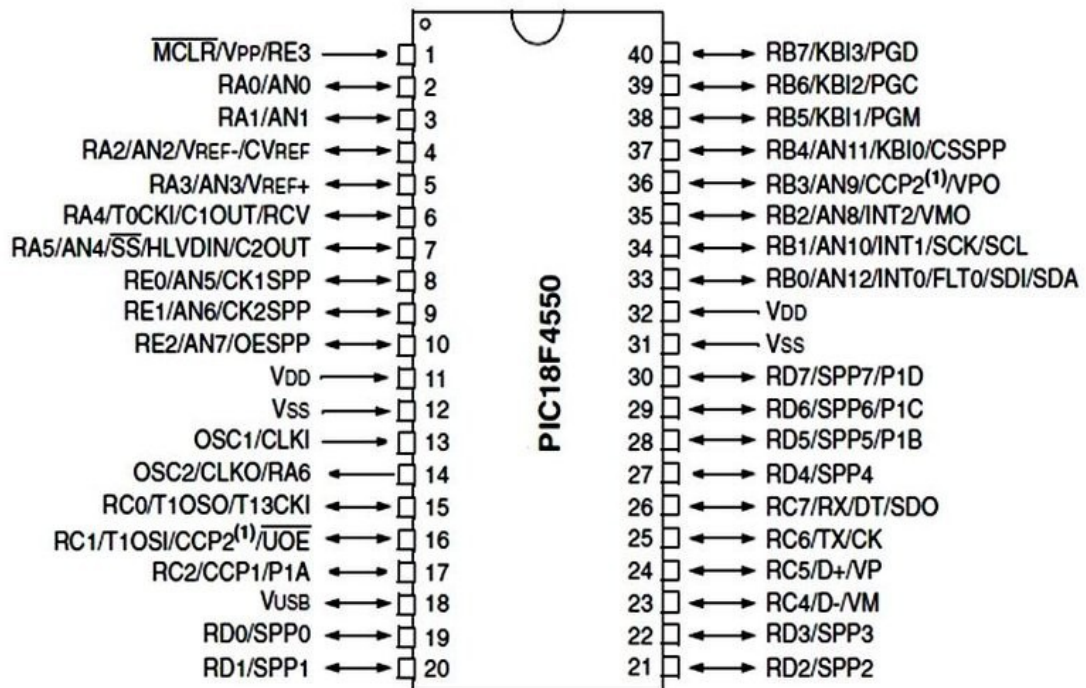


**Εικόνα 8: Μικροελεγκτής PIC 18F44550 της Microchip[5]**

Ο συγκεκριμένος μικροελεγκτής, αποτελείται από σαράντα ακροδέκτες. Παρέχει στο χρήστη τριάντα πέντε πόρτες εισόδου-εξόδου (I/O), εκ των οποίων οι 7 είναι και αναλογικές πόρτες. Διαθέτει ενσωματωμένο ταλαντωτή και δεν απαιτείται εξωτερικό κύκλωμα, απλοποιώντας έτσι το σχεδιασμό του κυκλώματος και ελευθερώνοντας δύο ακροδέκτες που μπορούν να χρησιμοποιηθούν ως πόρτες εισόδου εξόδου. Διαθέτει λειτουργία σειριακού προγραμματισμού, που μας επιτρέπει να τον χρησιμοποιήσουμε, μετά τη διαδικασία του προγραμματισμού, χωρίς να χρειάζεται να αφαιρεθεί από το κύκλωμα.

Αναλυτικά ο PIC 18F4550 είναι ένας οχτώ (8) bit και σαράντα-πιν μικροελεγκτής που ανήκει στην οικογένεια των PIC18. Έχει μνήμη, τύπου flash, για προγραμματισμό στα 32KB, RAM στα 2048Bytes, μνήμη EEPROM στα 256bytes και ταχύτητα επεξεργαστή στα 12MIPS (instructions per second). Διαθέτει εσωτερικό ταλαντωτή από 31Khz έως 8 Mhz. Κάποιες επιπλέον λειτουργίες, που τον ξεχωρίζουν από άλλες σειρές μικροελεγκτών, είναι οι πόρτες ανταλλαγής δεδομένων (Data Communication Ports). Διαθέτει 3 διαφορετικούς τύπους, USB port, Serial port και I2C - usb port.[5]

## 40-Pin

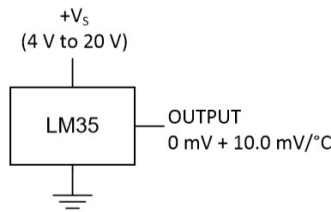


Εικόνα 9: Διάγραμμα ακροδεκτών μικροελεγκτή PIC18F4550[5]

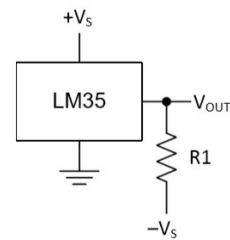
## 2.2 Αναλογικός αισθητήρας θερμοκρασίας ακριβείας LM35

Για την συγκεκριμένη κατασκευή επιλέχθηκε ένας αναλογικός αισθητήρας θερμοκρασίας της σειράς LM35 στην έκδοση των τριών ακροδεκτών, μορφής TO-92 (ευρέως ως transistor package). Μπορεί να χρησιμοποιηθεί με δύο τρόπους, ανάλογα με πιο εύρος θερμοκρασιών θέλει ο χρήστης να καλύψει. Όπως φαίνεται στην παρακάτω εικόνα, αριστερά για θερμοκρασίες από 2 έως 150 βαθμούς κελσίου, δεξιά επιλέξιμες θερμοκρασίες από -55 έως 150 βαθμούς κελσίου τοποθετώντας την επιθυμητή αντίσταση.

### Basic Centigrade Temperature Sensor (2°C to 150°C)



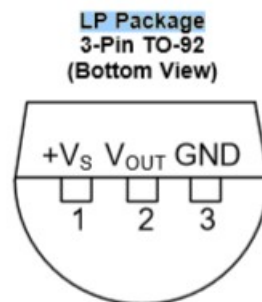
### Full-Range Centigrade Temperature Sensor



Choose  $R_1 = -V_S / 50 \mu\text{A}$   
 $V_{\text{OUT}} = 1500 \text{ mV at } 150^\circ\text{C}$   
 $V_{\text{OUT}} = 250 \text{ mV at } 25^\circ\text{C}$   
 $V_{\text{OUT}} = -550 \text{ mV at } -55^\circ\text{C}$

Εικόνα 10: Εύρος Θερμοκρασιών Im35 [6]

Ο LM35 είναι ρυθμισμένος σε βαθμούς Κελσίου. Για κάθε ένα βαθμό Κελσίου που εντοπίζει, μας δίνει +10 mV, με ακρίβεια μισού βαθμού. Μπορεί να λειτουργήσει με οποιαδήποτε τάση ανάμεσα στα 4V και 20 Volts. Έχει χαμηλή κατανάλωση της τάξης των 60  $\mu\text{A}$  και δεν χρειάζεται κάποιου είδους ψύξης. Η συνδεσμολογία του είναι απλή και φαίνεται στο παρακάτω διάγραμμα.[6]



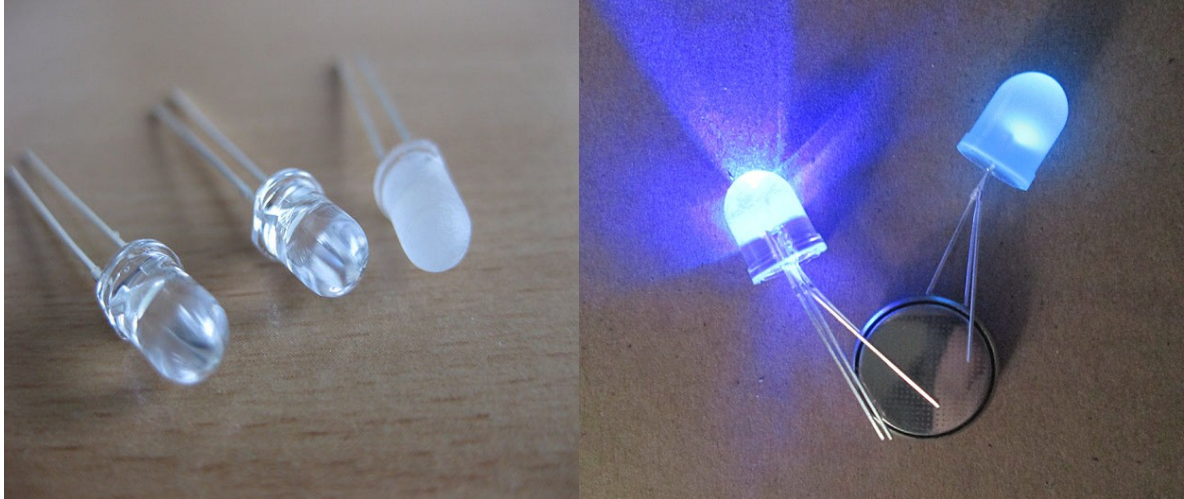
Εικόνα 11:Κάτοψη LM35[6]

## 2.3 Δίοδοι εκπομπής φωτός-LED

Η επιλογή των led, για ένα 3D led κύβο, μπορεί να επηρεάσει σε μεγάλο βαθμό την τελική εμφάνιση του, ειδικά αν πρόκειται για κύβο που παρουσιάζει κάποιο οπτικό εφέ. Σε αυτή την περίπτωση η καλύτερη επιλογή είναι diffused leds (διάχυτα) τα οποία στέλνουν το φως σε ίδια ποσότητα προς όλες τις κατευθύνσεις, αφού διαθέτουν μεγάλη γωνία θέασης. Ξεχωρίζουν σε εμφάνιση από το θάμπωμα στο εσωτερικό του διάφανου περιβλήματος τους, κάτι που δεν επιτρέπει στο led, όταν είναι σβηστό να επηρεαστεί από άλλα αναμμένα led.

Στην κατασκευή αυτή χρησιμοποιήθηκαν water clear led. Ξεχωρίζουν από την τελείως διάφανη κατασκευή τους, κάτι που προσφέρει εμφάνιση από μόνο του στον κύβο. Χαρακτηριστικό τους η μεγάλη διαφορά που προσφέρεται οπτικά στον χρήστη, όταν

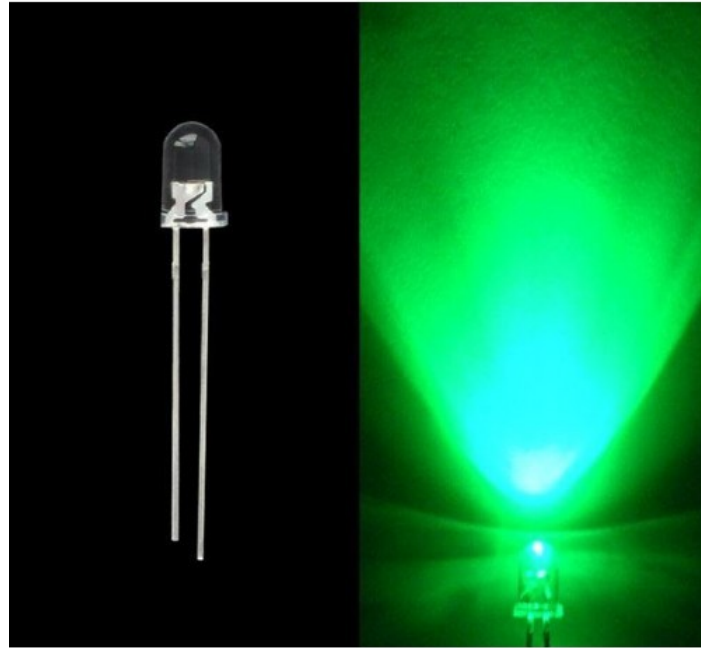
εισέλθουν από κατάσταση off σε on και το αντίστροφο. Αυτό οφείλεται στην κατασκευή τους, αφού λόγω του διάφανου υλικού τους έχουν την μεγαλύτερη φωτεινότητα από κάθε άλλο είδος φωτοδίοδου. Μειονέκτημα τους, η μικρή γωνία θέασης μόλις 30 μοιρών.



