



ΜΕΤΑΠΤΥΧΙΑΚΗ ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

---

**Δημιουργία πρωτότυπου  
εκπαιδευτικού παιχνιδιού για την  
εξοικείωση με την έννοια των τριών  
διαστάσεων**

---

*Συγγραφέας:*  
Νικολαΐδης Ιωάννης

*Επιβλέπων:*  
Δρ. Μπεχτσής Δημήτριος

*Στα πλαίσια του μεταπτυχιακού προγράμματος  
STEM και νέες τεχνολογίες στην εκπαίδευση*

*στο*

*Τμήμα Μηχανικών Παραγωγής και Διοίκησης*

16 Οκτωβρίου 2023



# Περιεχόμενα

<b>1</b>	<b>Πιθανές κατασκευές</b>	<b>3</b>
1.1	Εισαγωγή . . . . .	3
1.1.1	Βασικά ερωτήματα . . . . .	3
1.1.2	Ανάλυση . . . . .	3
1.1.3	Συνοπτική παρουσίαση λύσεων . . . . .	4
1.2	Μονοκονδυλιά . . . . .	4
1.2.1	Υλικά θεμελίου . . . . .	4
1.2.2	Προγραμματισμός θεμελίου . . . . .	4
1.2.3	Υλικά κυβιδίων . . . . .	4
1.2.4	Εικόνες . . . . .	5
1.3	Αριθμημένα κυβίδια . . . . .	5
1.3.1	Υλικά θεμελίου . . . . .	5
1.3.2	Προγραμματισμός θεμελίου . . . . .	5
1.3.3	Υλικά κυβιδίων . . . . .	5
1.4	Προσανατολισμένα κυβίδια . . . . .	5
1.4.1	Υλικά θεμελίου . . . . .	6
1.4.2	Προγραμματισμός θεμελίου . . . . .	6
1.4.3	Υλικά κυβιδίων . . . . .	6
1.4.4	Υλοποιητέα λογική κυβιδίων . . . . .	7
1.4.5	Εικόνες . . . . .	8
1.5	Συμμετρικά κυβίδια . . . . .	8
1.5.1	Υλικά θεμελίου . . . . .	9
1.5.2	Προγραμματισμός θεμελίου . . . . .	9
1.5.3	Υλικά κυβιδίων . . . . .	9
1.5.4	Προγραμματισμός κυβιδίων . . . . .	9
<b>2</b>	<b>Κατασκευαστικές λεπτομέρειες</b>	<b>11</b>
2.1	Εισαγωγή . . . . .	11
2.2	Συμμετρία εδρών . . . . .	11
2.2.1	Κατασκευή συμμετρικών συζευκτών . . . . .	11
2.2.2	Μηχανικοί συζευκτές . . . . .	12
2.2.3	Ηλεκτρικοί συζευκτές . . . . .	13
2.2.4	Μείωση πλήθους ακίδων . . . . .	14
	Παρουσίαση λύσης . . . . .	14
2.3	Δομικό υλικό εκάστοτε έδρας . . . . .	14
2.4	Συμμετρία κυβιδίων . . . . .	15
2.5	Συνδεσμολογία . . . . .	16
2.5.1	Συνδεσμολογία ακροδεκτών δεδομένων . . . . .	17
	Επιλογή τιμών αντιστατών . . . . .	17
2.6	Κυκλωματικό διάγραμμα κυβιδίου . . . . .	19
2.6.1	Επιλογή μικροελεγκτή . . . . .	19
2.7	Δομή θεμελίου . . . . .	19

<b>3</b>	<b>Προγραμματισμός συμμετρικών κυβιδίων</b>	<b>23</b>
3.1	Κυβικές περιστροφές . . . . .	23
3.1.1	Καταμέτρηση προσανατολισμών . . . . .	23
3.1.2	Οκταεδρική συμμετρία . . . . .	23
3.1.3	Προγραμματισμός οκταεδρικής συμμετρίας . . . . .	24
3.2	Χωρικές συντεταγμένες και προσανατολισμός . . . . .	25
3.3	Ανίχνευση οδηγίων . . . . .	25
3.4	Υλοποίηση του κώδικα . . . . .	26
3.4.1	Επιλογή γλώσσας . . . . .	26
3.4.2	Οργάνωση κώδικα . . . . .	26
3.4.3	Λεπτομέρειες υλοποίησης . . . . .	27
	Ιδιοκτησία και δανεισμός . . . . .	27
	Κτήση περιφερειακών . . . . .	28
3.4.4	Πρωτόκολλο επικοινωνίας . . . . .	28
3.4.5	Χρήση της κάθε μνήμης . . . . .	28
3.5	Προγραμματισμός θεμελίου . . . . .	30
3.6	Διαγράμματα ροής . . . . .	30
<b>4</b>	<b>Κατασκευή, πειράματα, αξιολόγηση</b>	<b>37</b>
4.1	Αρχικές δοκιμές . . . . .	37
4.2	Κατασκευαστικές λεπτομέρειες . . . . .	37
4.2.1	Εσωτερική πλευρά πλακετών . . . . .	37
4.3	Πρώτα πειράματα . . . . .	38
4.4	Τελική αποτίμηση επιλογών . . . . .	40
4.4.1	Συμμετρία κυβιδίων . . . . .	40
4.4.2	Κατασκευαστικές επιλογές . . . . .	40
	Πλακέτες . . . . .	40
	Μαγνήτες . . . . .	41
	Ακροδέκτες με ελατήρια (Pogo pins) . . . . .	41
	Δικτύωμα διαίρεσης τάσης . . . . .	42
4.4.3	Προγραμματισμός . . . . .	42
	Γλώσσα προγραμματισμού . . . . .	42
	Πρωτόκολλο επικοινωνίας . . . . .	44
<b>5</b>	<b>Μελλοντικές βλέψεις</b>	<b>45</b>
5.1	Κόστος κατασκευής . . . . .	45
5.1.1	Μικροελεγκτές . . . . .	45
5.1.2	Μαγνήτες . . . . .	45
5.1.3	Ακίδες . . . . .	46
5.1.4	Πλακέτες και πλαστικά κελύφη . . . . .	46
5.1.5	Σύννοψη . . . . .	46
5.2	Πιθανές βελτιώσεις . . . . .	46
5.2.1	Αναλογική ανάγνωση της γραμμής χωρίς περιορισμούς . . . . .	46
5.2.2	Βελτιώσεις δικτύωματος διαίρεσης τάσης . . . . .	47
	Επιτάχυνση επικοινωνίας . . . . .	47
	Αναγνώριση αποσύνδεσης . . . . .	47
	Διαχωρισμός 9 καταστάσεων . . . . .	47
5.2.3	Διακοπές (interrupts) . . . . .	50
5.2.4	Πλαστικό περίβλημα . . . . .	50
5.2.5	Ανίχνευση ολοκλήρωσης κατασκευής . . . . .	50
5.2.6	Φτηνότεροι μικροελεγκτές . . . . .	51

Μικροελεγκτές ATtiny402 . . . . .	51
Απο/πολυπλέκτες . . . . .	51
Μνήμη προγράμματος (Flash) . . . . .	51
Μνήμη SRAM . . . . .	52
5.2.7 Έλεγχος τροφοδοσίας κυβιδίων . . . . .	52
5.2.8 Μαζική παραγωγή . . . . .	53
<b>A΄ Φωτογραφίες άλλων λύσεων</b>	<b>55</b>
A΄.1 Μονοκονδυλιά . . . . .	55
A΄.2 Προσανατολισμένα . . . . .	56
A΄.3 Συμμετρικά με διάτρητες πλακέτες . . . . .	59
A΄.4 Διάφορα . . . . .	60



# Εισαγωγή

## Ανάλυση στόχων

Στόχος της παρούσας διπλωματικής εργασίας είναι η δημιουργία ενός παιχνιδιού το οποίο θα απαιτεί από ένα παιδί να προσανατολίζεται νοερά στις τρεις χωρικές διαστάσεις. Τα κυριότερα χαρακτηριστικά του παιχνιδιού, όπως εξ αρχής τα συλλάβαμε, ήταν αρκετά καθορισμένα· παρ' όλ' αυτά, υπήρξαν και αρκετές σχεδιαστικές αποφάσεις που έπρεπε να ληφθούν κατά τη διάρκεια της διπλωματικής.

Το πρώτο μας βήμα, λοιπόν, ήταν ο καθορισμός των θεμελιωδέστερων χαρακτηριστικών του παιχνιδιού ως εξής:

**Αριθμός παικτών:** Το παιχνίδι κατά βάση θα προορίζεται για χρήση από έναν παίκτη, καθώς ουσιαστικά θα πρόκειται για ένα γρίφο τον οποίο το παιδί θα πρέπει να επιλύσει. Ενδεχόμενες παραλλαγές για περισσότερους παίκτες θα επαφίενται στη δημιουργικότητα του εκάστοτε γονέα/διδασκάλου.

**Απευθυνόμενη ηλικιακή ομάδα:** τα παιδιά δημοτικού, ήτοι ηλικίας περίπου 6-12 ετών. Πάντως, οι ακριβότερες λύσεις που θα παραθέσουμε τείνουν να είναι αρκετά εύκολες ώστε να προσφέρονται και για νήπια.

**Επίλυση παιχνιδιού:** Το παιχνίδι θα επιλύεται δημιουργώντας ένα τρισδιάστατο σχήμα, με τη συναρμολόγηση καταλλίλων δομικών στοιχείων, τα οποία εφ' εξής θα ονομάζονται για συντομία «κυβίδια». Αυτά θα έχουν μάλλον τη μορφή κύβων, αλλά η καθ' αυτή κατασκευή τους αποτελεί σχεδιαστική απόφαση που θ' αναλυθεί αργότερα.

**Διαδικασία επίλυσης:** Η επίλυση θα πραγματοποιείται δίνοντας στο παιδί οδηγίες βήμα-προς-βήμα για το πού θα πρέπει κάθε φορά να τοποθετήσει το νέο κυβίδιο, ξεκινώντας κάθε φορά από μία πάγια βάση. Η ακριβής διαδικασία με την οποία θα δίνονται στο παιδί οι οδηγίες αποτελεί μία ακόμα σχεδιαστική απόφαση που θ' αναλυθεί αργότερα.

**Στήριξη κατασκευής:** Σε όλες τις λύσεις που σκεφτήκαμε, υπάρχει μία κεντρική βάση (εφ' εξής «θεμέλιο») πάνω στην οποία θα στηρίζεται όλη η κατασκευή. Αυτή η κεντρική βάση θα έχει και ένα μικροελεγκτή, ο οποίος θα βοηθά στην παρουσίαση των οδηγιών στον παίκτη. Προφανώς, όλες οι κατευθύνσεις που θα δίδονται στον παίκτη θα είναι προσανατολισμένες με βάση το θεμέλιο.

Έχοντας εδραιώσει τα παραπάνω, το επόμενο μας βήμα ήταν η επεξήγηση των σχεδιαστικών αποφάσεων που λήφθηκαν για να δοθεί στο παιχνίδι συγκεκριμένη μορφή.

## Περαιτέρω αναγκαίες αποφάσεις

Τα θεμελιώδη χαρακτηριστικά που περιγράψαμε ως τώρα αφορούν, κατά βάση, τη γενική σύλληψη του παιχνιδιού. Κανένα από αυτά δεν επηρεάζει με άμεσο τρόπο το κόστος και την πολυπλοκότητα κατασκευής του τελικού προϊόντος. Τα χαρακτηριστικά αυτά αναλύονται σε ξεχωριστό κεφάλαιο, καθώς όλες οι υπόλοιπες επιλογές εξαρτώνται από το ποια επίπεδα κόστους και της πολυπλοκότητας θεωρούνται αποδεκτά.

## Μορφοποίηση ηλεκτρονικής μορφής του τεύχους

Σε περίπτωση που η ανάγνωση του τεύχους γίνεται ηλεκτρονικά, τονίζουμε τα εξής:

**Υπερσύνδεσμοι:** Οτιδήποτε είναι μέσα σε έγχρωμο πλαίσιο αποτελεί υπερσύνδεσμο. Με πράσινο χρώμα είναι οι υπερσύνδεσμοι εσωτερικά του αρχείου, ενώ με κυανό όσοι είναι εξωτερικά. Ένα παράδειγμα δύο τέτοιων υπερσυνδέσμων δίπλα-δίπλα βρίσκεται στην ενότητα 5.1.

Σημαντική σημείωση: Τα έγχρωμα πλαίσια δεν τυπώνονται!

**Σχέδια:** Τα περισσότερα σχέδια (όχι οι φωτογραφίες) είναι σε μορφή SVG, και άρα μπορεί κανείς να κάνει αυθαίρετα πολύ zoom. Αυτό εξυπηρετεί ιδίως στα κυκλώματα, για να βλέπει κανείς λεπτομέρειες συνδέσεων.



## Κεφάλαιο 1

# Πιθανές κατασκευές

### 1.1 Εισαγωγή

#### 1.1.1 Βασικά ερωτήματα

Τα βασικότερα χαρακτηριστικά που διαχωρίζουν τις πιθανές λύσεις που σκεφτήκαμε είναι, αφ' ενός ο βαθμός διαδραστικότητας, αφ' ετέρου η πολυπλοκότητα των πιθανών σχημάτων. Στις πιο απλές, η διαδραστικότητα είναι ουσιαστικά ανύπαρκτη, και η γκάμα των σχημάτων που μπορούν να δημιουργηθούν είναι πολύ περιορισμένη. Στις πιο περίπλοκες, το παιχνίδι είναι 100% διαδραστικό, και μπορεί να δημιουργηθεί οποιοδήποτε σχήμα ανεξαρτήτως πόσο περίπλοκο.