**Εικόνα 12: Diffused –Water clear φωτοδίοδος**

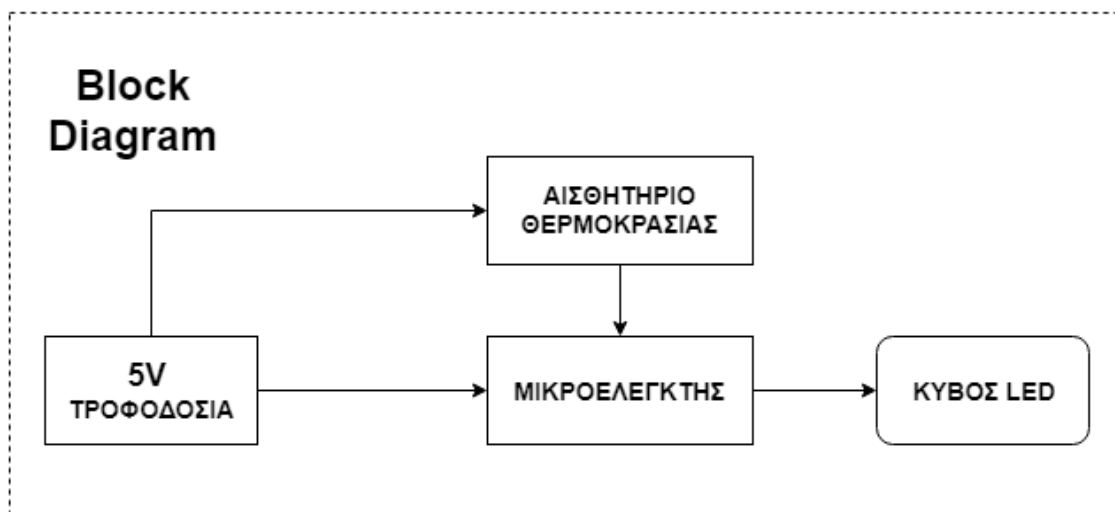
Τα χαρακτηριστικά των φωτοδίοδων που χρησιμοποιήθηκαν είναι:

- Water clear φωτοδίοδοι πράσινου χρώματος με διάφανο περίβλημα
- 5mm διάμετρος
- 3.2V – 3.8V τάση λειτουργίας
- 20mA (τυπικά), 30mA (ανώτατη) κατανάλωση
- 520nm Μήκος κύματος
- φωτεινότητα 15.000 mcd (millicandela)
- 30 μοίρες γωνία θέασης



Εικόνα 13: Πράσινη με διάφανο περίβλημα φωτοδίοδος

## 2.4 Block διάγραμμα κατασκευής



Εικόνα 14: Block διάγραμμα κύβου

Στο παραπάνω διάγραμμα, μπορούμε να πάρουμε μία πρώτη εικόνα, μέσω του διαγράμματος, για τα υλικά που χρησιμοποιήθηκαν και τη σύνδεση τους στην κατασκευή αυτή.

Αρχικά η κατασκευή τροφοδοτείται από μία πηγή τάσης 5Volt και τουλάχιστον 1Ampere. Η πηγή τροφοδοτεί, τόσο το αισθητήριο θερμοκρασίας όσο και τον μικροελεγκτή και κατά συνέπεια τα led του κύβου.

Για να λειτουργήσει σωστά το αισθητήριο, είναι αναγκαία η σταθερή τάση, ώστε οι παράμετροι που έχουμε θέσει να υπολογίσουν σωστά και να προσφέρουν το αποτέλεσμα στον μικροελεγκτή χωρίς απόκλιση. Οι υπολογισμοί αυτοί περιέχουν σαν σταθερές μεταβλητές την τάση τροφοδοσίας του κυκλώματος και την γείωση.

Έπειτα, ο μικροελεγκτής χρειάζεται επίσης σταθερή τάση, κάτι που εξασφαλίζεται και από τους πυκνωτές αποσύζευξης, για να μπορέσει να λειτουργήσει σύμφωνα με τις προδιαγραφές του κατασκευαστή. Από την λειτουργία του μικροελεγκτή επηρεάζονται οι τιμές τάσης για την πόλωση των διόδων αλλά και η ταχύτητα του επεξεργαστή και κατά συνέπεια η ορθή εκτέλεση του κώδικα.

Τέλος, για να προσφέρουν οι διόδοι φωτοεκπομπής τη σωστή φωτεινότητα, θα πρέπει πέρα από όλα τα παραπάνω, να έχουν γίνει σωστοί υπολογισμοί στις αντιστάσεις του κυκλώματος, ψαλιδίζοντας τα ρεύματα τόσο στα led όσο και στα τρανζίστορ των γειώσεων.

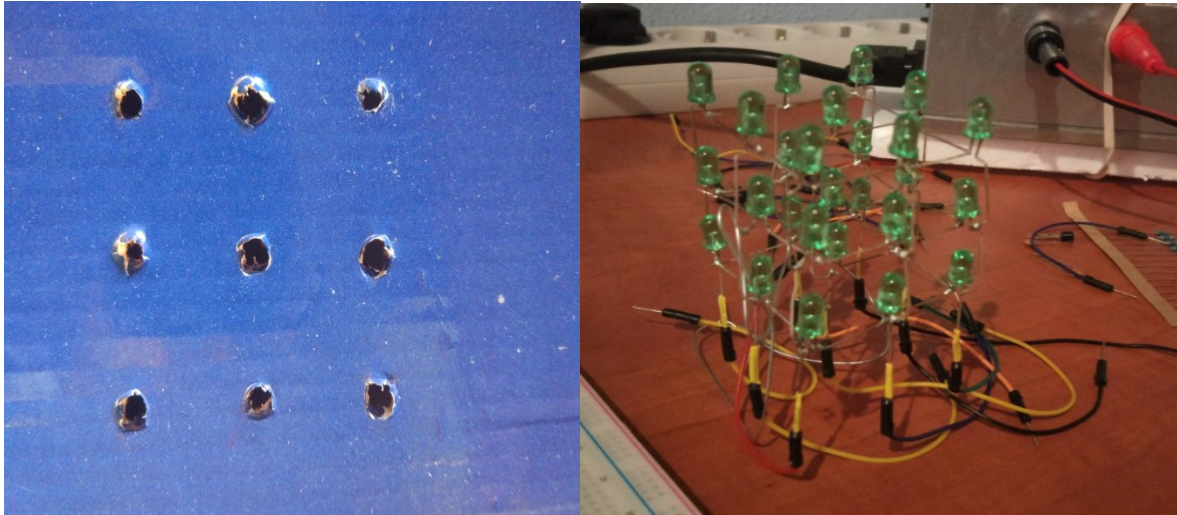




## Κεφάλαιο 3<sup>ο</sup> Κατασκευή

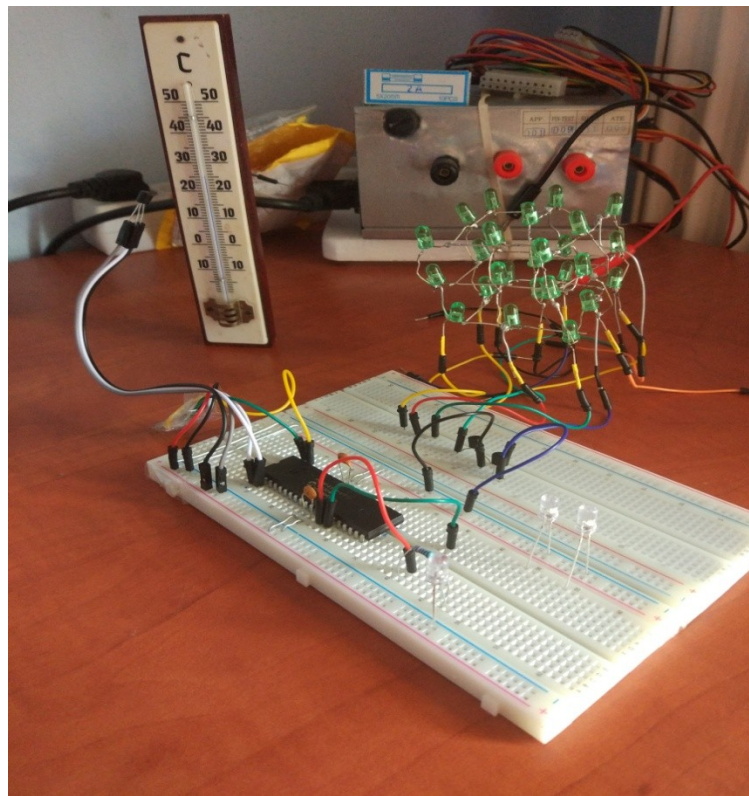
### 3.1 Κατασκευή του κύβου φωτοδιόδων

Αρχικά, προκειμένου να κατανοήσω τον τρόπο λειτουργίας του κύβου, κατασκεύασα έναν δοκιμαστικό κύβο 3x3x3, με διάχυτα πράσινα led 5mm, διαστάσεων 4x4x4cm. Έτσι ανοίχτηκαν τρύπες λίγο μικρότερες από 5mm, ώστε να στηριχτεί και να κολληθεί ο κύβος.



**Εικόνα 15-15.1: Κατασκευή δοκιμαστικού κύβου**

Με τον κύβο αυτό, έγιναν όλες οι απαραίτητες δοκιμές της κατασκευής και γράφτηκε ο πρώτος κώδικας.



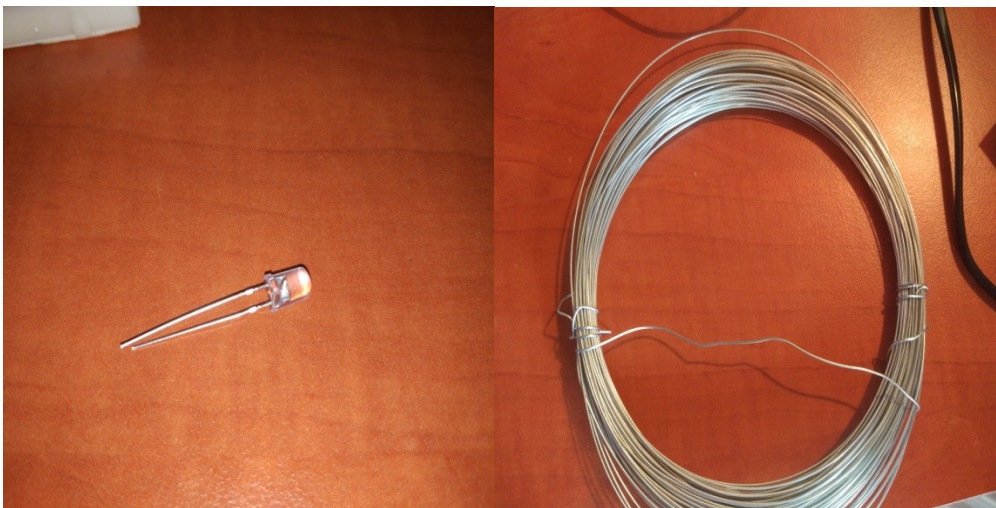
### **Εικόνα 16: Δοκιμές ορθής λειτουργίας σε breadboard**

Έπειτα, αφού αποφασίστηκε το τελικό μέγεθος του κύβου, δημιουργήθηκε η βάση του. Η κατασκευή θα έχει διαστάσεις κύβου 9x9x9cm, με μέγεθος led 5mm.

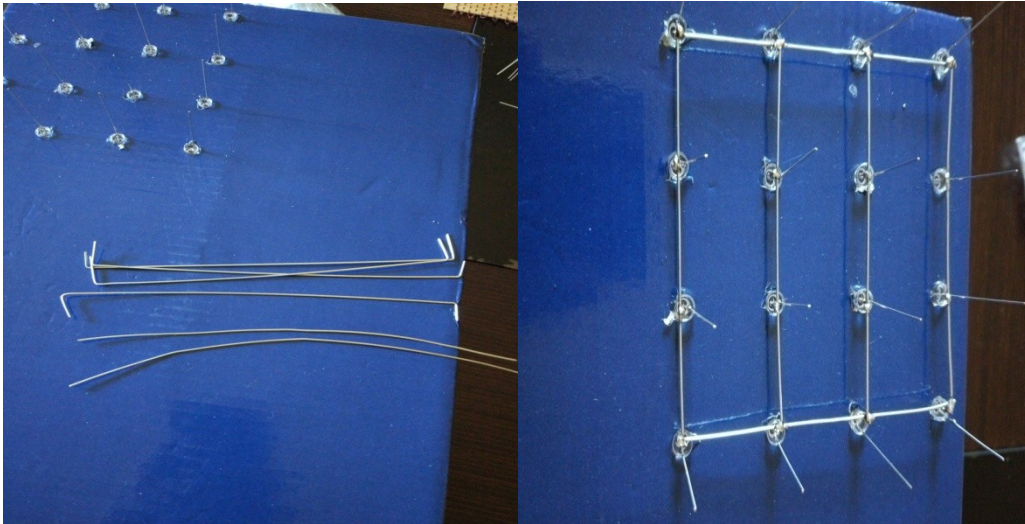


**Εικόνα 17: Βάση κατασκευής κύβου 4X4X4**

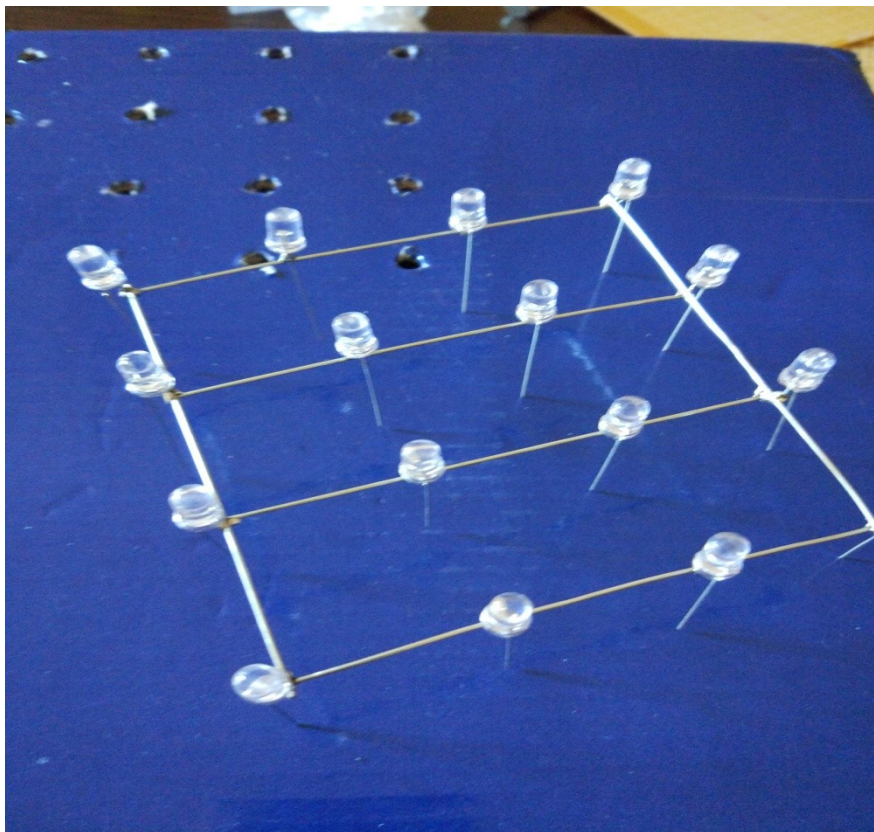
Σε τρίτη φάση, τοποθετήθηκαν τα led, και οι κάθοδοι τους κόπηκαν ώστε να κολληθεί στη θέση τους ασάλινο σύρμα πάχους ενός χιλιοστού, ενώνοντας όλες τις καθόδους του επιπέδου. Χρησιμοποιήθηκε συσκευή συγκόλλησης. Σε παρακάτω εικόνες φαίνεται η κατασκευή ενός επιπέδου του κύβου.



**Εικόνα 18-19: Δίοδος φωτοεκπομπής κατασκευής και ασάλινο σύρμα**



**Εικόνα 20-21: Τοποθέτηση led και σύνδεση καθόδων σε σειρά**



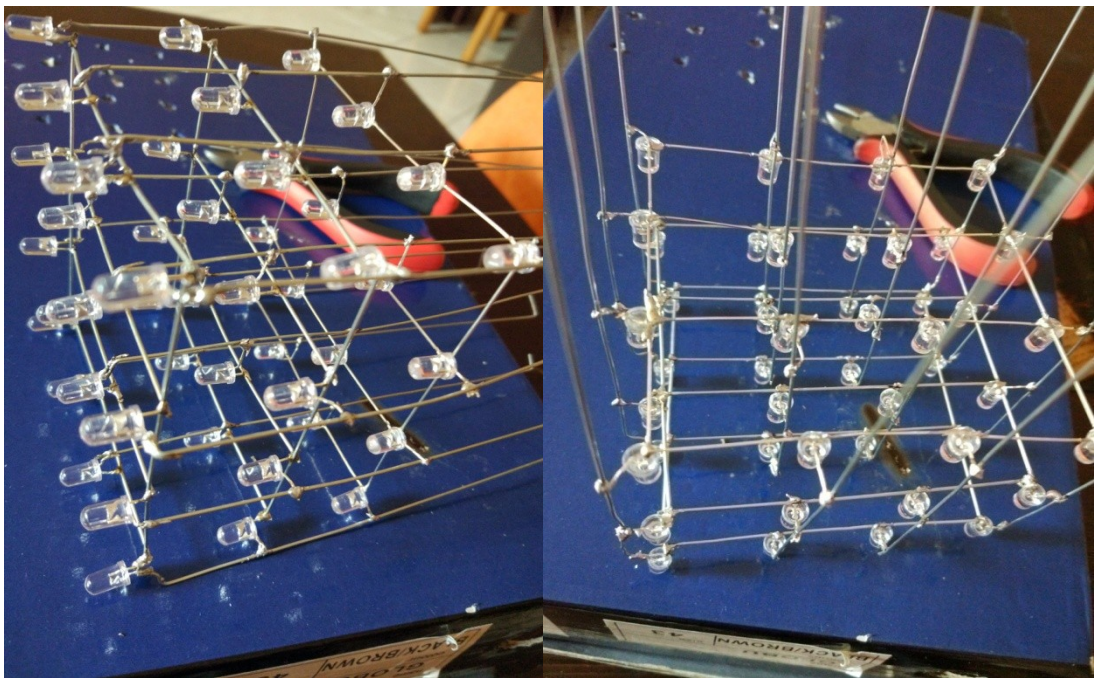
**Εικόνα 22: Ολοκλήρωση κατασκευής ενός επιπέδου**

Ολοκλήρωση κατασκευής όλων των επιπέδων του κύβου με τον παραπάνω τρόπο.



**Εικόνα 23: Κατασκευή τεσσάρων επιπέδων κύβου**

Στο τελικό στάδιο, κάθε επίπεδο τοποθετείτε πάνω στο άλλο, σχηματίζοντας έτσι την μορφή του κύβου. Οι ακροδέκτες των led (άνοδοι) κόβονται στο επιθυμητό μήκος και τοποθετούνται στύλοι ατσάλινου σύρματος ώστε να ενωθούν μεταξύ τους με τον παρακάτω τρόπο.



**Εικόνα 24-25: Σύνδεση των επιπέδων**



Οι αντιστάσεις που χρησιμοποιήθηκαν στο κύκλωμα έχουν σκοπό να ελέγξουν το ρεύμα που διαρρέει τις φωτοδιόδους αλλά και το ρεύμα των τρανζίστορ. Οι αντιστάσεις των led έχουν τιμή στα 220Ω και περιορίζουν το ρεύμα των led στα 12mA περίπου, ενώ των τρανζίστορ έχουν τιμή 2.2KΩ

### **Πυκνωτές**

Υπάρχουν 2 πυκνωτές στο κύκλωμα και είναι πυκνωτές αποσύζευξης, τιμής 0.1μF κεραμικοί ο καθένας . Τοποθετούνται στους ακροδέκτες τροφοδοσίας του μικροελεγκτή, όσο πιο κοντά γίνεται. Πρακτικά προστατεύουν τον μικροελεγκτή από τους θορύβους του κυκλώματος, προστατεύοντας τον έτσι από τυχόν παρεμβολές

### **BC338 (Q1-Q4)**

Χρησιμοποιήθηκαν 4 τρανζίστορ NPN, ένα για κάθε κάθοδο από τα 4 επίπεδα. Τα συγκεκριμένα μπορούν να αντέξουν έως και 800mA, οπότε είναι ικανά να οδηγήσουν τον κύβο μας. Ο συλλέκτης του τρανζίστορ είναι συνδεδεμένος με την κάθοδο του επιπέδου του κύβου. Ο εκπομπός συνδέεται με την γείωση. Τέλος η βάση του τρανζίστορ συνδέεται μέσω της αντίστασης στον μικροελεγκτή. Όταν ο μικροελεγκτής οδηγεί την έξοδο σε υψηλό δυναμικό, τότε το τρανζίστορ άγει, και το led γειώνεται.

### **Βύσμα τροφοδοσίας-κουμπί τροφοδοσίας**

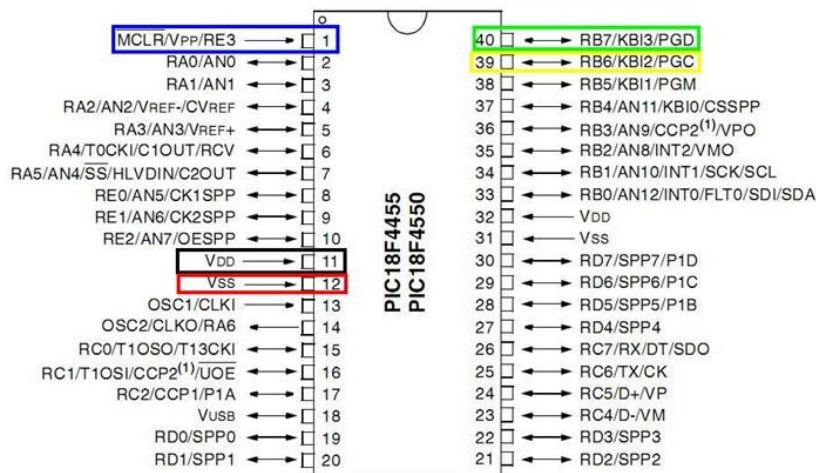
Η τροφοδοσία του κυκλώματος γίνεται μέσω ενός 2 πιν βύσματος και ελέγχεται από ένα κουμπί πριν φτάσει στο κύκλωμα. Το βύσμα πρέπει να είναι συνδεδεμένο σε τροφοδοτικό 5V και τουλάχιστον 1 Amp.

Επιπλέον, έχουν χρησιμοποιηθεί, μια 40 πιν θηλυκή βάση για τον μικροελεγκτή, μία 4 πιν θηλυκή βάση για τον αισθητήρα και μία 6 πιν βάση αρσενική για τον προγραμματιστή Pik kit 3 καθώς επίσης ατσάλινο σύρμα και καλώδιο.