Δευτερεύοντα χαρακτηριστικά είναι κυρίως δύο. Το πρώτο είναι η πολυπλοκότητα κατασκευής, ξεκινώντας από απλά κομμάτια πλαστικό και καταλήγοντας σε ενσωματωμένους μικροελεγκτές. Το δεύτερο είναι η ευκολία συναρμολόγησης, ξεκινώντας από κυβίδια που πρέπει να έχουν συγκεκριμένο προσανατολισμό και σειρά συναρμολόγησης, και καταλήγοντας σε αυθαίρετο προσανατολισμό και σειρά.

Σημειώτεον ότι η διαδραστικότητα της κάθε λύσης είναι σε μεγάλο βαθμό το πιο σημαντικό χαρακτηριστικό της: αυτή καθορίζει την πολυπλοκότητα της κατασκευής, και άρα και το χρηματικό κόστος. Επίσης, καθορίζει και το βαθμό συμμετοχής που χρειάζεται από το γονέα/διδάσκαλο που θα επιβλέπει: οι απλές λύσεις χρειάζονται συνεχή και άμεση συμμετοχή από τον επιβλέποντα ενήλικα, ενώ οι πιο πολύπλοκες λύσεις μπορούν να χρησιμοποιηθούν και χωρίς καν την παρουσία του.

#### 1.1.2 Ανάλυση

Στον πίνακα που ακολουθεί, το κάθε ένα απ' τα προαναφερθέντα χαρακτηριστικά ποσοτικοποιείται με μία λέξη-κλειδί. Ιδού οι επικεφαλίδες που χρησιμοποιήθηκαν σε κάθε στήλη, μαζί με τις λέξεις-κλειδιά που τις ποσοτικοποιούν:

**Επικοινωνία:** Η διαδραστικότητα του παιχνιδιού εξαρτάται ευθέως από το αν και κατά πόσο μπορούν τα κυβίδια να επικοινωνούν με το θεμέλιο. Οι πιθανότητες είναι: Ουδεμία, αν τα κυβίδια είναι απλό πλαστικό· Μονόδρομη, αν μπορούν να δεχτούν σήματα αλλά όχι να στείλουν· Αμφίδρομη, αν μπορούν και να δέχονται και να στέλνουν. Αυτό καθορίζει και τα απαραίτητα υλικά για το κάθε κυβίδιο.

**Διακλαδώσεις:** Όχι, αν πρέπει τα κυβίδια να σχηματίζουν μία συνεχόμενη αλυσίδα. Ναι, αν μπορεί να δημιουργηθεί οποιοδήποτε σχήμα.

**Προσανατολισμός:** Όχι, αν τα κυβίδια έχουν περιστροφική συμμετρία. Ναι, αν πρέπει να τοποθετηθούν με συγκεκριμένο προσανατολισμό.

**Αρίθμηση:** Όχι, αν τα κυβίδια μπορούν να τοποθετηθούν με αυθαίρετη σειρά.  
Ναι, αν το καθ' ένα έχει κάποιον αύξοντα αριθμό ο οποίος καθορίζει και τη σειρά συναρμολόγησής τους.

Στις λύσεις που θα παρουσιάσουμε, για συντομία και ευκολότερη αναφορά, θα δώσουμε μονολεκτικά ονόματα: **Μονοκονδυλιά**, **Αρίθμηση**, **Προσανατολισμένα**, και **Συμμετρικά**.

### 1.1.3 Συνοπτική παρουσίαση λύσεων

	Μονοκονδυλιά	Αρίθμηση	Προσανατολισμένα	Συμμετρικά
Επικοινωνία	<b>Ουδεμία</b>	<b>Ουδεμία</b>	<b>Μονόδρομη</b>	<b>Αμφίδρομη</b>
Διακλαδώσεις	<b>Όχι</b>	<b>Ναι</b>	<b>Ναι</b>	<b>Ναι</b>
Προσανατολισμός	<b>Όχι</b>	<b>Όχι</b>	<b>Ναι</b>	<b>Όχι</b>
Αρίθμηση	<b>Όχι</b>	<b>Ναι</b>	<b>Ναι</b>	<b>Όχι</b>

## 1.2 Μονοκονδυλιά

Η περίπτωση της Μονοκονδυλιάς είναι και η απλούστερη απ' όλες. Το θεμέλιο έχει απλώς έξι βελάκια, που δείχνουν προς τις έξι πιθανές κατευθύνσεις: Πάνω, Κάτω, Αριστερά, Δεξιά, Εμπρός, Πίσω. Κάθε φορά που κάποιο απ' αυτά τα βελάκια ανάβει, το παιδί θα πρέπει να τοποθετήσει ένα νέο κυβίδιο σε αυτό που τοποθέτησε αμέσως πριν, προς την κατεύθυνση που έχει μόλις υποδειχθεί. Αφότου το βάλει, θα πατάει ένα κατάλληλο πλήκτρο και θα ανάβει το βελάκι της επόμενης υπόδειξης.

### 1.2.1 Υλικά θεμελίου

Πέραν του κεντρικού μικροελεγκτή, θα χρειάζονται και 6 φωτάκια LED για την επικοινωνία των οδηγιών προς τον παίκτη.

### 1.2.2 Προγραμματισμός θεμελίου

Δεδομένου ότι η κάθε οδηγία θα αποτελείται από 6 δυνατές επιλογές (πάνω, κάτω, αριστερά, δεξιά, μπρος, πίσω), τρία bit φτάνουν και περισσεύουν για να την κωδικοποιήσουν. Για ευκολία στον προγραμματισμό, λοιπόν, κάθε μία θα αποθηκεύεται σε ένα nybble, και άρα κάθε byte θα κωδικοποιεί δύο οδηγίες. Το μόνο που χρειάζεται για το πρόγραμμα είναι να ανάβει με τη σειρά το κάθε βελάκι, ώσπου να πατήσει ο χρήστης το πλήκτρο για να ανάψει το επόμενο.

### 1.2.3 Υλικά κυβιδίων

Δεδομένου ότι τα κυβίδια δε θα περιέχουν καμία λογική μέσα τους, και δεδομένου ότι το κάθε κυβίδιο θα συνδέεται σε ένα και μόνο ένα άλλο κυβίδιο, αρκεί ως υλικό κατασκευής το σκέτο πλαστικό. Μάλιστα, από κατασκευαστικής άποψης, αρκεί να έχουν μία προεξοχή σε μία έδρα τους και εσοχές στις υπόλοιπες. Σημειωτέον ότι τέτοιου είδους κυβίδια υπάρχουν ήδη στο εμπόριο.

#### 1.2.4 Εικόνες

Μερικές φωτογραφίες των προκαταρκτικών εργασιών που είχαν γίνει ως προς αυτήν τη λύση μπορούν να βρεθούν στην υπο-ενότητα Α'1.

### 1.3 Αριθμημένα κυβίδια

Στην περίπτωση της Αρίθμησης, τα κυβίδια δεν είναι 100% ανταλλάξιμα και πανομοιότυπα μεταξύ τους. Το καθ' ένα έχει έναν αύξοντα αριθμό, και θα πρέπει να τοποθετούνται με αυστηρά αύξουσα σειρά. Αυτό δυσχεραίνει κάπως την κατασκευή, καθώς σε κάθε βήμα πρέπει ο παίκτης να αναζητεί ποιος κύβος έχει σειρά. Σε αντάλλαγμα, όμως, δίδεται η δυνατότητα να «επικοινωνήσει» κανείς πολύ πιο πολύπλοκα σχήματα στον παίκτη: αντί το κάθε κυβίδιο να μπαίνει στο αμέσως προηγούμενό του, μπορεί να πει κανείς «στον κύβο υπ' αριθμόν 5 βάλε από αριστερά του ένα κυβίδιο». Έτσι, η κατασκευή μπορεί να διακλαδίζεται, και να δημιουργεί σχήματα αυθαίρετης πολυπλοκότητας αντί για μονοκονδυλιές μόνο. Το μειονέκτημα, βέβαια, είναι ότι η κατασκευή δεν μπορεί να γίνει ιδιαίτερα στιβαρή.

#### 1.3.1 Υλικά θεμελίου

Εκτός από τα 6 LED που προαναφέραμε στην παράγραφο 1.2.1, θα χρειαστεί και κάποιου είδους ένδειξη για το σε ποιο κυβίδιο να μπει το επόμενο. Ο απλούστερος τρόπος να συμβεί αυτό είναι με δύο ενδείκτες 7 τμημάτων.

#### 1.3.2 Προγραμματισμός θεμελίου

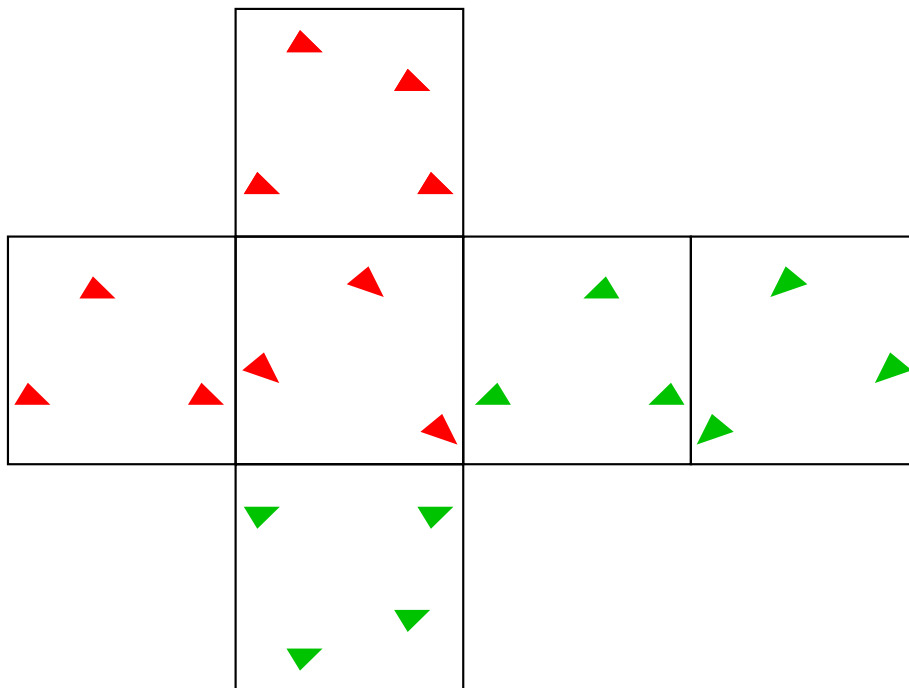
Σ' αυτήν τη λύση, για την κάθε οδηγία δε χρειάζεται μόνο η κατεύθυνση αλλά και ο αύξων αριθμός του κυβιδίου. Σε ένα byte λοιπόν μπορούμε να κωδικοποιήσουμε μέχρι 32 κυβίδια: 3 bit για την κατεύθυνση και 5 bit για τον αύξοντα αριθμό. Αν χρειαζόμαστε περισσότερα κυβίδια, σε 2 byte μπορούν να κωδικοποιηθούν μέχρι 8192 κυβίδια, ήτοι πρακτικά άπειρα.

#### 1.3.3 Υλικά κυβιδίων

Ως άνω, (βλ. παράγραφο 1.2.3) αλλά με κάποια αρίθμηση.

### 1.4 Προσανατολισμένα κυβίδια

Η λύση αυτή είναι η απλούστερη όσων χρειάζονται κάποιου είδους λογική μέσα στο κάθε κυβίδιο. Πλέον πρέπει, όχι μόνο τα κυβίδια να είναι αριθμημένα, αλλά η κατασκευή τους να είναι τέτοια που να μην τους επιτρέπει να συναρμολογηθούν αν δεν έχουν όλα τον ίδιο προσανατολισμό. Σε αντάλλαγμα, καθιστά τη δημιουργία σχημάτων πολύ ευκολότερη: Αντί να ανάβουν βελάκια στο θεμέλιο και να πρέπει ο παίκτης να καταλάβει τι πρέπει να κάνει μ' αυτά, θα ανάβει σε κάθε βήμα ένα λαμπάκι ακριβώς στην έδρα του κυβιδίου στο οποίο πρέπει να τοποθετηθεί το επόμενο. Έτσι, το μόνο που θα χρειάζεται να κάνει ο παίκτης θα είναι να τοποθετήσει ένα νέο κυβίδιο στην έδρα που βλέπει να φωτίζεται.



ΣΧΗΜΑ 1.1: Ένα πιθανό ανάπτυγμα των μηχανικών συζευκτών των Προσανατολισμένων Κυβιδίων.

#### 1.4.1 Υλικά θεμελίου

Εδώ το θεμέλιο δε χρειάζεται ούτε ένδειξη για τον Α/Α του κάθε κυβιδίου, ούτε ένδειξη για την κατεύθυνση. Το μόνο που χρειάζεται, πέραν του κεντρικού μικροελεγκτή, είναι ένα πλίκτρο με το οποίο να μπορεί να λέει ο χρήστης στο θεμέλιο «προχώρα στο επόμενο βήμα».