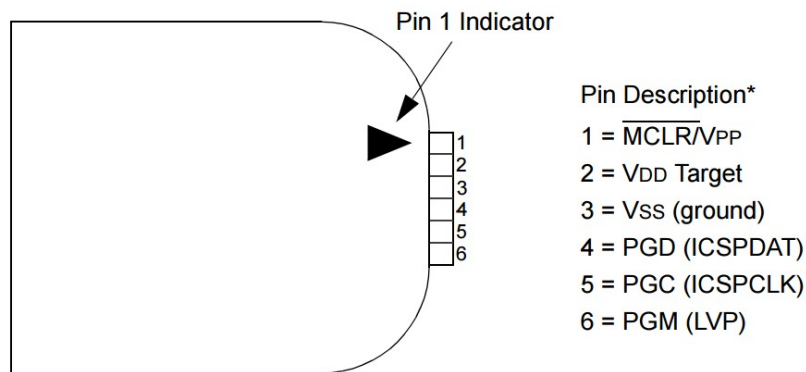
## **3.3 Προγραμματισμός μικροελεγκτή**

Ο 18F4550, μαζί με τον προγραμματιστή Pik Kit 3, μας δίνει την δυνατότητα να τον προγραμματίσουμε ενώ είναι συνδεδεμένος στο κύκλωμα. Για να το πετύχουμε αυτό αρκεί να ακολουθήσουμε την παρακάτω συνδεσμολογία.[7]





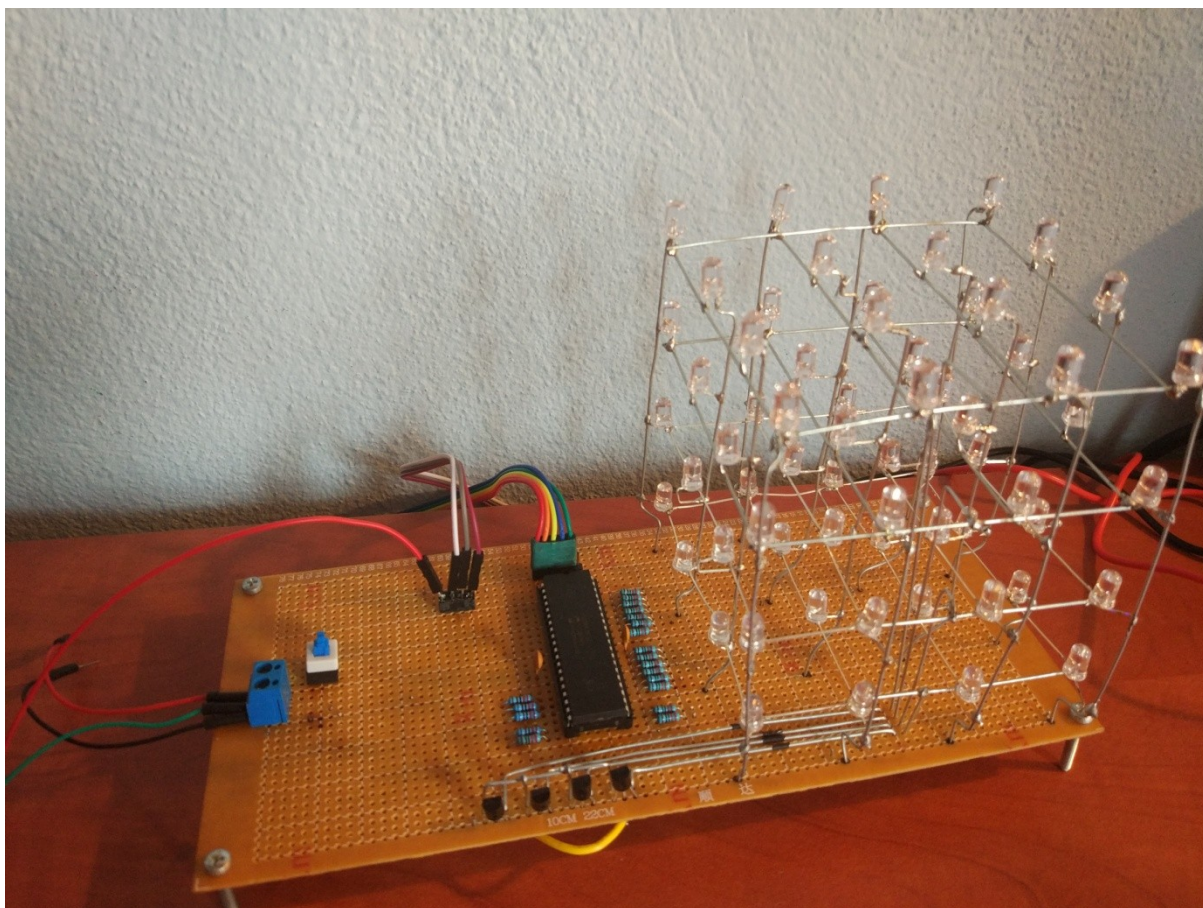
**Εικόνα 27: Ακροδέκτες προγραμματισμού 18F4550[5]**



**Εικόνα 28: Ακροδέκτες προγραμματιστή Pik kit 3[7]**

### 3.4 Τελική κατασκευή κύβου σε πλακέτα

Όλα τα υλικά έχουν τοποθετηθεί πάνω σε prototype board. Η διαστάσεις του είναι 22cmX10cm. Στις άκρες έχουν τοποθετηθεί βίδες ώστε όλη η κατασκευή να υψωθεί στα 2 εκατοστά.



**Εικόνα 29: Τελική μορφή της κατασκευής**



## Κεφάλαιο 4<sup>ο</sup> Λογισμικό

## 4.1 Τι είναι ο μικροελεγκτής

Οι μικροελεγκτές, είναι προγραμματιζόμενα ολοκληρωμένα κυκλώματα, τα οποία μόλις τροφοδοτηθούν με ρεύμα, εκτελούν εντολές που τους έχουμε δώσει σε μορφή κώδικα. Διαθέτουν επεξεργαστή, μνήμη, περιφερειακά κυκλώματα όπως επίσης και θύρες εισόδου-εξόδου(I/O) για επικοινωνία με εξωτερικές συσκευές. Μπορούν να χρησιμοποιηθούν οπουδήποτε υπάρχει ανάγκη για έλεγχο συστημάτων. Με τον όρο Embedded Systems (ενσωματωμένα συστήματα) εννοούμε οποιοδήποτε σύστημα βασίζεται σε μικροεπεξεργαστή. [8]

Υπάρχουν τρία στάδια που ακολουθεί κάποιος για να χρησιμοποιήσει ένα μικροελεγκτή:

-Αρχικά συνθέτει τον κώδικα στην γλώσσα προγραμματισμού που ο ίδιος επέλεξε. Υπάρχουν δύο κατηγορίες γλωσσών, χαμηλού επιπέδου όπως assembly-γλώσσα μηχανής και υψηλού επιπέδου όπως αυστηρά δομημένες γλώσσες ( π.χ. C) που χρειάζονται compiler για να μετατρέψει τον κώδικα σε γλώσσα μηχανής.

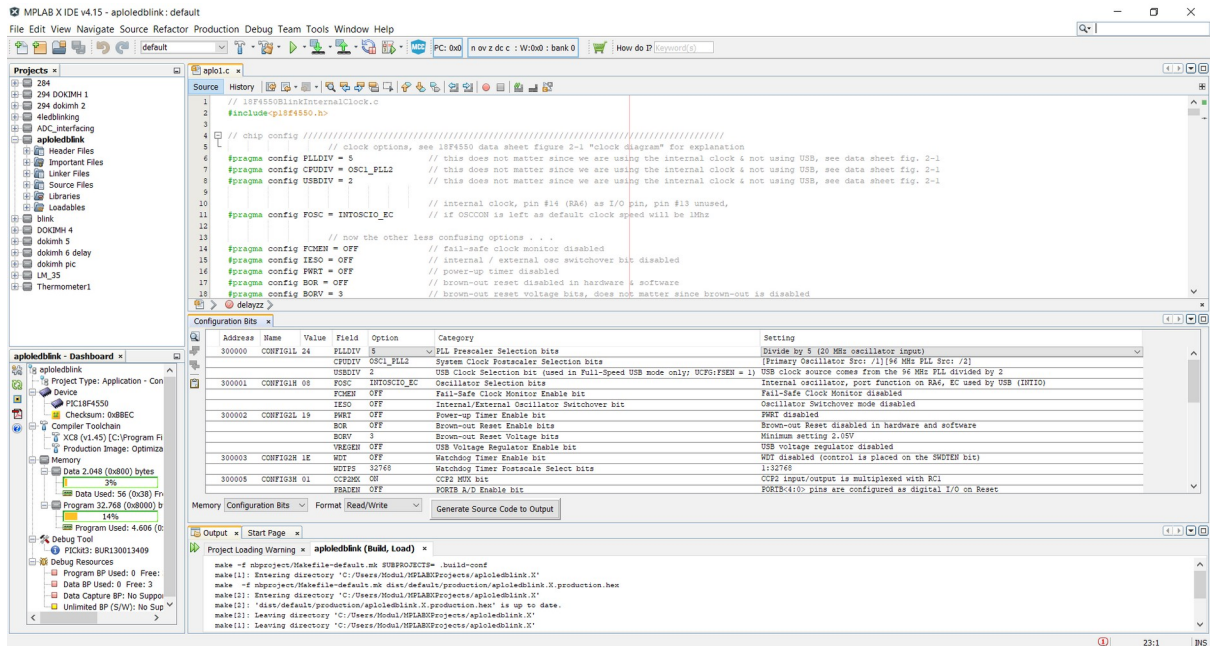
-Έπειτα τοποθετεί τον κώδικα στη μνήμη του μικροελεγκτή

-Τέλος, κάνει τις απαραίτητες συνδέσεις του μικροελεγκτή με εξωτερικά κυκλώματα ώστε να τα ελέγξει.

Αυτά τα τρία στάδια πρέπει να ακολουθήσουμε για να ελέγξουμε τον κύβο φωτοδιόδων.

## 4.2 Στάδιο πρώτο: Σύνθεση κώδικα και μετατροπή σε γλώσσα μηχανής

Η εταιρεία Microchip, παρέχει, δωρεάν στους χρήστες των μικροελεγκτών της, το πρόγραμμα MPLAB X IDE, που περιέχει εκτός των άλλων πρόγραμμα σύνθεσης κώδικα, προσομοίωσης ηλεκτρονικών κυκλωμάτων και compiler. Επίσης είναι συμβατό με τον προγραμματιστή που θα χρησιμοποιήσουμε, τον PicKit3. Για την σύνθεση του κώδικα, χρησιμοποιήθηκε γλώσσα υψηλού επιπέδου C. Στην παρακάτω εικόνα φαίνεται το πρόγραμμα MPLAB X IDE.[9][10]



Έπειτα χρησιμοποιήθηκε ο ενσωματωμένος compiler του προγράμματος για την μετατροπή του κώδικα σε hex αρχείο (hexadecimal source file = δεκαεξαδικό αρχείο) .

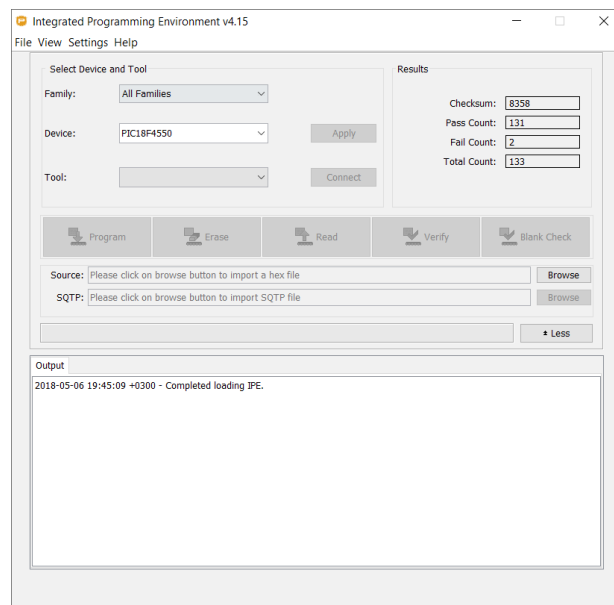
**Εικόνα 30: MPLAB X IDE**

### 4.3 Στάδιο δεύτερο: Τοποθέτηση κώδικα στη μνήμη του μικροελεγκτή

Για την τοποθέτηση του κώδικα, χρησιμοποιήθηκε το δωρεάν πρόγραμμα της microchip MPLAB IPE καθώς επίσης και ο Pickit3. Ο προγραμματιστής αυτός είναι οικονομικός, πολύ εύχρηστος και υποστηρίζεται από την ίδια εταιρεία με τον μικροελεγκτή. Συνδέεται με τον υπολογιστή με μία θύρα usb, και με τον μικροελεγκτή με 6 συνδέσεις, όπως αναλύσαμε στο προηγούμενο κεφάλαιο. Μπορεί να υποστηρίξει μία τεράστια γκάμα μικροελεγκτών, συμπεριλαμβανόμενα τα τελευταία και πιο απαιτητικά μοντέλα της εταιρείας. Φαίνεται στην παρακάτω εικόνα.



**Εικόνα 31: Προγραμματιστής PicKit3**



**Εικόνα 32: MPLAB IPE**

#### 4.4 Διάγραμμα ροής κύβου

Αρχικά δηλώνονται όλες οι απαραίτητες μεταβλητές για την λειτουργία του μικροελεγκτή και γίνεται αρχικοποίηση των θυρών, καταχωρητών και μεταβλητών. Επιπλέον δηλώνονται οι συναρτήσεις που χρησιμοποιούνται στον κώδικα και οι χρονικές καθυστερήσεις.

Έπειτα ο κώδικας ανάβει για κάποια δευτερόλεπτα τα led του κύβου, με σκοπό τον έλεγχο από τον χρήστη, εάν υπάρχει κάποιο πρόβλημα.

Στη συνέχεια ξεκινά το κυρίως πρόγραμμα. Πρώτον ενεργοποιείτε η αναλογική είσοδος του ελεγκτή στον ακροδέκτη ANO. Δεύτερον, καλείται η ρουτίνα που διαβάζει το αποτέλεσμα της θύρας και το καταχωρεί σε μία μεταβλητή. Τρίτον, καλείται ρουτίνα που μετατρέπει τα δεδομένα της πόρτας ANO , σε μορφή επεξεργάσιμη προς τον επεξεργαστή. Τέλος, ο μικροελεγκτής ανάβει τα κατάλληλα led ανάλογα με τη θερμοκρασία που έχει ο χώρος, καλεί μια ρουτίνα καθυστέρησης και επιστρέφει στην αρχή του κυρίως προγράμματος για να επαναλάβει ξανά τον έλεγχο.

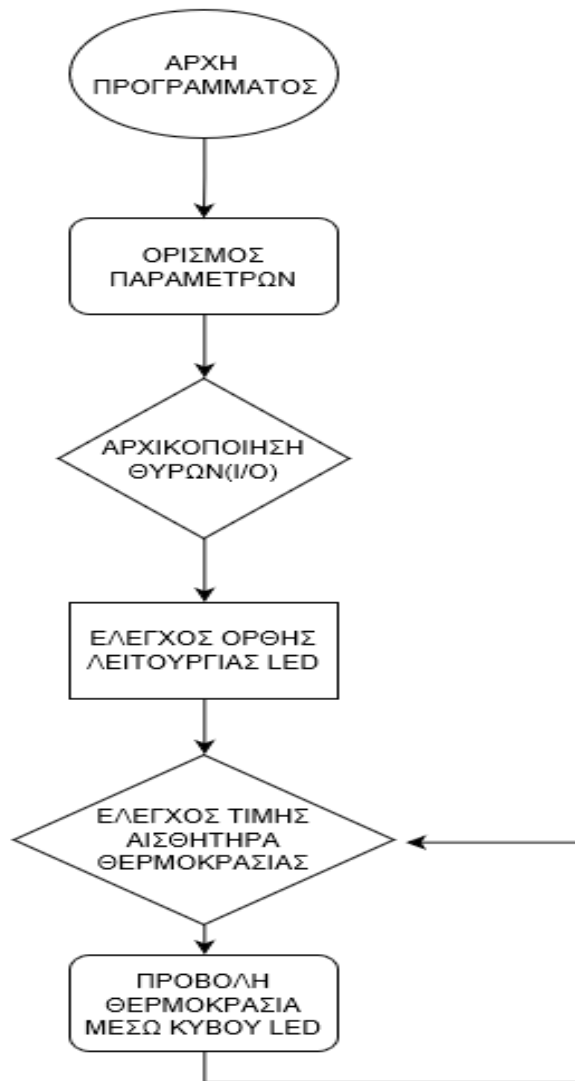
Ο μικροελεγκτής έχει συχνότητα λειτουργίας τα 8MHz, οπότε η διάρκεια κύκλου εντολής είναι 0.5μsec. Αυτό υπολογίζεται ως εξής:

Συχνότητα του μικροελεγκτή  $f_{osc}=8\text{MHz}$ . Τότε η περίοδος θα είναι  $T_{osc}=1/f_{osc}=0.125\mu\text{s}$ . Άρα ένας πλήρης κύκλος εκτέλεσης εντολής έχει διάρκεια 0.5μsec.

Επομένως, η διάρκεια εκτέλεσης ενός προγράμματος, ισούται με τον αριθμό των εντολών που το αποτελούν επί τη διάρκεια του κύκλου εκτέλεσης της εντολής. Κάποιες εντολές χρειάζονται περισσότερο από έναν κύκλο μηχανής για να εκτελεστούν. [11]



Διάγραμμα ροής κύβου



**Εικόνα 33: Διάγραμμα ροής κώδικα**

Ο κύβος έχει επιλεγεί να απεικονίζει θερμοκρασίες από 20 έως 50 βαθμούς Κελσίου, με ανάλυση δύο βαθμών. Ο χρήστης μπορεί πολύ εύκολα να μάθει τη θερμοκρασία του χώρου απλά συμβουλευόντας το παρακάτω διάγραμμα X-Y-Z αξόνων.

Στους 20 βαθμούς Κελσίου, ανάβουν τα led (1- 1,2,3,4 -A) (X-Y-Z) .

Στους 22 βαθμούς Κελσίου, ανάβουν τα led (1- 1,2,3,4 -B) και αυτά των 20 βαθμών.

.....

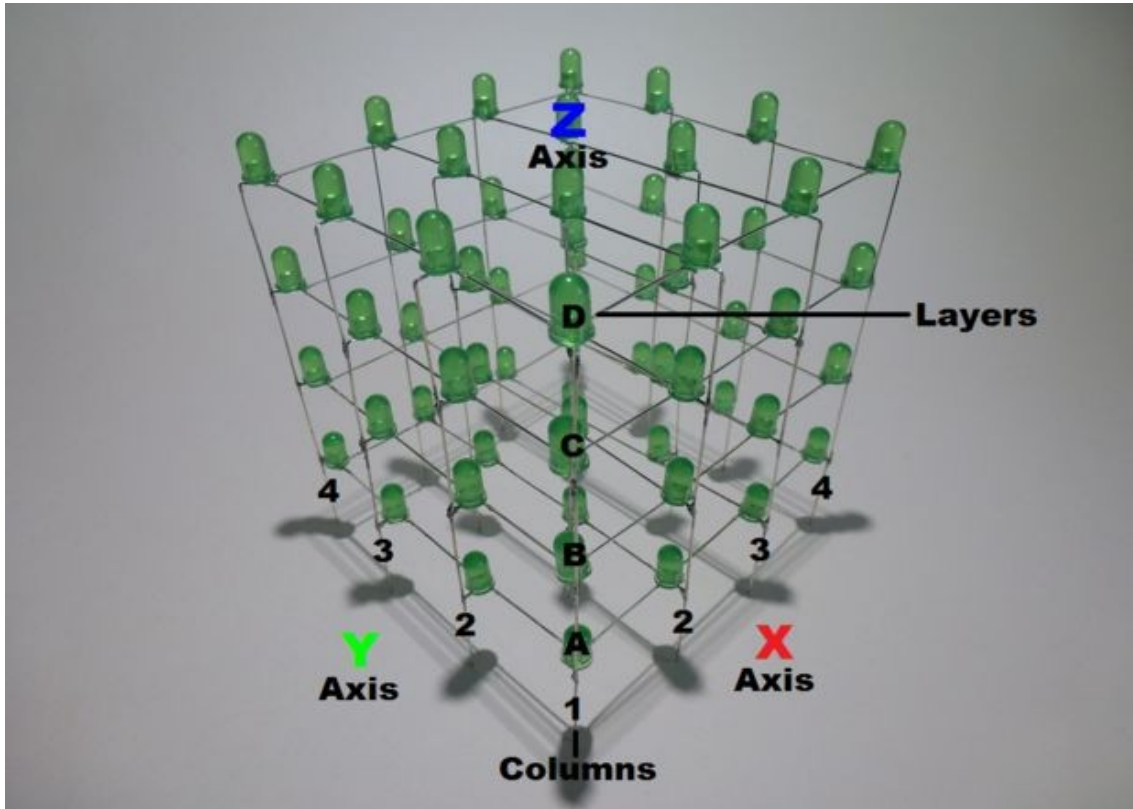
Στους 30 βαθμούς Κελσίου, ανάβουν τα led (2- 1,2,3,4 -B) και όλα τα προηγούμενα.

.....

Στους 40 βαθμούς Κελσίου, ανάβουν τα led (3- 1,2,3,4 –C) και όλα τα προηγούμενα.

.....

Στους 50 βαθμούς Κελσίου, ανάβουν τα led (4- 1,2,3,4 –D) και όλα τα προηγούμενα.



**Εικόνα 34:Λειτουργία απεικόνισης θερμοκρασίας κύβου[12]**

Ας δούμε ένα παράδειγμα κώδικα που ενεργοποιεί τον κύβο για θερμοκρασία 20 βαθμών κελσίου.

Αφού έχουν γίνει όλες οι απαραίτητες ρυθμίσεις παραμέτρων, ο κώδικας μπαίνει σε μία while, όπου οι εντολές που βρίσκονται μέσα θα εκτελούνται επ' άοριστον.

```
while(1)  
  
{  
  
    ADC_Init();  
    //Καλείτε ρουτίνα που ρυθμίζει τις παραμέτρους για τη λειτουργία Analog to Digital  
    και την θέτει σε κατάσταση ON  
  
    Start_Conversion();
```

//Καλείτε ρουτίνα που πραγματοποιεί την μετατροπή της τιμής από αναλογική σε ψηφιακή

```
adc_val= Get_ADC_Result();
```

//Καλείτε ρουτίνα που αποθηκεύει το αποτέλεσμα της μετατροπής σε μία μεταβλητή

```
Voltage = (long) adc_val*500.0;
```

```
adc_val = Voltage /1024.0;
```

```
k = adc_val;
```

//Η τιμή της μεταβλητής, μετατρέπεται σε βαθμούς Κελσίου μέσω κάποιων υπολογισμών, ανάλυση γίνεται στην επόμενη ενότητα και αποθηκεύεται σε μια μεταβλητή κ.

```
if (k>20){
```

```
    PORTBbits.RB3 = 1;
```

```
    PORTDbits.RD0 = 1;
```

```
    PORTDbits.RD6 = 1;
```

```
    PORTDbits.RD7 = 1;
```

```
    PORTDbits.RD3 = 1;
```

```
    __delay_ms(1);
```

```
    PORTBbits.RB3 = 0;
```

```
    PORTDbits.RD0 = 0;
```

```
    PORTDbits.RD6 = 0;
```

```
    PORTDbits.RD7 = 0;
```

```
    PORTDbits.RD3 = 0;
```

```
}
```

Εφόσον η τιμή  $\kappa$ , είναι μεγαλύτερη του 20 (βαθμοί κελσίου), ενεργοποιούνται τα led του κύβου που έχουμε ορίσει για την συγκεκριμένη θερμοκρασία. Με την ίδια λογική έχει γραφτεί ολόκληρος ο κώδικας για τις υπόλοιπες θερμοκρασίες.

Ο μικροελεγκτής είναι ρυθμισμένος να λειτουργεί στα 8MHz. Με τη βοήθεια της ταχύτητας του επεξεργαστή αλλά και λόγω του οπτικού φαινομένου POV, μπορούμε να αναβοσβήσουμε τόσο γρήγορα τα led, κρατώντας τα ουσιαστικά αναμμένα μόλις 1 ms, ώστε να έχουμε μια συσκευή απεικόνισης χωρίς τρεμοπαίξιμο (flickering).