#### 1.4.2 Προγραμματισμός θεμελίου

Η κωδικοποίηση της κάθε οδηγίας θα είναι λίγο-πολύ ίδια με την προηγούμενη ενότητα. Απλώς, επιλέξαμε να αναθέσουμε 8 bit στην κάθε κατεύθυνση καθώς αυτό απλουστεύει αρκετά το hardware που θα χρειαστεί. (Αλλωστε, και 250+ κυβίδια είναι υπεραρκετά.)

Από κει και πέρα, σε κάθε βήμα, το θεμέλιο θα μεταδίδει σειριακά 16 bit σε κάποια κατάλληλη γραμμή του. Τα μισά bit θα αντιστοιχούν στην ταυτότητα του κυβιδίου, και τα άλλα μισά στην εντολή – το αν η εντολή θα πρέπει να 'ναι πρώτη ή δεύτερη είναι σχεδιαστική επιλογή που εν τέλει δε χρειάστηκε ν' αναλυθεί. Εκ μέρους του θεμελίου, αυτό αρκεί: τα υπόλοιπα αναλαμβάνονται από το εκάστοτε κυβίδιο.

#### 1.4.3 Υλικά κυβιδίων

Το κάθε κυβίδιο θα πρέπει να είναι κατασκευασμένο έτσι ώστε η «πάνω» έδρα του να μπορεί να εφαρμόσει μόνο στην «κάτω» έδρα κάποιου άλλου, και μάλιστα με συγκεκριμένο προσανατολισμό. Αντίστοιχη μέριμνα πρέπει να ληφθεί και για τα ζεύγη «αριστερά/δεξιά» και «εμπρός/πίσω». Υπάρχουν διάφορες επιλογές για το πώς θα μπορούσε να επιτευχθεί αυτό· κυρίως βασίζονται σε ασύμμετρα σχήματα, όπως σκαλινά τρίγωνα. Ένα παράδειγμα φαίνεται στο σχήμα 1.1.

Στο σχήμα αυτό λοιπόν φαίνεται ότι η ζητούμενη ασυμμετρία έχει υλοποιηθεί τόσο όσον αφορά το σχήμα των εσοχών/εξοχών, όσο και όσον αφορά τη μεταξύ τους διάταξη, είτε και τα δύο. Χρειαζόμαστε σύνολο τρία τέτοια διαφορετικά σχήματα, καθώς θέλουμε ένα για την κάθε ένωση πάνω-κάτω, αριστερά-δεξιά, εμπρός-πίσω· από το κάθε ένα απ' αυτά τα ζεύγη, η μία έδρα θα έχει την «αρσενική» εκδοχή του σχήματος και η άλλη τη «θηλυκή». Στο σχήμα, οι μεν συζευκτικές φαίνονται με κόκκινο και οι δε με πράσινο.

Πέραν των διαφορετικών σχημάτων, θα μπορούσαμε να χρωματίσουμε και την κάθε έδρα με διαφορετικό χρώμα, έτσι ώστε όλες οι έδρες ίδιου χρώματος να κοιτάζουν προς την ίδια κατεύθυνση. Αυτό βέβαια θα ενείχε τον κίνδυνο να προσπαθήσει ο παίκτης να συνδέει ίδια χρώματα (λχ κόκκινο με κόκκινο) αντί για συμπληρωματικά (λχ κόκκινο με κυανό).

Πέραν του πλαστικού ως δομικό υλικό, το κάθε κυβίδιο θα χρειαστεί και κάποια ηλεκτρονικά κυκλώματα. Αν αποφασίσουμε να χρησιμοποιήσουμε ολοκληρωμένα που υπάρχουν ήδη στο εμπόριο, θα χρειαστούμε:

- Δύο καταχωρητές ολίσθησης (shift registers) των 8 bit
- Ένα συγκριτή (comparator) των 16 bit
- Ένα κοινό τρανζίστορ (BJT/MOSFET) για λογική άρνηση
- Έξι LED, ένα για κάθε έδρα

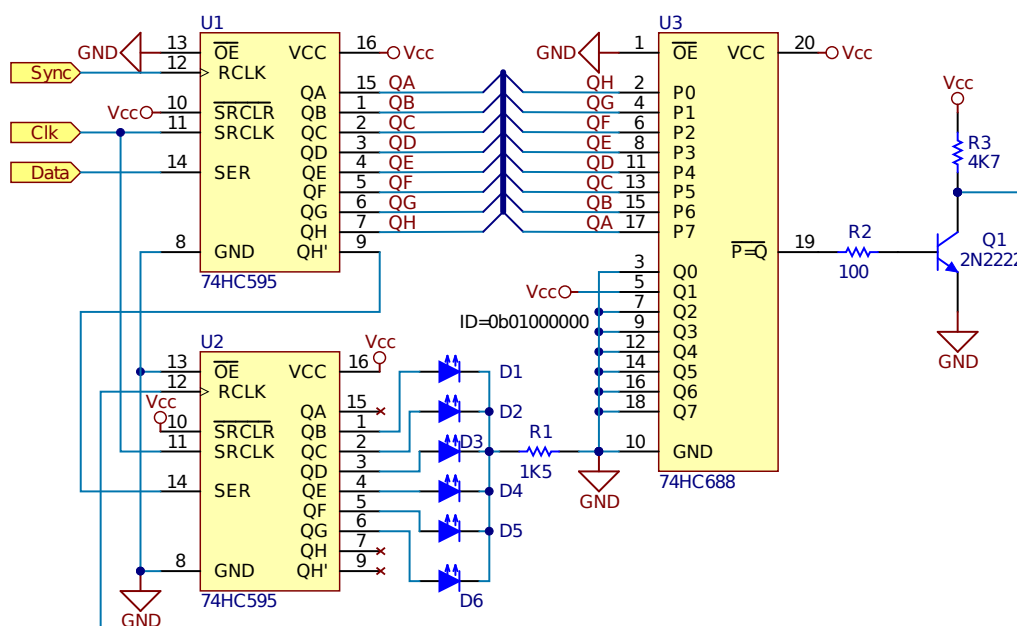
Το κάθε κυβίδιο, σε κάθε έδρα του, θα χρειάζεται και 5 ακροδέκτες:  $V_{cc}$ , Ground, Data, Clock, Sync. Όσοι ακροδέκτες έχουν ίδιο όνομα θα είναι βραχυκυκλωμένοι μεταξύ τους. Η σύνδεση δύο κυβιδίων θα ενώνει την πεντάδα των ακροδεκτών, έτσι ώστε κάθε ακροδέκτης να έχει αγωγή με όλους τους υπολοίπους ακροδέκτες της κατασκευής που έχουν ίδιο όνομα.

Έχοντας επιλέξει να κωδικοποιήσουμε την κάθε εντολή αφιερώνοντας ένα bit σε κάθε LED, αυτό μας δίνει αμέσως τη δυνατότητα να οδηγήσουμε τα LED απ' ευθείας από την έξοδο του καταχωρητή ολίσθησης. Ένα σχέδιο του κυκλώματος φαίνεται στο σχήμα 1.2.

#### 1.4.4 Υλοποιητέα λογική κυβιδίων

Όπως προαναφέραμε, για την κάθε οδηγία, το θεμέλιο θα μεταδίδει 16 bit στη σειρά. Αυτά θα μεταδίδονται σε όλα τα κυβίδια ταυτόχρονα, λόγω της αγωγικής σύνδεσης που προαναφέραμε. Τα 16 αυτά bit θα αποθηκεύονται ένα-προς-ένα στους δύο καταχωρητές ολίσθησης: ο ένας απ' αυτούς θα καταλήγει να περιέχει τον αύξοντα αριθμό του κυβιδίου, και ο άλλος την εντολή. Ο καταχωρητής του A/A θα οδηγεί την παράλληλη έξοδό του στη μία είσοδο του συγκριτή. Όσον αφορά την άλλη είσοδο του συγκριτή: Το κάθε ένα από τα bit της θα οδηγείται είτε στην τάση τροφοδοσίας είτε στη γείωση, με ένα συνδυασμό ο οποίος θα είναι κατασκευαστικά πάγιος και θα αντιστοιχεί στον A/A του κυβιδίου. Αν ο συγκριτής βρει ότι η ταυτότητα του κυβιδίου είναι η ίδια με την ταυτότητα που εκπέμφθηκε, τότε οδηγεί την εντολή στα 6 LED ώστε να ανάψει ή να σβήσει όποιο απ' αυτά θέλουμε.

Αξίζει να σημειωθεί ότι, κατ' αυτόν τον τρόπο, όλη η κατασκευή θα αποτελεί ένα ενιαίο κύκλωμα. Αυτό έχει το πλεονέκτημα της απλότητας, αλλά και το μειονέκτημα ότι οι παράσιτες χωρητικότητες είναι ευθέως ανάλογες με το μέγεθος



ΣΧΗΜΑ 1.2: Κυκλωματικό διάγραμμα Προσανατολισμένων κυβιδίων

της κατασκευής. Αυτό το πρόβλημα δε διερευνήθηκε στο παρόν πόνημα, αλλά εάν επρόκειτο να υλοποιήσει κανείς αυτήν τη λύση θα έπρεπε να διερευνηθεί.

### 1.4.5 Εικόνες

Στην υπο-ενότητα Α'2 παρατίθενται τα εξής:

- Κυκλωματικό διάγραμμα θεμελίου
- Φωτογραφία του πρώτου δοκιμαστικού κυκλώματος στο raster
- Κυκλωματικό διάγραμμα κυβιδίου σε μεγαλύτερο μέγεθος

## 1.5 Συμμετρικά κυβίδια

Αυτή η λύση είναι και η πιο αισθητικά όμορφη και η πιο χρηστικά απλή, αλλά επίσης είναι και η πιο δύσκολη και ακριβή στην κατασκευή και στην υλοποίηση. Επίσης, είναι η λύση που επιλέχθηκε για υλοποίηση εν τέλει.

Σ' αυτήν τη λύση, όπως και στην προηγούμενη, το κάθε κυβίδιο έχει LED στις έδρες του, με τις οποίες μπορεί να επικοινωνήσει στο χρήστη πού αλλού πρέπει να τοποθετηθούν νέα κυβίδια. Η διαφορά είναι ότι στη λύση αυτή όλα τα κυβίδια είναι πανομοιότυπα, και μάλιστα συμμετρικά: Δεν υπάρχει πλέον ούτε αρίθμηση, ούτε προσανατολισμός. Κάθε ένα απ' αυτά εμπεριέχει ένα φτηνό μικροελεγκτή, ο οποίος επικοινωνώντας με την υπόλοιπη δομή μπορεί να καταλάβει πού και πώς έχει τοποθετηθεί. Αποθηκεύοντας και αναλύοντας το πλήρες σχέδιο στη μνήμη του μικροελεγκτή, το κυβίδιο ανάβει ή σβήνει τα LED που υπάρχουν στις έδρες του. Μάλιστα, το κάθε ένα μπορεί να καταλάβει αν έχει τοποθετηθεί εσφαλμένα, και έτσι να καθοδηγεί τον παίκτη να το αφαιρέσει.

Η λύση που επιλέξαμε είναι να επικοινωνούν κάθε φορά μόνο όσα τεμάχια είναι άμεσα συνδεδεμένα μεταξύ τους: Το θεμέλιο θα επικοινωνεί μόνο με όσα κυβίδια έχουν τοποθετηθεί σε αυτό, και μετά η επικοινωνία θα γίνεται μόνο κυβίδιο-προς-κυβίδιο. Αυτό απλουστεύει κάποια πράγματα και περιπλέκει κάποια άλλα.

### 1.5.1 Υλικά θεμελίου

Ένας μικροελεγκτής και κάποιος τρόπος επιλογής του σχεδίου. Δε χρειάζεται πλέον κάποια άλλη είσοδος απ' το χρήστη προς το θεμέλιο. Θα χρειαστούν όμως κάποιες βάσεις (ενδεχομένως πάνω από μία) πάνω στις οποίες θα μπορούν να τοποθετηθούν τα κυβίδια.

### 1.5.2 Προγραμματισμός θεμελίου

Το σχέδιο θα είναι αποθηκευμένο στο θεμέλιο, όχι ως μία αλληλουχία από οδηγίες, αλλά ως μία αλληλουχία από συντεταγμένες. Αυτό το σχέδιο-αλληλουχία θα μεταδίδεται αυτούσιο από το θεμέλιο σε όσα κυβίδια συνδέονται σε αυτό, και μετά θα αναμεταδίδεται κυβίδιο-προς-κυβίδιο. Πέραν τούτου, θα χρειαστεί ένας τρόπος για κάθε νεο-συνδεόμενο κυβίδιο να αναγνωρίζει τον προσανατολισμό του και τη θέση του στο χώρο· θα πρέπει λοιπόν είτε το θεμέλιο είτε κάποιο άλλο κυβίδιο να μπορεί να προσδίδει αυτές τις πληροφορίες.

### 1.5.3 Υλικά κυβιδίων

Για να μπορούν τα κυβίδια να λειτουργήσουν «έξυπνα», θα χρειαστεί ένας φτηνός μικροελεγκτής, όπως εν προκειμένω ο Arduino Nano. Για την υλοποίηση του πρωτοκόλλου επικοινωνίας θα χρειαστούμε τρεις ακροδέκτες: Τάση τροφοδοσίας, γείωση, δεδομένα. Όλοι οι ακροδέκτες τάσης τροφοδοσίας θα είναι βραχυκυκλωμένοι μεταξύ τους, και ομοίως οι ακροδέκτες γείωσης. Πληρέστερη ανάλυση των απαιτήσεων των υλικών, και ανάλυση της συνδεσμολογίας των ακροδεκτών των δεδομένων, θα γίνει στο κεφάλαιο 2.