#### 4.5 Επεξήγηση ADC (Analog to Digital Converter)

Ο μικροελεγκτής 18F4550 διαθέτει αναλογικές εισόδους. Πρακτικά αυτό σημαίνει ότι μπορούμε να του δώσουμε κάποια τάση (τάση αναφοράς) και εκείνος να την μετατρέψει σε μορφή επεξεργάσιμη προς εκείνον.

Αναλυτικότερα, ο μικροελεγκτής διαθέτει ADC λειτουργία με ανάλυση 10bits. Η ανάλυση υποδηλώνει πόσες φορές μπορεί να διαιρεθεί η τάση αναφοράς. Για 10 bit ανάλυση ADC, μπορεί να διαιρεθεί έως και 1024 ( $2^{10}$ ). Επομένως για τάση αναφοράς 5V, η ελάχιστη τάση που μπορεί να εντοπίσει είναι  $5/1024 = 4.8\text{mV}$ . Αυτό σημαίνει πως αυτά τα 4.8 mV στην αναλογική είσοδο θα εντοπιστούν ως 1 ενώ το 9.6mV θα εντοπιστεί ως 2 (10 σε δυαδική μορφή). Όμως δεν θα μπορούσε να εντοπίσει την διαφορά μεταξύ 1mV και 3mV γιατί και τα δύο θα τα εντόπιζε ως 0. Έτσι, συμπεραίνουμε πως όσο μεγαλύτερη η ανάλυση, τόσο μικρότερη διαφορά μπορεί να εντοπίσει.

Κατά τη χρήση αυτής της λειτουργίας, ο χρήστης πρέπει να δηλώσει τις τιμές  $V_{\text{ref-}}$  και  $V_{\text{ref+}}$ , που στη δικιά μας περίπτωση είναι 0V και 5V αντίστοιχα. Στην ουσία, ο μικροελεγκτής θα αναγνωρίσει αυτές τις τιμές ως 0 για 0V και 1023 για 5V. Έπειτα, ανάλογα με την ταχύτητα του επεξεργαστή, πρέπει να δηλώσει όλες τις μεταβλητές που απαιτούνται για την ορθή λειτουργία του.

Στην συγκεκριμένη περίπτωση, η συνάρτηση που χρησιμοποιήθηκε για την μετατροπή της θερμοκρασίας είναι:

$$\text{Θερμοκρασία χώρου} = (5 \text{ volts} * 100 * \text{τιμή αισθητήρα}) / 1024$$



## Κεφάλαιο 5<sup>ο</sup> Συμπεράσματα

## 5.1 Συμπεράσματα

Η κατασκευή ενός κύβου φωτοδιόδων απαιτεί καλή προετοιμασία και σωστό σχεδιασμό, μιας και η εξωτερική εμφάνιση του, είναι αυτή που θα κρίνει εάν θα τραβήξει την προσοχή του θεατή ή όχι. Η ιδιότητα τους αυτή, ώθησε τους επαγγελματίες να τους χρησιμοποιούν για διαφημιστικούς σκοπούς, αφού με τον κατάλληλο προγραμματισμό, μπορούν να προβάλλουν σε αυτούς οποιοδήποτε μήνυμα θελήσουν.

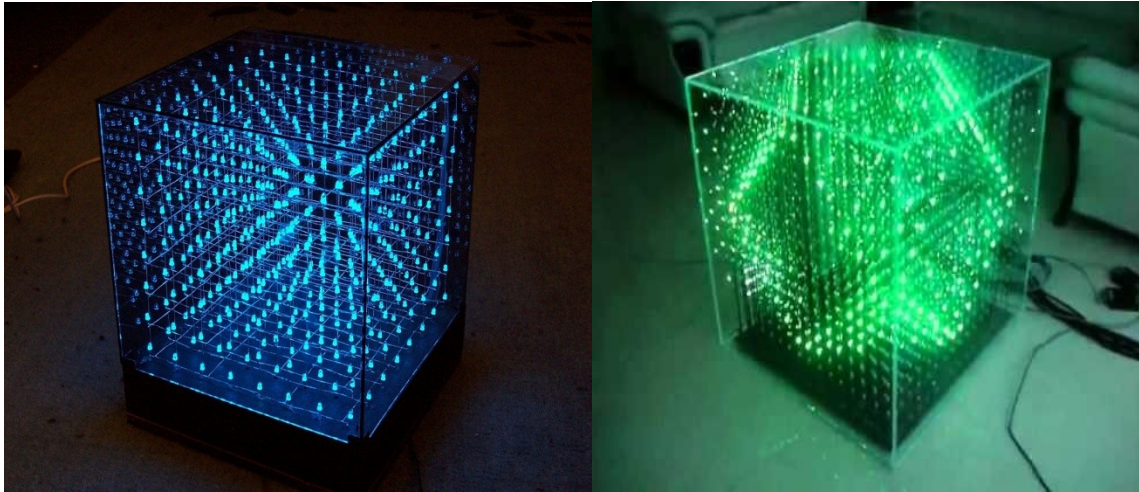
Κατά τη διάρκεια υλοποίησης μιας τέτοιας κατασκευής, είναι απαραίτητο να δοθεί προσοχή σε ορισμένα σημεία. Θα πρέπει να δοκιμαστούν πολλά υλικά που υπάρχουν διαθέσιμα στο εμπόριο, όπως η μεγάλη γκάμα φωτοδιόδων, ώστε να αποφασιστεί ποιο είναι το κατάλληλο για την κάθε περίπτωση. Επίσης, οι αποστάσεις που θα τοποθετηθούν τα led παίζουν καθοριστικό ρόλο ώστε να υπάρχει συμμετρία και πρόσβαση στο εσωτερικό του κύβου, δημιουργώντας έτσι την τρισδιάστατη ψευδαίσθηση.

Ταυτόχρονα, απαιτείται μεγάλη προσοχή στις κολλήσεις, αφού μπορεί να καεί κάποια δίοδος και η αντικατάστασή της είναι εξαιρετικά δύσκολη. Για αυτό το λόγο πρέπει να γίνονται συχνά έλεγχοι στα led. Τέλος, η συχνές δοκιμές σε κάθε στάδιο που ολοκληρώνεται, αποτελεί σημαντικό βήμα για την πρόληψη τυχών σφαλμάτων.

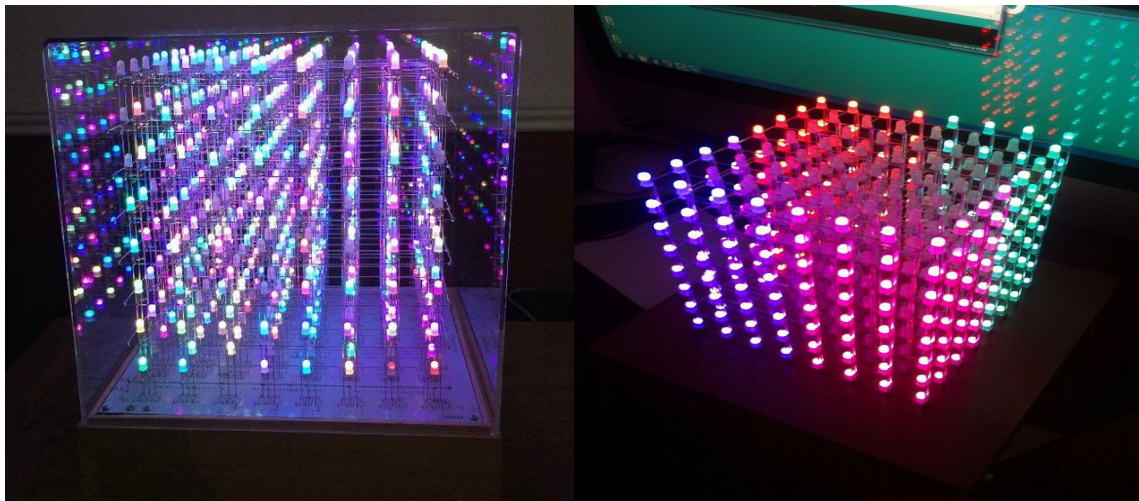
Σκοπός της συγκεκριμένης εργασίας, είναι να παρουσιάσει ότι ένας κύβος φωτοδιόδων μπορεί να συνδυαστεί με οποιοδήποτε αισθητήρα και να προσφέρει εκτός από την εξωτερική του εμφάνιση, σημαντικά στοιχεία, όπως για παράδειγμα η θερμοκρασία του χώρου στην παρούσα κατασκευή. Ουσιαστικά, ο συγκεκριμένος κύβος έχει την λειτουργία ενός θερμόμετρου.

## 5.2 Παρόμοιες κατασκευές LED κύβων

Παγκοσμίως έχουν υλοποιηθεί διάφορες εκδοχές ηλεκτρονικών LED κύβων. Από μικρές κατασκευές έως ολόκληρα κτήρια καλυμμένα με φωτοδιόδους. Μπορεί κανείς να βρει κύβους με μονόχρωμες φωτοδιόδους αλλά και πιο εντυπωσιακές με RGB LED. Παρακάτω παρουσιάζονται ορισμένα παραδείγματα.



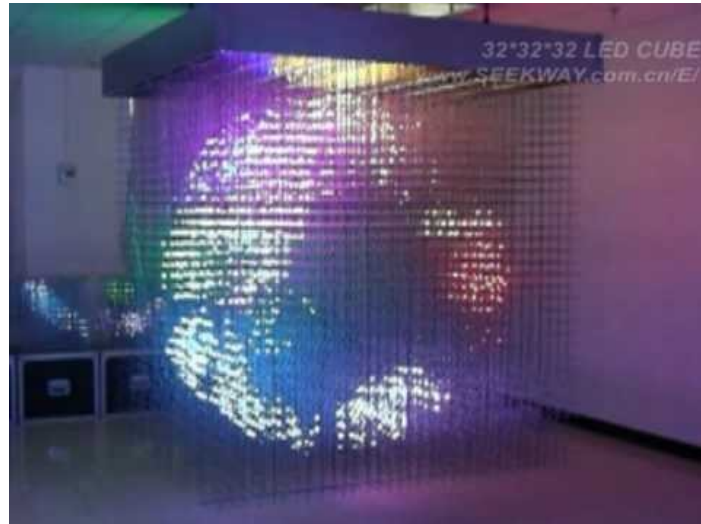
**Εικόνα 35-36: Μονοχρωματικός κύβος led[12]**



**Εικόνα 37-37.1: RGB LED κύβος[13]**

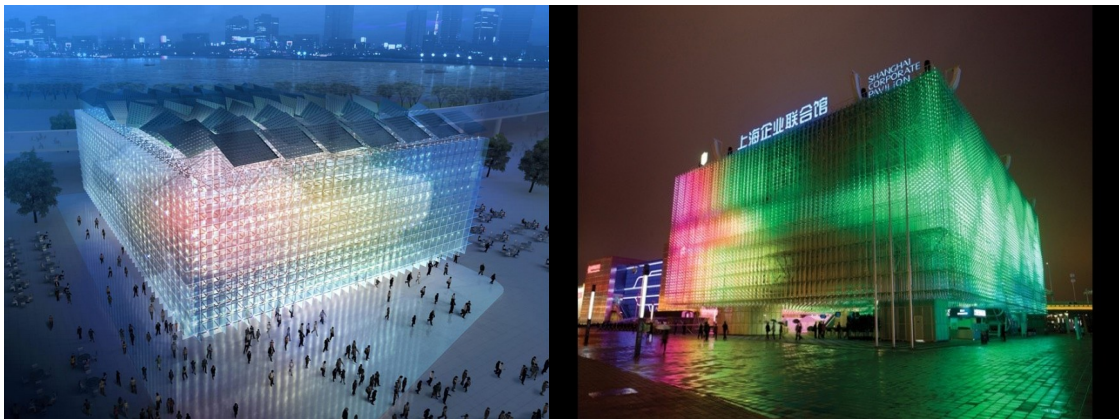
Ο βραβευμένος παγκοσμίως μεγαλύτερος led κύβος βρίσκεται στην Κίνα και είναι της εταιρείας China's Seekway Technology, Ltd.. Το μέγεθος του είναι  $32^3$  δηλαδή 32,768 LED. [14]





**Εικόνα 38: Μεγαλύτερος led κύβος παγκοσμίως**

Έμπνευση αυτής της πτυχιακής εργασίας, αποτέλεσε το κτήριο Dream cube που βρίσκεται στη Σαγκάη της Κίνας. Δημιουργήθηκε το 2010 στα πλαίσια της παγκόσμιας έκθεσης. Το κτίριο αυτό τόσο εξωτερικά όσο και εσωτερικά καλύπτεται από φωτοδιόδους και διαθέτει διάφορες αίθουσες σε κάθε μια από τις οποίες προσφέρεται και μια διαφορετική λειτουργία. Συγκεκριμένα, υπάρχει αίθουσα όπου με τη βοήθεια αισθητήρων, ανάλογα με την ένταση του χειροκροτήματος των ατόμων, η αίθουσα αλλάζει χρώμα. Επίσης σε κάποια άλλη αίθουσα, την ώρα που οι παρευρισκόμενοι χορεύουν, κάμερες καταγράφουν και προβάλλουν τις κινήσεις τους, με μορφή ολογράμματος, στην εξωτερική πλευρά του κτηρίου. [15]



**Εικόνα 39-40: Εξωτερική πλευρά κτηρίου**



**Εικόνα 41-42 : Εσωτερικό κτηρίου**



## Παραρτήματα

## Παράρτημα Α: Υλικά κατασκευής

Τα υλικά που χρησιμοποιήθηκαν στην κατασκευή:

<b>ΥΛΙΚΑ</b>	<b>ΠΟΣΟΤΗΤΑ</b>
Water cleat led 5mm	<b>64</b>
Αντίσταση 220 Ω ¼ W 5%	<b>16</b>
Αντίσταση 2.2 ΚΩ ¼ W 5%	<b>4</b>
Πυκνωτές κεραμικούς 0.1 μF	<b>2</b>
Τρανζίστορ BC338	<b>4</b>
Μικροελεγκτής 18F4550 I/P	<b>1</b>
40 PIN IC SOCKET	<b>1</b>
4 PIN FEMALE SOCKET	<b>1</b>
6 PIN MALE SCOKET	<b>1</b>
Βύσμα τροφοδοσίας	<b>1</b>
Κουμπί τροφοδοσίας	<b>1</b>
Ατσάλινο σύρμα	<b>10μέτρα</b>
Prototype board 22x10cm	<b>1</b>
Ατσάλινες βίδες στήριξης και παξιμάδια	<b>4+4</b>

## Παράρτημα Β: Κώδικας

// 18F4550 Lm35ledcube

#include <xc.h>

#define \_XTAL\_FREQ 8000000

#include<p18f4550.h>

/\*\*\*\*\*\* ρύθμιση παραμέτρων Pic \*\*\*\*\*/

#pragma config PLLDIV = 5 // Δεν έχει σημασία εφόσον χρησιμοποιείτε εσωτερικός ταλαντωτής και δεν χρησιμοποιείτε η θύρα usb

#pragma config CPUDIV = OSC1\_PLL2 // Δεν έχει σημασία εφόσον χρησιμοποιείτε εσωτερικός ταλαντωτής και δεν χρησιμοποιείτε η θύρα usb

#pragma config USBDIV = 2 // Δεν έχει σημασία εφόσον χρησιμοποιείτε εσωτερικός ταλαντωτής και δεν χρησιμοποιείτε η θύρα usb

// internal clock, pin #14 (RA6) as I/O pin, pin #13 unused,

#pragma config FOSC = INTOSCIO\_EC // Εσωτερικός ταλαντωτής, pin #14 (RA6) ως θύρα I/O, pin #13 δεν χρησιμοποιείτε

/\* υπόλοιπες ρυθμίσεις παραμέτρων με την βοήθεια της λειτουργία “set configuration bits” του MPLAB X IDE και επεξήγηση λειτουργίας\*/

#pragma config FCMEN = OFF // fail-safe clock monitor disabled

#pragma config IESO = OFF // internal / external osc switchover bit disabled

#pragma config PWRT = OFF // power-up timer disabled

#pragma config BOR = OFF // brown-out reset disabled in hardware & software

#pragma config BORV = 3 // brown-out reset voltage bits, does not matter since brown-out is disabled

#pragma config VREGEN = OFF // USB voltage regulator, would need to turn on if using internal USB voltage regulator for USB comm.

```

#pragma config WDT = OFF // watchdog timer disabled

#pragma config WDTPS = 32768 // watchdog timer postscale, does not
matter since watchdog timer is disabled

#pragma config CCP2MX = ON // use RC1 (pin #16) as CCP2 MUX (this is
the default pin for CCP2 MUX)

#pragma config PBADEN = OFF // RB0, RB1, RB2, RB3, & RB4
are configured as digital I/O on reset

#pragma config LPT1OSC = OFF // disable low-power option for timer 1
(timer 1 in regular mode)

#pragma config MCLRE = OFF // master clear disabled, pin #1
is for VPP and / or RE3 use

#pragma config STVREN = ON // stack full/underflow will
cause reset

#pragma config LVP = OFF // single-supply ICSP disabled

#pragma config ICPRT = OFF // in-circuit
debug/programming port (ICPORT) disabled, this feature is not available on 40 pin
DIP package

#pragma config XINST = OFF // instruction set extension and
indexed addressing mode disabled (this is the default setting)

#pragma config DEBUG = OFF // background debugger
disabled, RA6 & RB7 configured as general purpose I/O pins

#pragma config CP0 = OFF, CP1 = OFF, CP2 = OFF, CP3 = OFF //
code protection bits off

#pragma config CPB = OFF // boot block code protection off

#pragma config CPD = OFF // data EEPROM code
protection off

#pragma config WRT0 = OFF, WRT1 = OFF, WRT2 = OFF, WRT3 = OFF //
write protection bits off

#pragma config WRTC = OFF // config registers write
protection off

```

```

#pragma config WRTB = OFF // boot block is not write
protected

#pragma config WRTD = OFF // data EEPROM is not write
protected

#pragma config EBTR0 = OFF, EBTR1 = OFF, EBTR2 = OFF, EBTR3 = OFF //
table read protection bits off

#pragma config EBTRB = OFF // boot block table read
protection off

```

```

/*****Συναρτήσεις*****/

```

```

void ADC_Init(void); // Αρχικοποίηση ADC λειτουργίας(initialize the adc)

```

```

unsigned int Get_ADC_Result(void);

```

```

void Start_Conversion(void); //αρχή μετατροπής

```

```

/* ***** delay ***** */

```

```

void delay1sec(void)

```

```

{
    int i;

```

```

        for(i=0;i<100;i++)

```

```

        { __delay_ms(10);

```

```

            /* τέλος καθυστέρησης */
        } }

```

```

/* ***** Κυρίως πρόγραμμα ***** */

```

```

void main(void)

```

```

{

```



```
unsigned int adc_val,k;
unsigned long Voltage;

OSCCONbits.IRCF2 = 1; //set internal clock to 8MHZ
OSCCONbits.IRCF1 = 1;
OSCCONbits.IRCF0 = 1;

ADCON1bits.PCFG0 = 1; // SET ALL AN PINS TO DIGITAL I/O
ADCON1bits.PCFG1 = 1;
ADCON1bits.PCFG2 = 1;
ADCON1bits.PCFG3 = 1;

// SET PORTS TO OUTPUT
TRISBbits.TRISB0 = 0; // set ports to output
TRISBbits.TRISB1 = 0;
TRISBbits.TRISB2 = 0;
TRISBbits.TRISB3 = 0;
TRISBbits.TRISB4 = 0;
TRISBbits.TRISB5 = 0;
TRISBbits.TRISB6 = 0;

TRISDbits.TRISD0 = 0; // set ports to output
TRISDbits.TRISD1 = 0;
TRISDbits.TRISD2 = 0;
TRISDbits.TRISD3 = 0;
TRISDbits.TRISD4 = 0;
TRISDbits.TRISD5 = 0;
```

```
TRISDbits.TRISD6 = 0;
TRISDbits.TRISD7 = 0;

TRISCbits.TRISC0 = 0; // set ports to ouput
TRISCbits.TRISC1 = 0;
TRISCbits.TRISC2 = 0;
TRISCbits.TRISC6 = 0;
TRISCbits.TRISC7 = 0;
```

```
//INIT OUTPUT PINS SE O
```

```
PORTBbits.RB0 = 0; //init ouput pins to off
PORTBbits.RB1 = 0;
PORTBbits.RB2 = 0;
PORTBbits.RB3 = 0;
PORTBbits.RB4 = 0;
PORTBbits.RB5 = 0;
PORTBbits.RB6 = 0;
```

```
PORTDbits.RD0 = 0; //init ouput pins to off
PORTDbits.RD1 = 0;
PORTDbits.RD2 = 0;
PORTDbits.RD3 = 0;
PORTDbits.RD4 = 0;
PORTDbits.RD5 = 0;
PORTDbits.RD6 = 0;
PORTDbits.RD7 = 0;
```

**PORTCbits.RC0 = 0; //init ouput pins to off**

**PORTCbits.RC1 = 0;**

**PORTCbits.RC2 = 0;**

**PORTCbits.RC6 = 0;**

**PORTCbits.RC7 = 0;**

**TRISAbits.RA0 = 1;       //ra0 Είναι το αισθητήριο θερμοκρασίας**

*/\*\*\*\*\* Πρόγραμμα ελέγχου led\*\*\*\*\*/*

**PORTBbits.RB6 = 1;**

**PORTBbits.RB5 = 1;**

**PORTBbits.RB4 = 1;**

**PORTBbits.RB3 = 1;**

**PORTBbits.RB2 = 1;**

**PORTBbits.RB1 = 1;**

**PORTBbits.RB0 = 1;**

**PORTDbits.RD0 = 1;**

**PORTDbits.RD1 = 1;**

**PORTDbits.RD2 = 1;**

**PORTDbits.RD3 = 1;**

**PORTDbits.RD4 = 1;**

**PORTDbits.RD5 = 1;**

**PORTDbits.RD6 = 1;**

**PORTDbits.RD7 = 1;**

**PORTCbits.RC0 = 1;**

**PORTCbits.RC1 = 1;**

**PORTCbits.RC2 = 1;**

**PORTCbits.RC6 = 1;**

**PORTCbits.RC7 = 1;**

**delay1sec();**

**PORTBbits.RB6 = 0;**

**PORTBbits.RB5 = 0;**

**PORTBbits.RB4 = 0;**

**PORTBbits.RB3 = 0;**

**PORTBbits.RB2 = 0;**

**PORTBbits.RB1 = 0;**

**PORTBbits.RB0 = 0;**

**PORTDbits.RD0 = 0;**

**PORTDbits.RD1 = 0;**

**PORTDbits.RD2 = 0;**

**PORTDbits.RD3 = 0;**

**PORTDbits.RD4 = 0;**

**PORTDbits.RD5 = 0;**

**PORTDbits.RD6 = 0;**

**PORTDbits.RD7 = 0;**

**PORTCbits.RC0 = 0;**

```

PORTCbits.RC1 = 0;

PORTCbits.RC2 = 0;

PORTCbits.RC6 = 0;

PORTCbits.RC7 = 0;

delay1sec();

/*****Τέλος πρόγραμμα ελέγχου led*****/

while(1)
{
    ADC_Init();          //Σετάρισμα adc και ενεργοποίηση
    Start_Conversion(); //Μετατροπή
    adc_val= Get_ADC_Result(); //Αποτέλεσμα
    Voltage = (long) adc_val*500.0;
    adc_val = Voltage /1024.0;
    k = adc_val; //Αποθήκευση σε μεταβλητή

/***** Θερμοκρασίες*****/

// 20-30 βαθμούς κελσίου

    if (k>20){

        PORTBbits.RB3 = 1;

        PORTDbits.RD0 = 1;

        PORTDbits.RD6 = 1;

        PORTDbits.RD7 = 1;

        PORTDbits.RD3 = 1;

        __delay_ms(1);