### 1.5.4 Προγραμματισμός κυβιδίων

Κάθε φορά που το κυβίδιο θα λαμβάνει τροφοδοσία, θα εξετάζει από ποια έδρα και ακμή λαμβάνει τα δεδομένα του. Μετά, θα διαβάζει τις συντεταγμένες του προηγούμενου τεμαχίου (θεμέλιο ή κυβίδιο), και τον προσανατολισμό στον οποίο έχει συνδεθεί· με βάση αυτό, θα καταλαβαίνει την τοποθεσία του και τον προσανατολισμό του στο χώρο. Μετά, θα λαμβάνει μία-προς-μία όλες τις συντεταγμένες που αποτελούν το σχέδιο, και θα το εξετάζει για να δει αν θα χρειαστεί να ανάψει LED σε κάποια απ' τις έδρες του. Τέλος, θα περιμένει ώστε να μεταδώσει τις ίδιες ακριβώς πληροφορίες στα ενδεχόμενα νέα κυβίδια που θα συνδεθούν σε αυτό. Η διαδικασία θα αναλυθεί περαιτέρω στο κεφάλαιο 3.





## Κεφάλαιο 2

# Κατασκευαστικές Λεπτομέρειες

### 2.1 Εισαγωγή

Όπως αναφέραμε και νωρίτερα, απ' όλες τις λύσεις που παραθέσαμε, αυτή που επιλέχθηκε για υλοποίηση ήταν αυτή των Συμμετρικών Κυβιδίων. Έτσι, κρίνουμε σκόπιμο να ξεκινήσουμε με βάση τους κατασκευαστικούς περιορισμούς που εισάγει η πλήρης αυτή συμμετρία.

### 2.2 Συμμετρία εδρών

Ξεκινούμε με τις εξής παραδοχές:

- Όλες οι έδρες του κάθε κύβου πρέπει να είναι πανομοιότυπες.
- Ο κάθε κύβος πρέπει να μπορεί να συνδεθεί με οποιονδήποτε προσανατολισμό, φυσικά εφ' όσον αυτός ευθυγραμμίζει τις έδρες του με τα 3 βασικά επίπεδα. (Σύνολο 24 προσανατολισμοί.)
- Οτιδήποτε χρησιμοποιήσουμε για μηχανική ή ηλεκτρολογική σύζευξη θα έχει ένα «θιλυκό» κι ένα «αρσενικό» τμήμα. Λόγου χάρι, για την περίπτωση της μηχανικής σύζευξης: ένας μαγνήτης έχει βόρειο και νότιο πόλο· μία προεξοχή πρέπει να έχει μία αντίστοιχη εσοχή· το βελκρό χρειάζεται βελούδο και άγκιστρο· και ούτω καθ' εξής.

Με βάση αυτούς τους περιορισμούς, ιδού πώς διαμορφώνονται οι κατασκευαστικές ανάγκες:

#### 2.2.1 Κατασκευή συμμετρικών συζευκτών

Σημείωση: Στο παρόν σύγγραμμα, με τον όρο «συζευκτικές», θα αναφερόμαστε τόσο στα αντικείμενα που επιτυγχάνουν μηχανική σύζευξη όσο και σε αυτά που επιτυγχάνουν ηλεκτρική σύζευξη.

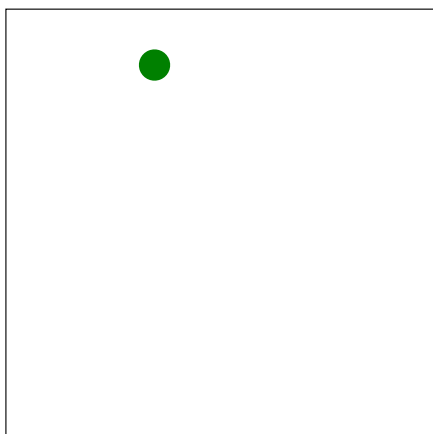
Έστω μία έδρα ενός κύβου, στην οποία τοποθετούμε σε ένα τυχαίο σημείο έναν αρσενικό συζευκτήρα, όπως φαίνεται στο σχήμα 2.1.

Η έδρα αυτή πρέπει να φαίνεται ακριβώς ίδια ακόμα κι αν περιστραφεί κατά κάποιο ακέραιο πλήθος ορθών γωνιών. Κατά συνέπεια, πρέπει να τοποθετήσουμε άλλους τρεις συζευκτικές, έτσι ώστε να σχηματιστεί ένα λοξό τετράγωνο και να υπάρχει τετραπλή περιστροφική συμμετρία, όπως φαίνεται στο σχήμα 2.2.

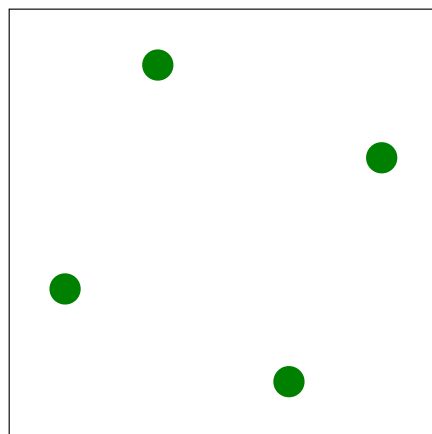
Φυσικά, ως τώρα εξετάσαμε μόνο αρσενικούς συζευκτικές. Για να μπορέσει η έδρα μας να συνδεθεί με την έδρα κάποιου άλλου κυβιδίου, θα πρέπει

να υπάρχει συμπληρωματικότητα ανάμεσα σε αρσενικούς και θηλυκούς συζευκτίρες. Αυτό καταλήγει να σημαίνει ότι, αν αντικατοπτρίσουμε το σχήμα των αρσενικών συζευκτίρων, θα πρέπει να προκύψει το σχήμα των θηλυκών (σχήμα 2.3). Βλέπουμε έτσι ότι οι αρσενικοί και θηλυκοί συζευκτίρες της κάθε γραμμής σχηματίζουν έτσι ένα οκτάγωνο, του οποίου η κάθε γωνία είναι  $135^\circ$ . Επίσης, όλοι ισαπέχουν από το κέντρο, οπότε το εν λόγω οκτάγωνο είναι εγγράμμο σε κύκλο.

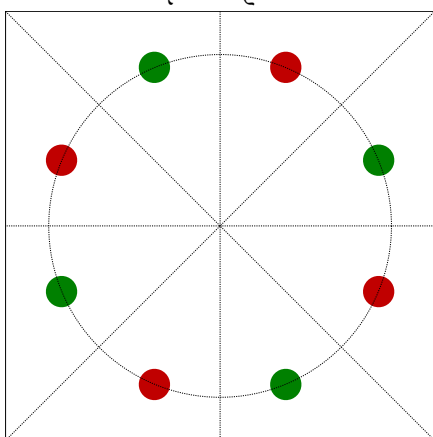
Αξίζει να σημειωθεί ένα συνακόλουθο των όσων έχουμε αναφέρει ως τώρα: Κανείς συζευκτίρας δεν μπορεί να τοποθετηθεί σε κάποιον από τους άξονες συμμετρίας. Ιδού η απόδειξη: Οι θέσεις των θηλυκών συζευκτίρων μπορούν να προκύψουν από τη θέση ενός και μόνο αρσενικού, αντικατοπτρισμένου μία φορά για κάθε άξονα συμμετρίας. Αν τοποθετούσαμε κάποιον αρσενικό συζευκτίρα πάνω σε κάποιον άξονα συμμετρίας, και τον αντικατοπτρίζαμε ως προς αυτόν τον άξονα, θα καταλήγαμε να πρέπει να τοποθετήσουμε στην ίδια θέση ένα θηλυκό κι έναν αρσενικό συζευκτίρα, το οποίο είναι άτοπο όπως φαίνεται στο σχήμα 2.4.



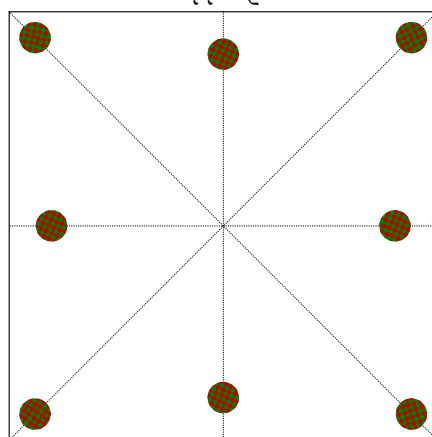
ΣΧΗΜΑ 2.1: Ένας συζευκτίρας σε μία έδρα



ΣΧΗΜΑ 2.2: Τετραπλασιασμός για συμμετρία



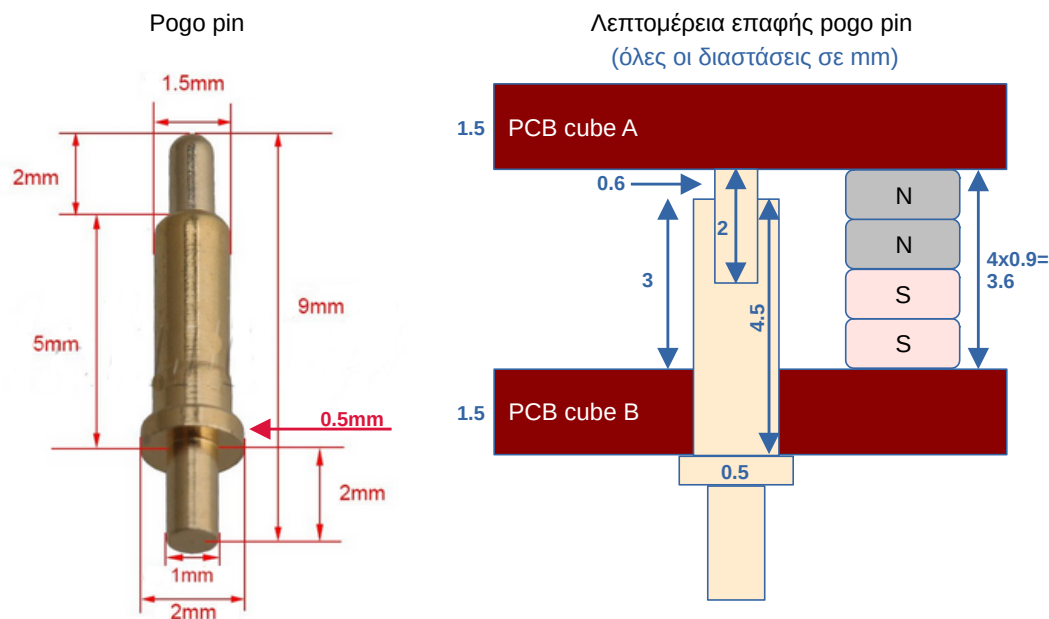
ΣΧΗΜΑ 2.3: Αντικατοπτρισμός των συμπληρωματικών συζευκτίρων



ΣΧΗΜΑ 2.4: Σύγκρουση συζευκτίρων

### 2.2.2 Μηχανικοί συζευκτίρες

Για τη μηχανική σύζευξη επιλέξαμε να χρησιμοποιήσουμε μαγνήτες.



ΣΧΗΜΑ 2.5: Μία ακίδα με ελατήριο (αγγλιστί rogo pin) στην ίδια κλίμακα με τους μαγνήτες.

#### Πλεονεκτήματα:

- (+) Καθιστούν τη συναρμολόγηση πολύ πιο εύκολη
- (+) Δεν απαιτούν ακριβείς ή δυνατές κινήσεις
- (+) Είναι φτηνοί, οπότε μπορούν να τοποθετηθούν σε μεγάλες ποσότητες χωρίς πρόβλημα

#### Μειονεκτήματα:

- (-) Δεν αντέχουν στις διατμητικές τάσεις

Για τη μηχανική σύζευξη, η χρήση επί πλέον συζευκτών οδηγεί σε καλύτερη και πιο ομοιόμορφη σύζευξη. Γι' αυτόν το λόγο, η τοποθέτησή τους επιλέχθηκε να γίνει ακριβώς όπως στο σχήμα 2.3, κολλώντας τους πάνω στις πλακέτες με κόλλα στιγμής.

### 2.2.3 Ηλεκτρικοί συζευκτές

Για την ηλεκτρική σύζευξη, επιλέξαμε ακίδες με ελατήρια (αγγλιστί rogo pin, βλ. σχήμα 2.5) και νησίδες χαλκού (pad).

#### Πλεονεκτήματα:

- (+) Εύρωστα απέναντι σε κατασκευαστικές ατέλειες
- (+) Εύρωστα απέναντι σε παραμορφώσεις της κατασκευής

#### Μειονεκτήματα:

- (-) Δεν προσφέρουν καμία μηχανική σύζευξη
- (-) Απωθούν τις επιφάνειες μεταξύ τους
- (-) Ακριβά σε μεγάλη ποσότητα (0.05€ - 0.10€ έκαστο)

### 2.2.4 Μείωση πλήθους ακίδων

Παρατηρούμε ότι, αν επρόκειτο να χρησιμοποιήσουμε πολλά `rogo pin`, όχι μόνο θα ανέβαινε υπέρογκα το κόστος, αλλά θα δυσχεραινόταν έντονα και η μηχανική σύζευξη. Κατά συνέπεια, θα μας ήταν συμφέρον να μειώσουμε το απαραίτητο πλήθος τους.

Αντιθέτως, τα `pad` δεν έχουν κανένα απ' αυτά τα δύο αρνητικά. Χάρη στη φωτολιθογραφία, η κατασκευή τους ουσιαστικά δεν έχει κόστος. Επίσης, παρ' ότι δε συνεισφέρουν στη μηχανική σύζευξη, αν μη τι άλλο δεν τη δυσχεραίνουν.

Η τελευταία απαραίτητη παρατήρηση είναι ότι, ενώ η χρήση πολλών μηχανικών συζευκτών καθιστά τη μηχανική σύζευξη πιο ομοιόμορφη και σταθερή, η χρήση πολλών ηλεκτρικών συζευκτών για την ίδια γραμμή είναι άχρηστη. Άπαξ και συνδεθεί ένα `rogo pin` με ένα `pad`, η ηλεκτρική σύζευξη έχει επιτευχθεί, και μετέπειτα τυχούσες συζεύξεις θα ήταν άχρηστες.

### Παρουσίαση λύσης

Με βάση όλα αυτά, κρίναμε σκόπιμο να βάζουμε σε κάθε έδρα του κυβιδίου, για κάθε γραμμή, 4 `pad` και 1 `rogo pin`. Έτσι, οπωσδήποτε και να περιστραφεί το κυβίδιο, ακριβώς ένα `rogo pin` θα ενωθεί με ακριβώς ένα `pad`, και η ηλεκτρική σύζευξη θα επιτευχθεί. Πέραν τούτου, όπως θα δούμε αργότερα, αυτό θα μας επιτρέψει να καταλάβουμε και με ποιον ακριβώς προσανατολισμό έχει συνδεθεί το κυβίδιο. Μία πιθανή διάταξη φαίνεται στο σχήμα 2.6.

Οι ηλεκτρικές γραμμές που επιλέξαμε να χρησιμοποιήσουμε είναι τρεις: Τάση τροφοδοσίας, γείωση, δεδομένα. Κατά συνέπεια, σε κάθε έδρα, θα χρειαστούμε 12 `pad` και 3 `rogo pin`. Τονίζουμε ότι οι ακροδέκτες που αντιστοιχούν σε διαφορετικές γραμμές μπορούν να απέχουν διαφορετικές αποστάσεις από τις ακμές της έδρας.

Με βάση όλα αυτά, μία πιθανή διάταξη των μαγνητών και των ακίδων/νησίδων μπορεί να φανεί στο σχήμα 2.7:

**Κόκκινοι κύκλοι** είναι οι βόρειοι μαγνητικοί πόλοι

**Γαλανοί κύκλοι** είναι οι νότιοι μαγνητικοί πόλοι

**Λαδί τετράγωνα** είναι οι νησίδες χαλκού

**Μωβ τετράγωνα** είναι οι ακίδες με ελατήρια

**Κίτρινος κύκλος** είναι το κεντρικό LED

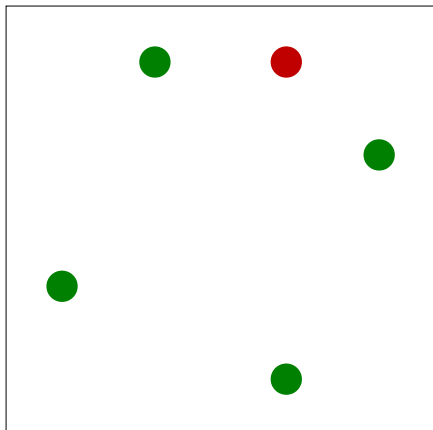
$V_{in}$  είναι η τάση τροφοδοσίας

$D$  είναι τα δεδομένα ( $D_{out}$  για το `rogo pin` εξόδου,  $D_{in0}$  έως και  $D_{in3}$  για τα `pad` εισόδου)

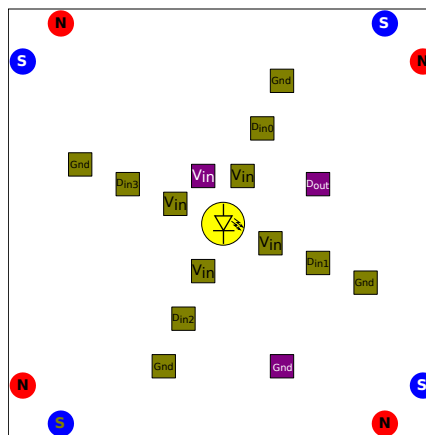
$Gnd$  είναι η γείωση.

## 2.3 Δομικό υλικό εκάστοτε έδρας

Ως δομικό υλικό για την κάθε μεμονωμένη έδρα επιλέχθηκε απλώς η ίδια πλακέτα που τυπώθηκε για τα ηλεκτρικά κυκλώματα. Οι ηλεκτρολογικοί ακροδέκτες



ΣΧΗΜΑ 2.6: Διάταξη ηλεκτρικών ακροδεκτών. Με πράσινο τα pad και με κόκκινο το rogo pin.



ΣΧΗΜΑ 2.7: Μία πλήρης έδρα κυβιδίου

που ενώνουν τις πλακέτες κάθε κυβιδίου μεταξύ τους προσδίδουν επίσης και την απαραίτητη μηχανική σύζευξη.

Είχε επίσης εξεταστεί ως πιθανότητα η τρισδιάστατη εκτύπωση ειδικών πλαστικών επιφανειών για το σκοπό αυτό, αλλά εν τέλει για την κατασκευή των πρωτοτύπων απορρίφθηκε.

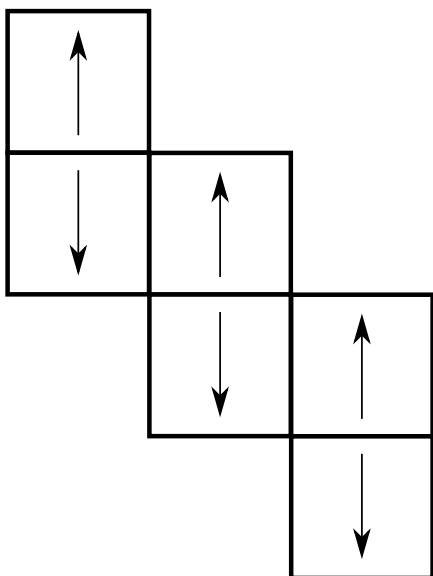
## 2.4 Συμμετρία κυβιδίων

Παρατηρούμε ότι πλέον οι ακμές της κάθε έδρας δεν είναι πανομοιότυπες. Μία θα έχει το rogo pin δεδομένων, μία το rogo pin τροφοδοσίας, μία το rogo pin γείωσης, και μία δε θα έχει καθόλου rogo pin. Αυτό προσδίδει στις έδρες μία μικρή ασυμμετρία ως προς τους αρσενικούς ακροδέκτες.

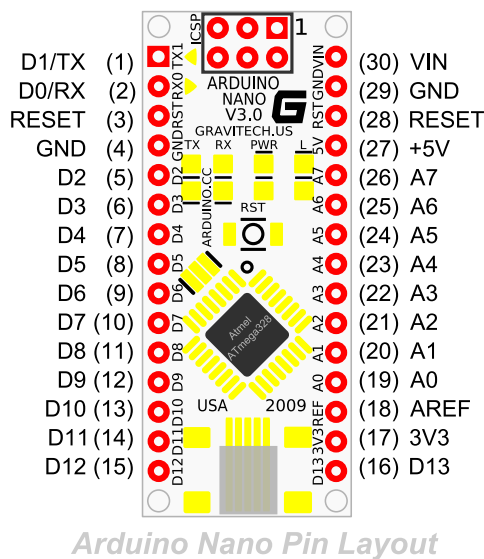
Για λόγους ευχρηστίας και αισθητικής, όμως, θα ήταν βολικό να είναι το κυβίδιο όσο το δυνατόν συμμετρικότερο. Θα συνέφερε λοιπόν να είναι δυνατός ο προσδιορισμός των προσανατολισμών όλων των υπολοίπων εδρών κοιτώντας μόνο μία.

Προς έκπληξή μας, η λύση αυτού του γεωμετρικού προβλήματος δεν ήταν απλή. Για την ακρίβεια, μπορέσαμε να βρούμε μόλις τέσσερις πιθανές διευθετήσεις οι οποίες να έχουν αυτό το χαρακτηριστικό. Το ανάπτυγμα μίας απ' αυτές φαίνεται στο σχήμα 2.8. Οι άλλες τρεις μπορούν να παραχθούν από αυτήν: Είτε αντικατοπτρίζοντάς την, είτε περιστρέφοντας όλες τις έδρες  $180^\circ$ , είτε και τα δύο. Τα τέσσερα αυτά πιθανά αναπτύγματα φαίνονται συγκεντρωτικά στο σχήμα Α.11.

Εστω ότι παίρνουμε το ανάπτυγμα αυτό, ξεκινώντας από κάποια τυχαία έδρα. Την ακριβή ταυτότητα αυτής της έδρας δεν την ξέρουμε: ξέρουμε μόνο τον προσανατολισμό της. Μετακινούμαστε στην έδρα προς την οποία δείχνει το βέλος της εν λόγω έδρας. Το βέλος της έδρας στην οποία θα καταλήξουμε θα δείχνει πάντοτε προς τα δεξιά, ανεξαρτίτως απ' την έδρα από την οποία ξεκινήσαμε. Αντιστοίχως, αν μετακινηθούμε στην έδρα που βρίσκεται στην αντίθετη κατεύθυνση απ' αυτήν προς την οποία δείχνει το βελάκι, αυτή θα δείχνει πάντοτε προς την αντίθετη κατεύθυνση.



ΣΧΗΜΑ 2.8: Ανάπτυγμα κυβιδίου



ΣΧΗΜΑ 2.9: Διάγραμμα ακροδεκτών Arduino Nano

Το ανάπτυγμα αυτό εν τέλει δε χρησιμοποιήθηκε στην τελική κατασκευή, λόγω κακής συνεννόησης κατά την παραγγελία των πλακετών των κυβιδίων. Κρίναμε όμως σκόπιμη τη συμπερίληψή του για πιθανή μελλοντική χρήση.

## 2.5 Συνδεσμολογία

Ο μικροελεγκτής που επιλέξαμε για τα κυβίδια, ο Arduino Nano, δεν έχει ιδιαίτερα πολλές γραμμές. Πρέπει λοιπόν να βεβαιωθούμε ότι η συνδεσμολογία μας δεν είναι ιδιαίτερα πολύπλοκη. Για αυτόν το σκοπό, ας παραθέσουμε το διάγραμμα ακροδεκτών του μικροελεγκτή καθ' αυτού, το οποίο φαίνεται στο σχήμα 2.9.

Όπως προαναφέραμε, όλοι οι ακροδέκτες  $V_{in}$  ενός κυβιδίου θα είναι βραχυκυκλωμένοι μεταξύ τους· ομοίως και οι ακροδέκτες  $Gnd$ . Αυτό σημαίνει ότι για όλα αυτά χρειαζόμαστε όλες-κι-όλες δύο γραμμές, τις οποίες δεν έχουμε τρόπο να γλιτώσουμε.

Σε κάθε έδρα θα χρειάζεται ένα LED, και κάθε LED πρέπει να λειτουργεί ξεχωριστά. Αυτό χρειάζεται 6 γραμμές, για τις οποίες επιλέξαμε τις  $D_8$  έως και  $D_{13}$ .

Σε κάθε έδρα υπάρχει επίσης μία ακίδα δεδομένων. Με τις έξι συνολικά αυτές ακίδες, το κυβίδιο θα μπορεί αν χρειαστεί να αποστέλλει δεδομένα. Άλλες έξι γραμμές, λοιπόν, για τις οποίες επιλέξαμε τις  $D_2$  έως και  $D_7$ .

Απομένουν 24 ακροδέκτες λήψης δεδομένων (4 σε κάθε έδρα) τους οποίους πρέπει να μπορούμε να διαχωρίσουμε μεταξύ τους. Αυτές θα συνδεθούν ανά 4 στις γραμμές  $A_0$  έως και  $A_5$ , με τον τρόπο που θα αναλυθεί αμέσως μετά. Νευραλγική σημασία έχει το γεγονός ότι αυτές οι γραμμές μπορούν να αναγνωστούν είτε με ψηφιακό τρόπο είτε με αναλογικό.

Παρατηρούμε λοιπόν ότι οι γραμμές μας φτάνουν κάπως οριακά· μάλιστα, έχουμε χρησιμοποιήσει μέχρι και τη γραμμή  $D_{13}$ , στην οποία συνδέεται και το ενσωματωμένο LED της πλακέτας. Έχουμε αφήσει διαθέσιμες τις γραμμές  $D_0$  και  $D_1$ , ώστε να επιτρέψουμε στο μικροελεγκτή να προγραμματίζεται μέσω της θύρας USB. Επίσης, αχρησιμοποίητες έχουν μείνει οι γραμμές  $A_6$  και  $A_7$ , οι

οποίες μπορούν να χρησιμοποιηθούν μόνο ως αναλογικές εισοδοί και ως εκ τούτου δε μας εξυπηρετούν.