```

```

        PORTBbits.RB3 = 0;
        PORTDbits.RD0 = 0;
        PORTDbits.RD6 = 0;
        PORTDbits.RD7 = 0;
        PORTDbits.RD3 = 0;
    }
    if (k>22){
        PORTBbits.RB3 = 1;
        PORTCbits.RC2 = 1;
        PORTDbits.RD6 = 1;
        PORTDbits.RD7 = 1;
        PORTDbits.RD3 = 1;
        __delay_ms(1);
        PORTBbits.RB3 = 0;
        PORTCbits.RC2 = 0;
        PORTDbits.RD6 = 0;
        PORTDbits.RD7 = 0;
        PORTDbits.RD3 = 0;
    }
    if (k>24){
        PORTBbits.RB3 = 1;
        PORTCbits.RC1 = 1;
        PORTDbits.RD6 = 1;
        PORTDbits.RD7 = 1;
        PORTDbits.RD3 = 1;
        __delay_ms(1);
        PORTBbits.RB3 = 0;

```

```

        PORTCbits.RC1 = 0;
        PORTDbits.RD6 = 0;
        PORTDbits.RD7 = 0;
        PORTDbits.RD3 = 0;
    }
    if (k>26){
        PORTBbits.RB3 = 1;
        PORTCbits.RC0 = 1;
        PORTDbits.RD6 = 1;
        PORTDbits.RD7 = 1;
        PORTDbits.RD3 = 1;
        __delay_ms(1);
        PORTBbits.RB3 = 0;
        PORTCbits.RC0 = 0;
        PORTDbits.RD6 = 0;
        PORTDbits.RD7 = 0;
        PORTDbits.RD3 = 0;
    }
    if (k>28){
        PORTBbits.RB4 = 1;
        PORTBbits.RB0 = 1;
        PORTDbits.RD5 = 1;
        PORTDbits.RD2 = 1;
        PORTDbits.RD0 = 1;
        __delay_ms(1);
        PORTBbits.RB4 = 0;
        PORTBbits.RB0 = 0;
    }

```

```
    PORTDbits.RD5 = 0;
    PORTDbits.RD2 = 0;
    PORTDbits.RD0 = 0;
}
// 30-40 βαθμούς Κελσίου
```

```
if (k>30){
```

```
    PORTBbits.RB4 = 1;
    PORTBbits.RB0 = 1;
    PORTDbits.RD5 = 1;
    PORTDbits.RD2 = 1;
    PORTCbits.RC2 = 1;
    __delay_ms(1);
    PORTBbits.RB4 = 0;
    PORTBbits.RB0 = 0;
    PORTDbits.RD5 = 0;
    PORTDbits.RD2 = 0;
    PORTCbits.RC2 = 0;
}
```

```
if (k >32){
```

```
    PORTBbits.RB4 = 1;
    PORTBbits.RB0 = 1;
    PORTDbits.RD5 = 1;
    PORTDbits.RD2 = 1;
    PORTCbits.RC1 = 1;
    __delay_ms(1);
    PORTBbits.RB4 = 0;
```



```

    PORTBbits.RB0 = 0;
    PORTDbits.RD5 = 0;
    PORTDbits.RD2 = 0;
    PORTCbits.RC1 = 0;
}
if (k>34){
    PORTBbits.RB4 = 1;
    PORTBbits.RB0 = 1;
    PORTDbits.RD5 = 1;
    PORTDbits.RD2 = 1;
    PORTCbits.RC0 = 1;
    __delay_ms(1);
    PORTBbits.RB4 = 0;
    PORTBbits.RB0 = 0;
    PORTDbits.RD5 = 0;
    PORTDbits.RD2 = 0;
    PORTCbits.RC0 = 0;
}
if (k>36){
    PORTBbits.RB5 = 1;
    PORTBbits.RB1 = 1;
    PORTDbits.RD4 = 1;
    PORTCbits.RC6 = 1;
    PORTDbits.RD0 = 1;
    __delay_ms(1);
    PORTBbits.RB5 = 0;
    PORTBbits.RB1 = 0;

```

```

        PORTDbits.RD4 = 0;
        PORTCbits.RC6 = 0;
        PORTDbits.RD0 = 0;
    }
    if (k>38){
        PORTBbits.RB5 = 1;
        PORTBbits.RB1 = 1;
        PORTDbits.RD4 = 1;
        PORTCbits.RC6 = 1;
        PORTCbits.RC2 = 1;
        __delay_ms(1);
        PORTBbits.RB5 = 0;
        PORTBbits.RB1 = 0;
        PORTDbits.RD4 = 0;
        PORTCbits.RC6 = 0;
        PORTCbits.RC2 = 0;
    }
    // 40-50 βαθμούς Κελσίου
    if (k>40){
        PORTBbits.RB5 = 1;
        PORTBbits.RB1 = 1;
        PORTDbits.RD4 = 1;
        PORTCbits.RC6 = 1;
        PORTCbits.RC1 = 1;
        __delay_ms(1);
        PORTBbits.RB5 = 0;
        PORTBbits.RB1 = 0;
    }

```

```

        PORTDbits.RD4 = 0;
        PORTCbits.RC6 = 0;
        PORTCbits.RC1 = 0;
    }
if (k>42){
    PORTBbits.RB5 = 1;
    PORTBbits.RB1 = 1;
    PORTDbits.RD4 = 1;
    PORTCbits.RC6 = 1;
    PORTCbits.RC0 = 1;
    __delay_ms(1);
    PORTBbits.RB5 = 0;
    PORTBbits.RB1 = 0;
    PORTDbits.RD4 = 0;
    PORTCbits.RC6 = 0;
    PORTCbits.RC0 = 0;
}
if (k >44){
    PORTBbits.RB6 = 1;
    PORTBbits.RB2 = 1;
    PORTCbits.RC7 = 1;
    PORTDbits.RD1 = 1;
    PORTDbits.RD0 = 1;
    __delay_ms(1);
    PORTBbits.RB6 = 0;
    PORTBbits.RB2 = 0;
    PORTCbits.RC7 = 0;
}

```

```

        PORTDbits.RD1 = 0;
        PORTDbits.RD0 = 0;
    }
    if (k >46){
        PORTBbits.RB6 = 1;
        PORTBbits.RB2 = 1;
        PORTCbits.RC7 = 1;
        PORTDbits.RD1 = 1;
        PORTCbits.RC2 = 1;
        __delay_ms(1);
        PORTBbits.RB6 = 0;
        PORTBbits.RB2 = 0;
        PORTCbits.RC7 = 0;
        PORTDbits.RD1 = 0;
        PORTCbits.RC2 = 0;
    }
    if (k >48){
        PORTBbits.RB6 = 1;
        PORTBbits.RB2 = 1;
        PORTCbits.RC7 = 1;
        PORTDbits.RD1 = 1;
        PORTCbits.RC1 = 1;
        __delay_ms(1);
        PORTBbits.RB6 = 0;
        PORTBbits.RB2 = 0;
        PORTCbits.RC7 = 0;
        PORTDbits.RD1 = 0;
    }

```

```

        PORTCbits.RC1 = 0;
    }
    if (k >50){
        PORTBbits.RB6 = 1;
        PORTBbits.RB2 = 1;
        PORTCbits.RC7 = 1;
        PORTDbits.RD1 = 1;
        PORTCbits.RC0 = 1;
        __delay_ms(1);
        PORTBbits.RB6 = 0;
        PORTBbits.RB2 = 0;
        PORTCbits.RC7 = 0;
        PORTDbits.RD1 = 0;
        PORTCbits.RC0 = 0;
    }
}

```

*//Ορισμοί Συναρτήσεων*

```

void ADC_Init()
{
    ADCON0=0b00000000; //A/D Module is OFF and Channel 0 is selected
    ADCON1=0b00001110; // Reference as VDD & VSS, AN0 set as analog pins
    ADCON2=0b10010010; // Result is right Justified
    //Acquisition Time 2TAD
    //ADC Clk FOSC/64
    ADCON0bits.ADON=1; //Turn ON ADC module

```

```
    __delay_ms(1);  
}
```

```
void Start_Conversion()  
{  
    ADCON0bits.GO=1;  
  
}
```

```
unsigned int Get_ADC_Result()  
{  
    unsigned int ADC_Result=0;  
    while(ADCON0bits.DONE);  
    ADC_Result=ADRESL;  
    ADC_Result|=((unsigned int)ADRESH) << 8;  
    return ADC_Result;  
}
```

*/\* Τέλος κώδικα \*/*



# Αναφορές

- [1] Wikipedia, ( 2018, Μάρτιος). Persistence of vision. [στο διαδίκτυο] Διαθέσιμο στο: [https://en.wikipedia.org/wiki/Persistence\\_of\\_vision](https://en.wikipedia.org/wiki/Persistence_of_vision)
- [2] Coolweb (2014, Φεβρουάριος). Δίοδοι. [στο διαδίκτυο] Διαθέσιμο στο: <http://coolweb.gr/fotodiodoi-diodoi-leds/>
- [3]Tutorial cytron, (2012, Αύγουστος). Construct led cube. [στο διαδίκτυο] Διαθέσιμο στο: <https://tutorial.cytron.io/2012/08/29/construct-a-8x8x8-led-cube/>
- [4] Jaycon Systems (2015, Νοέμβριος). Getting started with a temperature sensor. [στο διαδίκτυο] Διαθέσιμο στο: <https://www.jayconsystems.com/tutorials/Temperature-sensor/>
- [5] Microchip Technology(2006). Microchip-PIC 18F4550 Microcontroller [στο διαδίκτυο] Διαθέσιμο στο: <http://www.microchip.com/datasheet/PIC18F4550>
- [6] Texas Instruments, (2017 Δεκέμβριος). LM35 Precision Centigrade Temperature Sensors. [στο διαδίκτυο] Διαθέσιμο στο: <http://www.ti.com/lit/ds/symlink/lm35.pdf>
- [7] Microchip Technology(2006). Microchip-PICkit 3 In-Circuit Debugger[στο διαδίκτυο] Διαθέσιμο στο: <http://www.microchip.com/Developmenttools/ProductDetails.aspx?PartNO=PG164130>
- [8] Γ. Κόγιας,(2005). Αρχιτεκτονική, οργάνωση και προγραμματισμός μικροϋπολογιστών, Σύγχρονη Εκδοτική.
- [9] Stephen G. Kochan (2010). Προγραμματισμός με τη C, Τρίτη έκδοση. Αθήνα: Παπασωτηρίου Εκδόσεις
- [10]Γ. Σ. Τσελίκη - Ν. Δ. Τσελικά(2010). “ C: από τη Θεωρία στην Εφαρμογή ”, Αθήνα .
- [11] Πεκμεστζή Κιαμάλ (2009). "ΣΥΣΤΗΜΑΤΑ ΜΙΚΡΟΎΠΟΛΟΓΙΣΤΩΝ ΙΙ - Μικροελεγκτές AVR και PIC" - Εκδόσεις Συμμετρία.
- [12] Leyanda DIY PROJECTS.de (2015). 8X8X8 LED CUBE [στο διαδίκτυο] Διαθέσιμο στο: [http://leyanda.de/index.php?option=com\\_content&view=article&id=3](http://leyanda.de/index.php?option=com_content&view=article&id=3)



[13] Instructables.com (2015 Δεκέμβριος). Arduino mega 8x8x8 rgb led cube [ στο διαδίκτυο] Διαθέσιμο στο: <http://www.instructables.com/id/Arduino-Mega-8x8x8-RGB-LED-Cube/>

[14]Makezine (2012 Νοέμβριος). Biggest LED cube [στο διαδίκτυο] Διαθέσιμο στο: <https://makezine.com/2012/11/18/worlds-biggest-led-cube/>

[15]segd.org (2010). Dream cube 2010 world expo pavilion [στο διαδίκτυο] Διαθέσιμο στο: <https://segd.org/dream-cube-2010-world-expo-pavilion>