### 2.5.1 Συνδεσμολογία ακροδεκτών δεδομένων

Προαναφέραμε ότι συνολικά οι ακροδέκτες λήψης δεδομένων είναι 24 τον αριθμό, και ότι όταν κάποιο rogo pin συνδεθεί σε κάποιον απ' αυτούς πρέπει να καταλάβουμε σε ποιον ακριβώς είναι. Θεωρητικά, η απλούστερη λύση θα ήταν να τους συνδέσουμε σε 24 διαφορετικές γραμμές, αλλά ο μικροελεγκτής μας δεν έχει ούτε κατά διάνοια τόσες γραμμές διαθέσιμες. Υπάρχει, όμως, ένα τέχνασμα που μπορούμε να εφαρμόσουμε.

Ένα πράγμα που ξέρουμε είναι το εξής: από τη στιγμή που κάθε έδρα θα έχει μόνο ένα rogo pin δεδομένων, τάση θα έχει μόνο ένα από τα 4 pad δεδομένων της κάθε έδρας. Αυτό μας επιτρέπει να χρησιμοποιήσουμε το δικτύωμα διαίρεσης τάσης του σχήματος 2.10, συνδεδεμένο με κάποια από τις γραμμές  $A_0$  έως και  $A_5$ . Η τάση των τεσσάρων αυτών εξόδων διαιρείται με ελαφρώς διαφορετικό τρόπο, όπως φαίνεται στην εξίσωση 2.1:

$$V_x = \frac{V_{CC}R_4}{\sum_{i=x}^4 R_i} \quad (2.1)$$

Έτσι, αν η ανάγνωση της γραμμής αυτής γίνει με αναλογικό τρόπο, μπορούμε να συμπεράνουμε ποιος ακροδέκτης είναι που έχει τροφοδοτηθεί· ειδικά, αν η ανάγνωση γίνει με ψηφιακό τρόπο, μπορούμε να λάβουμε δεδομένα χωρίς να μας νοιάζει ποιος είναι ο τροφοδοτούμενος ακροδέκτης.

### Επιλογή τιμών αντιστατών

Από τη στιγμή που μας νοιάζει τόσο η αναλογική όσο και η ψηφιακή ανάγνωση, υπάρχει μία σημαντική και λεπτή ισορροπία που πρέπει να επιτευχθεί ανάμεσα στην κατανάλωση ρεύματος και στην ευρωστία απέναντι στο θόρυβο. Οι παράγοντες που πρέπει να ληφθούν υπ' όψιν έχουν ως εξής:

- Πρέπει οι αντιστάτες να είναι αρκετά μεγάλοι ώστε να μην απαιτούν πολύ ρεύμα, καθώς θα οδηγούνται από εξόδους μικροελεγκτών και άρα το διαθέσιμο ρεύμα θα είναι περιορισμένο.
- Ταυτόχρονα, πρέπει οι αντιστάτες να είναι αρκετά μικροί ώστε να καθίσταται το κύκλωμα εύρωστο απέναντι στον ηλεκτρικό θόρυβο.
- Για να είναι η αναλογική ανάγνωση ακριβής και εύρωστη απέναντι στο θόρυβο, πρέπει οι 4 πιθανές τιμές τάσης να διαφέρουν όσο το δυνατόν περισσότερο μεταξύ τους, και άρα να μπορούν να χαμηλώσουν πολύ.
- Για να είναι η ψηφιακή ανάγνωση ακριβής και εύρωστη απέναντι στο θόρυβο, πρέπει οι τιμές τάσης να είναι όλες υψηλές, και άρα να διαφέρουν λίγο μεταξύ τους. Σύμφωνα με το φύλλο δεδομένων του ATmega328p, για να ερμηνευθεί μία τάση εισόδου ως λογικό 1 πρέπει κατ' ελάχιστον να ισούται με  $0.6V_{CC}$ , ήτοι 3V εν προκειμένω.

Θα κάνουμε προς το παρόν μια θεωρητική εισαγωγή, και μετά θα αναλύσουμε τις τιμές που επιλέξαμε όντως εν τέλει.

Έστω ότι θέλουμε η μετρούμενη τάση να μειώνεται κατά ένα σταθερό ποσό κάθε φορά που μετακινούμαστε έναν ακροδέκτη πιο πάνω. Διαρρούμε λοιπόν την τάση  $V_{CC}$  κατά μία σταθερά, έστω  $n$ , και ορίζουμε ότι η τάση του κάθε ακροδέκτη θα πέφτει κάθε φορά κατά  $V_{CC}/n$ . Με βάση αυτό, οι απαραίτητες αντιστάσεις  $R_1$  έως  $R_3$  υπολογίζονται όπως φαίνεται στην εξίσωση 2.2:

$$R_x = \frac{nR_4}{(n+x-4)(n+x-3)} \quad (2.2)$$

ή, με άλλα λόγια, ο αριθμητής κάθε φορά παραμένει σταθερός και ο παρονομαστής αυξάνεται τετραγωνικά.

Με βάση αυτό, όλοι οι αντιστάτες υπολογίζονται με βάση τον αντιστάτη πρόσδεσης  $R_4$ . Απομένει, λοιπόν, η παράθεση ενός κριτηρίου για την επιλογή του.

Αυτό μπορούμε να το επιλέξουμε με βάση το ρεύμα που θέλουμε να διαρρέει τους αντιστάτες στη μόνιμη κατάσταση. Το άθροισμα των τεσσάρων αυτών αντιστατών υπολογίζεται όπως φαίνεται στην εξίσωση 2.3:

$$R_4 + \sum_{x=1}^3 \frac{nR_4}{(n+x-4)(n+x-3)} = \frac{nR_4}{n-3} \quad (2.3)$$

και άρα το ρεύμα θα κυμαίνεται στα επίπεδα που φαίνονται στην εξίσωση 2.4:

$$\frac{V_{CC}(n-3)}{nR_4} \leq I \leq \frac{V_{CC}}{R_4} \quad (2.4)$$

Υπ' όψιν θα λάβουμε το πρώτο απ' τα δύο, για λόγους οι οποίοι θα γίνουν αντιληπτοί σε επόμενο κεφάλαιο. Οπότε, η ζητούμενη τιμή θα προκύψει λύνοντας ως προς  $R_4$ , όπως φαίνεται στην εξίσωση 2.5:

$$R_4 = \frac{V_{CC}(n-3)}{nI} \quad (2.5)$$

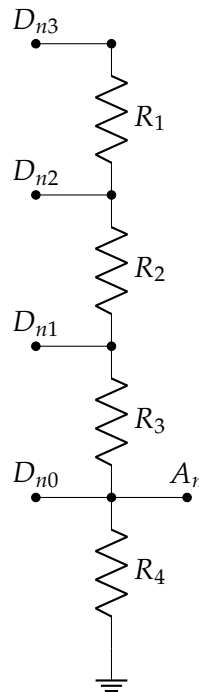
Έχοντας λοιπόν ένα διαθέσιμο εύρος τάσεων από 3V έως και 5V, μία πρώτη προσέγγιση θα ήταν να το ισομοιάσουμε, και άρα να επιτύχουμε τιμές τάσης ίσες με 3.5V, 4V, 4.5V, και 5V· με άλλα λόγια,  $n = 10$ . Ενδεικτικά, σ' αυτήν την περίπτωση, αν θέλουμε ρεύμα ίσο με 1mA, τότε οι απαιτούμενες τιμές των αντιστατών θα είναι  $R_1 = 625\Omega$ ,  $R_2 = 486\Omega$ ,  $R_3 = 389\Omega$ ,  $R_4 = 3.5k\Omega$ . Εναλλακτική επιλογή θα ήταν να επιλέξουμε  $n = 16$  ή  $n = 32$  ώστε να απλουστευτεί η ανάγνωση της αναλογικής τιμής της τάσης στο δυαδικό σύστημα.

Σύμφωνα με το φύλλο δεδομένων του ATmega328p, το ρεύμα που θα καταλήγει σε κάθε γραμμή-είσοδο θα ισούται με 1mA, και η πτώση τάσης λόγω του απαιτούμενου ρεύματος θα ισούται με 9.2%. Εφ' εξής αμφότερα αυτά τα μεγέθη θα θεωρούνται αμελητέα.

Για την κατασκευή του πρωτοτύπου, βέβαια, επιλέξαμε τις τιμές με βάση την ελάχιστη δυνατή κατανάλωση κατά τη μόνιμη κατάσταση. Έτσι, οι αντιστάτες έχουν περίπου εικοσαπλάσιες τιμές από τις προαναφερθείσες, και ισούνται με 10kΩ, 10kΩ, 10kΩ, και 82kΩ. Αυτό δημιούργησε κάποια προβληματάκια που θα αναλυθούν εν ευθέτω χρόνω.

**Σημαντική σημείωση:** Από τη στιγμή που ο αντιστάτης  $R_4$  προσδέει την είσοδο  $A_n$  στη γείωση, αυτό σημαίνει ότι δεν υπάρχει κανένας τρόπος διαχωρισμού ανάμεσα στις καταστάσεις «αυτή η έδρα είναι ασύνδετη» και «αυτή η έδρα είναι





ΣΧΗΜΑ 2.10: Συνδεσμολογία των ακροδεκτών δεδομένων

συνδεδεμένη, αλλά η ακίδα που συνδέθηκε σ' αυτήν είναι σε χαμηλή λογική στάθμη». Για την αποφυγή αύξησης του κόστους, ο διαχωρισμός αυτός γίνεται με ευριστικές μεθόδους μέσα στον πηγαίο κώδικα.

## 2.6 Κυκλωματικό διάγραμμα κυβιδίου

Το κυκλωματικό διάγραμμα φαίνεται στο σχήμα 2.11.

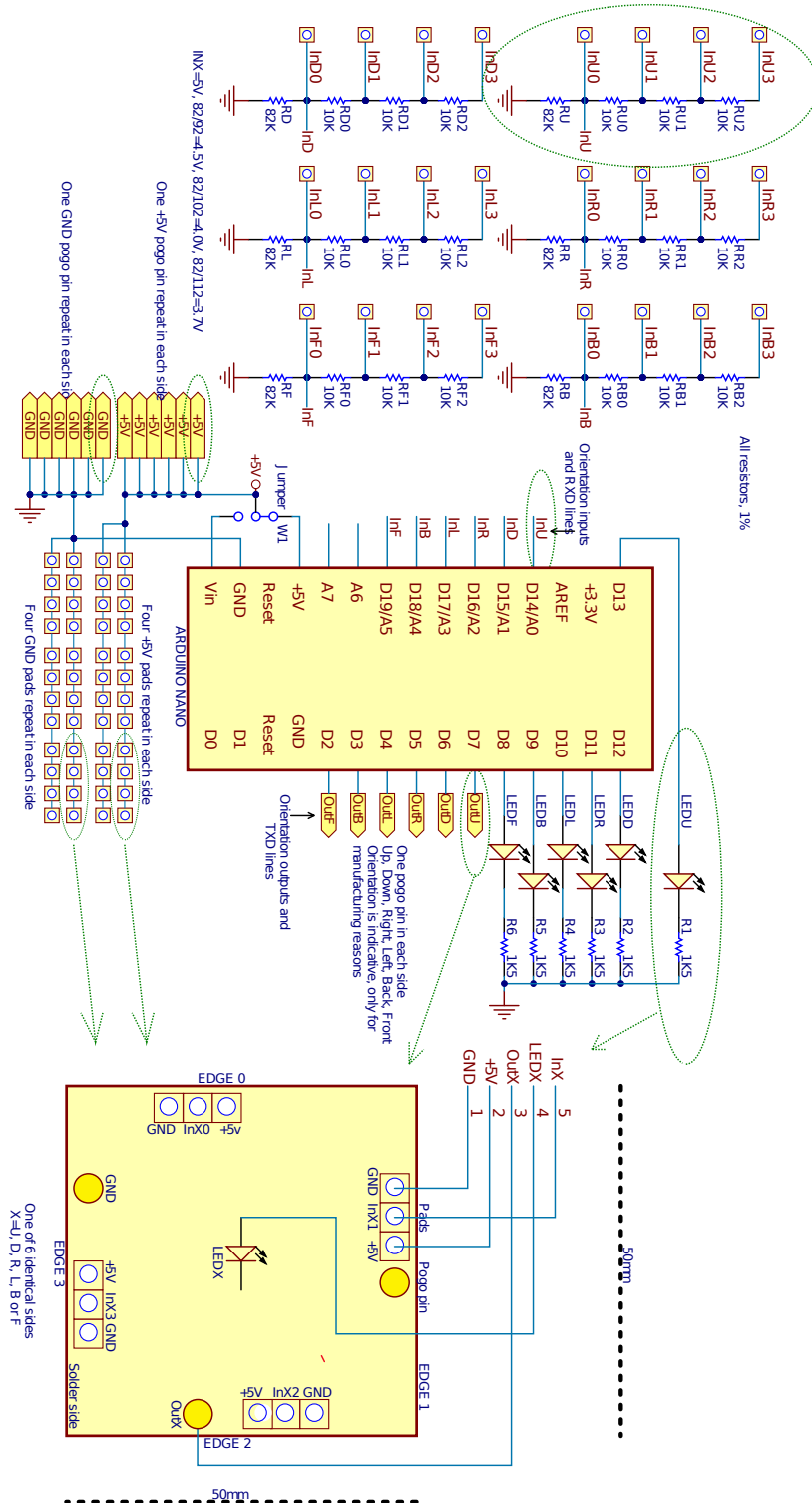
### 2.6.1 Επιλογή μικροελεγκτή

Πρέπει να τονιστεί ότι η επιλογή της πλήρους πλακέτας Arduino Nano αντί του σκέτου μικροελεγκτή ATmega328p έχει ως στόχο καθαρά την ευκολία ανάπτυξης του πρωτοτύπου. Για την παραγωγή του τελικού προϊόντος, θα μπορούσε ο μικροελεγκτής να χρησιμοποιηθεί μόνος του, μειώνοντας έτσι το κόστος και τις διαστάσεις των κυβιδίων.

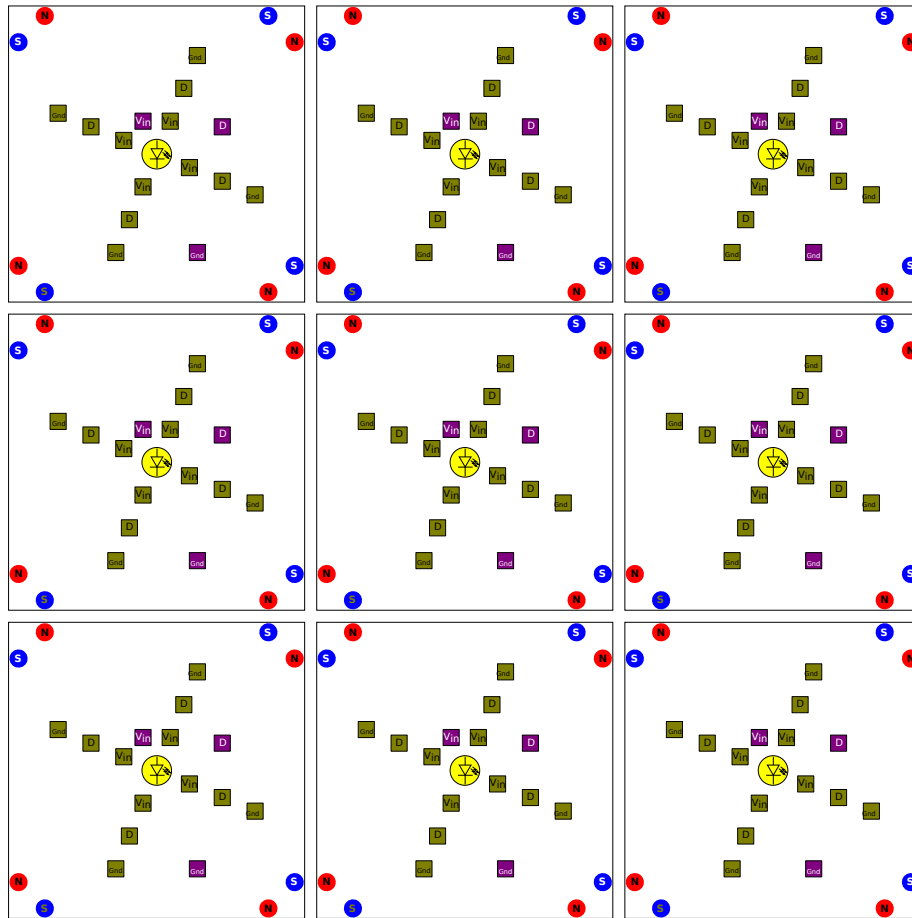
## 2.7 Δομή θεμελίου

Το απλούστερο πιθανό θεμέλιο, αγνοώντας τη διεπαφή επιλογής σχεδίου, θα ήταν μία απλή βάση πανομοιότυπη με μία έδρα ενός κυβιδίου, όπως φαίνεται στο σχήμα 2.7. Παρ' όλ' αυτά, αυτό δεν είναι απαραίτητο· το θεμέλιο μπορεί να έχει αυθαίρετα πολλές τέτοιες βάσεις, ώστε η κατασκευή να μπορεί να στηρίζεται από περισσότερα του ενός σημεία. Ένα τέτοιο παράδειγμα φαίνεται στο σχήμα 2.12. Για τους σκοπούς του παρόντος πονήματος, επιλέξαμε μόνο μία βάση ανά θεμέλιο, για να διευκολύνουμε την κατασκευή.

Αξίζει να σημειωθεί ότι όλα τα υλικά που θα χρησιμοποιηθούν για το θεμέλιο θα χρησιμοποιηθούν μόνο μία φορά, και άρα δεν επηρεάζουν την τιμή σχεδόν



ΣΧΗΜΑ 2.11: Πλήρες κυκλωματικό διάγραμμα κυβιδίου

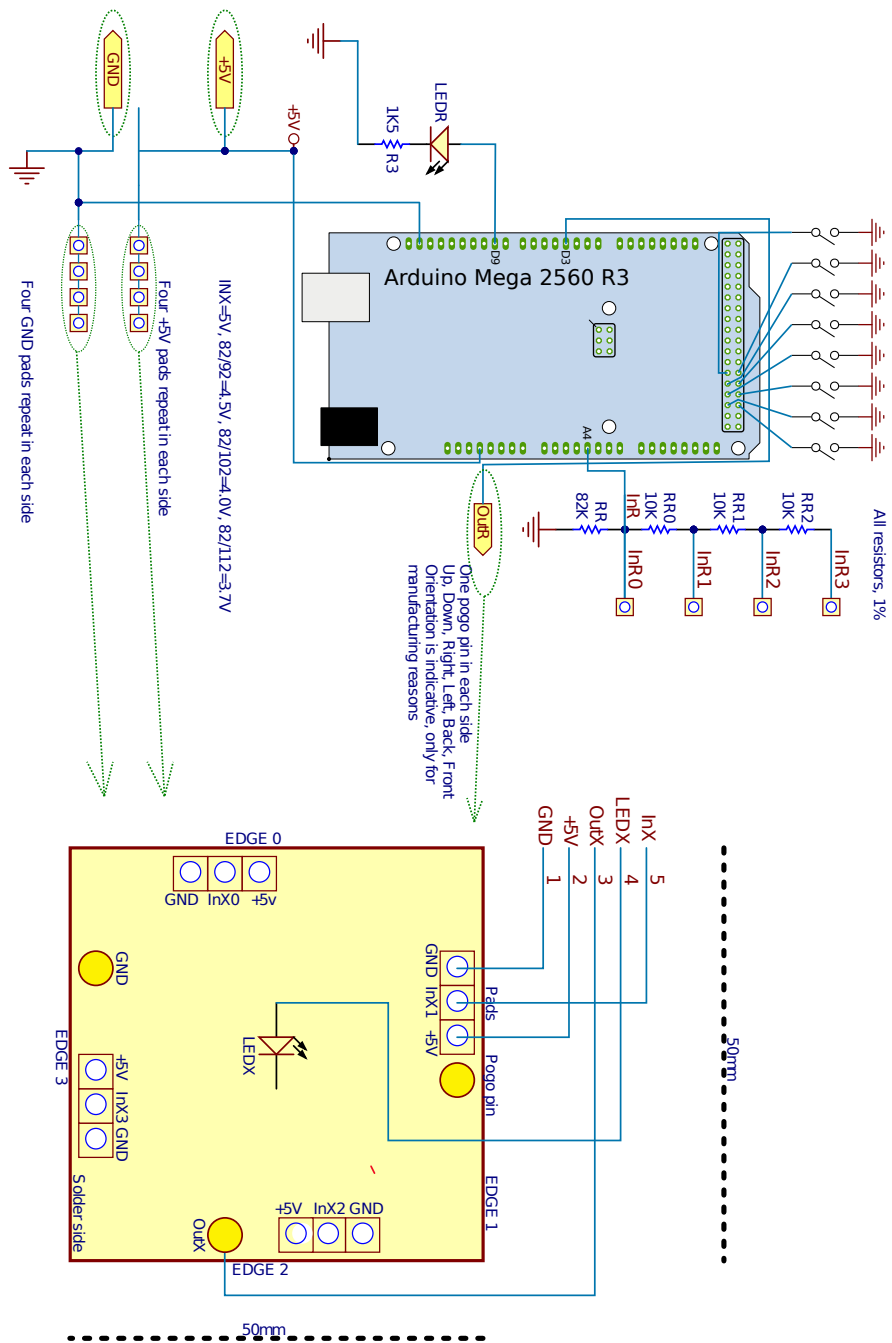


ΣΧΗΜΑ 2.12: Μία πιθανή διάταξη των βάσεων του θεμελίου

καθ' όλου. Αυτό μας επιτρέπει να χρησιμοποιήσουμε κάποιον πιο μεγάλο μικροελεγκτή γι' αυτό, όπως εν προκειμένω τον Arduino Mega 2560.

Υπάρχουν και κάποιες αλλαγές που μπορούν να γίνουν από κυβίδιο σε θεμέλιο, αν και στη δική μας κατασκευή δεν τις έχουμε κάνει. Αυτές είναι ότι πχ οι αντιστάτες του δικτυώματος διαίρεσης τάσης μπορούν να αντικατασταθούν από βραχυκυκλώματα, καθώς το θεμέλιο ξέρει τον προσανατολισμό του χωρίς να χρειαστεί αναλογική ανάγνωση κάποιου είδους. Πέραν τούτου, πρέπει στο θεμέλιο ο χρήστης κάπως να μπορεί να επιλέξει ποιο ακριβώς είναι το σχέδιο που θα αποφασίσει να χτίσει.

Το πλήρες κυκλωματικό διάγραμμα του θεμελίου φαίνεται στο σχήμα 2.13. Πάνω από το μικροελεγκτή φαίνονται οι μικροδιακόπτες (DIP switches) επιλογής σχεδίου. Στο θεμέλιο έχει συνδεθεί μόνο μία βάση, καθώς αυτό ακριβώς υλοποιήθηκε στο δικό μας πρωτότυπο· προφανώς, για ένα πλήρες προϊόν μπορούν να συνδεθούν περισσότερες.



ΣΧΗΜΑ 2.13: Κυκλωματικό διάγραμμα θεμελίου

## Κεφάλαιο 3

# Προγραμματισμός συμμετρικών κυβιδίων

### 3.1 Κυβικές περιστροφές

Προτού εξηγήσουμε τις καθ' αυτές προγραμματιστικές επιλογές που κάναμε, κρίνουμε σκόπιμη την ανάλυση των πιθανών προσανατολισμών στους οποίους θα μπορεί να περιέλθει το κάθε κυβίδιο, και των περιστροφών που μπορούν να οδηγήσουν από τον ένα στον άλλο.

#### 3.1.1 Καταμέτρηση προσανατολισμών

Ο υπολογισμός του συνολικού πλήθους των προσανατολισμών είναι μάλλον εύκολος. Κάθε προσανατολισμός μπορεί να περιγραφεί μοναδικά από το ποια έδρα του είναι στην κορυφή και το ποιον προσανατολισμό αυτή έχει. Οι έδρες είναι 6 και οι πιθανοί προσανατολισμοί της κάθε μίας είναι 4, άρα οι συνολικοί προσανατολισμοί είναι σύνολο  $6 * 4 = 24$ .

#### 3.1.2 Οκταεδρική συμμετρία

Για τη μαθηματική ανάλυσή μας, ας θεωρήσουμε το γεωμετρικό τόπο των σημείων για τα οποία ισχύει  $\max(|x|, |y|, |z|) = 3$ : με άλλα λόγια, έναν κύβο με κέντρο την αρχή των αξόνων και μήκος ακμής 6.

Ο στόχος μας μπορεί να αναχθεί στην εύρεση όλων των περιστροφών οι οποίες, αν εφαρμοστούν στον κύβο αυτόν, οδηγούν και πάλι στον ίδιο ακριβώς γεωμετρικό τόπο. Υπάρχει τρόπος να υπολογιστεί αυτό με τον αναλυτικό τύπο της τριδιάστατης περιστροφής, αλλά υπάρχει και μία άλλη πολύ απλούστερη λύση: Είναι αναγκαία συνθήκη, μετά την περιστροφή, οι τρεις απόλυτες τιμές  $|x|, |y|, |z|$  να συνεχίσουν να υπάρχουν στο τελικό άνυσμα. Οι θεμελιώδεις πράξεις που μπορούν να οδηγήσουν σε αυτό είναι αποκλειστικά και μόνο δύο:

- Αντιστροφή του προσήμου μίας από τις συντεταγμένες (καθώς αυτό διατηρεί την απόλυτο τιμή του)
- Αντιμετάθεση δύο συντεταγμένων μεταξύ τους (καθώς αυτό διατηρεί τις τιμές σε διαφορετικές θέσεις)

Τρεις αριθμοί  $[x, y, z]$  μπορούν να διαταχθούν με 6 διαφορετικούς τρόπους: πέραν τούτου, ανάλογα με το ποιος διατηρεί και ποιος αλλάζει το πρόσημό του, υπάρχουν 8 διαφορετικές περιπτώσεις. Έτσι προκύπτουν  $6 * 8 = 48$  διαφορετικοί γραμμικοί μετασχηματισμοί: ακριβώς το διπλάσιο απ' αυτό που θέλουμε.

Μία προσεκτικότερη παρατήρηση μπορεί να ισομοιράσει αυτούς τους γραμμικούς μετασχηματισμούς ακριβώς σε δύο κατηγορίες, με βάση το πλήθος των θεμελιωδών πράξεων (αντιστροφών/αντιμεταθέσεων) που τους απαρτίζουν. Συγκεκριμένα:

- Κάθε μία απ' τις θεμελιώδεις αυτές πράξεις είναι ένας *αντικατοπτρισμός*.
- Κάθε γραμμικός μετασχηματισμός που αποτελείται από *άρτιο* πλήθος αντικατοπτρισμών αποτελεί *περιστροφή*<sup>1</sup>.
- Κάθε γραμμικός μετασχηματισμός που αποτελείται από *περιττό* πλήθος θεμελιωδών πράξεων αποτελεί είτε απλό αντικατοπτρισμό είτε *καταχρηστική περιστροφή*, ήτοι συνδυασμό περιστροφής και αντικατοπτρισμού.

Με βάση αυτά, αν συμπεριλάβουμε και την εκφυλισμένη περίπτωση «δεν αλλάζουμε τίποτα», προκύπτουν ακριβώς 24 περιστροφές. Αυτό σημαίνει ότι κάθε προσανατολισμός μπορεί να μετατραπεί σε οποιονδήποτε άλλον με μία και μόνο περιστροφή.

Αυτού του είδους η συμμετρία, μαζί με τους μετασχηματισμούς που προκύπτουν από αυτήν, είναι γνωστή στη βιβλιογραφία ως οκταεδρική συμμετρία. Το όνομα εξηγείται από το γεγονός ότι τα οκτάεδρα έχουν ακριβώς την ίδια συμμετρία με τους κύβους, οπότε παρ' ότι οι κύβοι είναι εξάεδρα και όχι οκτάεδρα εντάσσονται στην ίδια ομάδα συμμετρίας.

Ιδού πώς ερμηνεύεται γεωμετρικά η κάθε περίπτωση:

- Αν απλώς αλλάζουμε πρόσημο σε 2 απ' τις συντεταγμένες, αυτό αποτελεί μία περιστροφή  $180^\circ$  γύρω από έναν από τους *βασικούς* άξονες.
- Αν κάνουμε μία αντιμετάθεση και μία αναστροφή κρατώντας μία συντεταγμένη ακριβώς ίδια, αυτό αποτελεί μία περιστροφή  $\pm 90^\circ$  γύρω από έναν από τους *βασικούς* άξονες.
- Αν αντιμεταθέσουμε δύο συντεταγμένες και αναστρέψουμε το πρόσημο της τρίτης, αυτό αποτελεί μία περιστροφή  $180^\circ$  γύρω από κάποιον από τους άξονες που διαπερνούν τις *ακμές* του κύβου.
- Αν κάνουμε δύο αντιμεταθέσεις, και προαιρετικά επίσης δύο αντιστροφές προσήμου, αυτό αποτελεί μία περιστροφή  $\pm 120^\circ$  γύρω από κάποιον απ' τους άξονες που διαπερνούν τις *κορυφές* του κύβου.

### 3.1.3 Προγραμματισμός οκταεδρικής συμμετρίας

Ας επιλέξουμε κάποιο άνυσμα ως άνυσμα αναφοράς, έστω το  $[1,2,3]$ . Από τη στιγμή που με μία και μόνο μία κατάλληλη περιστροφή μπορεί αυτό το άνυσμα να συμπέσει με οποιοδήποτε άλλο, αυτό μας επιτρέπει να αναπαριστούμε μέσα στο πρόγραμμα μας τόσο τα άνυσματά όσο και τις πιθανές περιστροφές με τον ίδιο ακριβώς τρόπο: Κάθε άνυσμα αναπαριστάται ουσιαστικά ως η περιστροφή που χρειάζεται για να συμπέσει με αυτό το άνυσμα αναφοράς.

Το πρόβλημα λοιπόν ανάγεται στο πώς θα αναπαρασταθεί κάθε περιστροφή. Είδαμε νωρίτερα ότι χρειάζονται το πολύ 6 θεμελιώδεις αντικατοπτρισμοί. Αυτό μας επιτρέπει να τους αντιστοιχίσουμε έναν-προς-έναν με τα 6 bit ενός ακεραίου

<sup>1</sup>Αυτό αποδεικνύεται ότι είναι γενικότερος νόμος της γεωμετρίας, και ισχύει για οποιοδήποτε πλήθος διαστάσεων.

από 0 έως και 47. Μετά, εξετάζοντας –πάντοτε με την ίδια σειρά!– ένα-προς-ένα τα bit αυτού του ακεραίου και κάνοντας τους αντίστοιχους αντικατοπτρισμούς, μπορεί να υλοποιηθεί κάθε δυνατή περιστροφή και έτσι κάθε δυνατό άνυσμα.

Για απλούστευση, οι καταχρηστικές περιστροφές αναπαριστώνται στον ίδιο τύπο δεδομένων με τις φυσιολογικές. Παρ' όλ' αυτά, ο διαχωρισμός τους είναι δυνατός εξετάζοντας το αν η δυαδική αναπαράστασή τους έχει άρτιο ή περιττό πλήθος άσπων.

Με βάση αυτά, έχουμε όλες τις πληροφορίες που χρειαζόμαστε προκειμένου να προσανατολίσουμε μοναδικά τον κάθε κύβο.

### 3.2 Χωρικές συντεταγμένες και προσανατολισμός

Η μετάδοση των χωρικών δεδομένων και του προσανατολισμού γίνεται αλυσιδωτά, με τρόπο που θυμίζει τη μαθηματική επαγωγή. Το θεμέλιο ξέρει αυτές τις πληροφορίες και μπορεί να τις μεταδώσει· μετά, αν κάποιιο κυβίδιο ξέρει αυτές τις πληροφορίες, τότε μπορεί να τις μεταδώσει· συνδυάζοντας αυτά τα δύο δεδομένα, συμπεραίνουμε ότι όλα τα κυβίδια μπορούν να ξέρουν τη θέση και τον προσανατολισμό τους στο χώρο.

Ας δούμε το σχήμα 2.7 στη σελίδα 15, υποθέτοντας ότι πρόκειται για μία βάση στο θεμέλιο. Παρατηρούμε ότι το `rogo pin` μετάδοσης δεδομένων  $D$  έχει γνωστό προσανατολισμό: κοιτάει «προς τα πάνω» και βρίσκεται «δεξιά και λίγο μπρος». Αυτό είναι κατασκευαστικά πάγιο, καθώς οι κατευθύνσεις αυτές κρίνονται με βάση το θεμέλιο.

Αν τώρα κάποιιο κυβίδιο συνδεθεί με το θεμέλιο, θα μπορέσει να καταλάβει ποιο `rad` συνδέθηκε με το `rogo pin`  $D$  του θεμελίου. Με βάση αυτό, θα μπορέσει να καταλάβει ότι το `rad` αυτό κοιτάει «προς τα κάτω» και βρίσκεται κι αυτό «δεξιά και λίγο μπρος». Με βάση αυτό, ο προσανατολισμός του νέου κυβιδίου έχει καθοριστεί μοναδικά.

Επόμενο βήμα είναι να επικοινωνήσει το θεμέλιο στο κυβίδιο ποιες είναι οι χωρικές συντεταγμένες της βάσης στην οποία έγινε η σύνδεση· έστω ότι είναι η κεντρική, και ότι άρα οι χωρικές της συντεταγμένες είναι  $[0, 0, 0]$ . Δεδομένου ότι το κυβίδιο ξέρει πλέον τον προσανατολισμό του, γνωρίζει ότι το σημείο αυτό βρίσκεται «ακριβώς από κάτω του», και μέσω αυτού μπορεί να συμπεράνει ότι οι δικές του χωρικές συντεταγμένες πρέπει αναγκαστικά να είναι  $[0, 0, 1]$ . Έχοντας πλέον όλες τις πληροφορίες που χρειάζεται, το νέο κυβίδιο μπορεί να τις αναμεταδώσει σε όποια νέα κυβίδια ενδεχομένως συνδεθούν με αυτό.

Σημαντική παρατήρηση: Άπαξ και σε ένα κυβίδιο θα έχουν αποδοθεί χωρικές συντεταγμένες, αυτές θα είναι μοναδικές. Άρα, δε χρειάζεται άλλος τρόπος ταυτοποίησης των κυβιδίων.

### 3.3 Ανίχνευση οδηγιών

Η επόμενη πληροφορία που χρειάζεται ένα κυβίδιο είναι το σε ποιες έδρες του πρέπει να συνδεθούν νέα κυβίδια, ώστε να μπορέσει να καθοδηγήσει το χρήστη κατάλληλα.

Η λύση που επιλέχθηκε είναι να αποθηκεύεται αρχικά το σχέδιο στο θεμέλιο ως μία αλληλουχία χωρικών συντεταγμένων  $[x, y, z]$ . Αυτό το σχέδιο, αφότου κάποιιο άρτι συνδεδεμένο κυβίδιο καταλάβει τη θέση και τον προσανατολισμό του, θα μεταδίδεται αυτούσιο, τριάδα-προς-τριάδα. Εξετάζοντας την κάθε τριάδα,

θα ελέγχει αν είναι σε γειτονική θέση απ' αυτό: αν είναι όντως, θα υπολογίζει σε ποια έδρα του αντιστοιχεί και θα ανάβει το αντίστοιχο LED και το αντίστοιχο `rogo pin` δεδομένων. Έτσι, αν τυχόν κάποιο άρτι συνδεδεμένο κυβίδιο δεν παρατηρήσει τάση σε κανέναν από τους ακροδέκτες λήψης δεδομένων, θα μπορέσει να καταλάβει ότι έχει συνδεθεί λάθος, και να το επικοινωνήσει αυτό στο χρήστη με κάποιον κατάλληλο τρόπο.

## 3.4 Υλοποίηση του κώδικα

### 3.4.1 Επιλογή γλώσσας

Για την υλοποίηση του κώδικα, επιλέχθηκε η γλώσσα προγραμματισμού Rust.

#### Πλεονεκτήματα:

- (+) Εύκολη στη συγγραφή
- (+) Ταχεία στην εκτέλεση
- (+) Εγγυημένη ασφάλεια μνήμης
- (+) Διαθέτει την επιλογή `no_std`, η οποία εγγυάται ότι ο κώδικας θα μπορεί να τρέξει σε μικροελεγκτή

#### Μειονεκτήματα:

- (-) Δύσκολη στην εκμάθηση (εμπόδιο εκ των προτέρων υπερβληθέν)
- (-) Δε συνεργάζεται εύκολα με τις πάγιες βιβλιοθήκες του Arduino
- (-) Η υποστήριξη των επεξεργαστών AVR είναι ακόμα σε βρεφικό στάδιο

### 3.4.2 Οργάνωση κώδικα

Ο κώδικας κατέληξε να διαχωριστεί σε 7 διαφορετικές βιβλιοθήκες, (ή «κιβώτια» όπως αποκαλούνται στη Rust,) οι οποίες συνολικά απαρτίζουν 13 αρχεία κώδικα. Το συνολικό μήκος, αγνοώντας κενές γραμμές, τεκμηρίωση, και σχόλια αλλά συμπεριλαμβάνοντας τους ελέγχους που γράφτηκαν για τη λειτουργικότητα, ανέρχεται περίπου στις 3500 γραμμές κώδικα.

Η συνοπτική ανάλυση των κιβωτίων έχει περίπου ως εξής:

- `skouria`: Επιτρέπει την πληκτρολόγηση διαφόρων λέξεων-κλειδιών της Rust στα Ελληνικά, με σκοπό την ευκολότερη συγγραφή και ανάγνωση του κώδικα για Ελληνόφωνους.
- `cube_rotations`: Αναλαμβάνει αποκλειστικά και μόνο τις πράξεις περιστροφής, οι οποίες εξυπηρετούν στην αναγνώριση του προσανατολισμού ενός κυβιδίου.
- `small_cubes`: Αποτελείται από 3 αρχεία, και αναλαμβάνει λίγο-πολύ όλα τα υπόλοιπα τμήματα του προγράμματος των κυβιδίων που δεν εξαρτώνται από το hardware: Ανίχνευση θέσης και προσανατολισμού, μετάδοση/λήψη σχεδίου, ανίχνευση θέσεων για ενδεχόμενα νέα εφαπτόμενα κυβίδια.
- `fundament`: Αναλαμβάνει όσα τμήματα του προγράμματος του θεμελίου είναι ανεξάρτητα από το hardware.



- `helpers`: Αποτελείται κι αυτό από 3 αρχεία, και αναλαμβάνει όσες βοηθητικές πράξεις εξαρτώνται από το hardware αλλά είναι κοινές και για το θεμέλιο και για τα κυβίδια. Αυτές είναι: Αναγνώριση συνδεδεμένης ακμής, υλοποίηση απαραίτητων μεθόδων για κάθε έδρα, πρωτόκολλο επικοινωνίας μεταξύ μικροελεγκτών, συνάρτηση πανικού<sup>2</sup>.
- `nano-tests`: Αξιοποιεί όλες τις παραπάνω βιβλιοθήκες για τον προγραμματισμό των κυβιδίων.
- `mega-fundament`: Αξιοποιεί όλες τις παραπάνω βιβλιοθήκες για τον προγραμματισμό του θεμελίου.

Ο κώδικας καθ' αυτός μπορεί να βρεθεί στην ιστοσελίδα [LINK ΕΔΩ](#), και η τεκμηρίωση μπορεί να βρεθεί στην ιστοσελίδα [LINK ΕΔΩ](#).

### 3.4.3 Λεπτομέρειες υλοποίησης

Προκειμένου να γίνουν κατανοητές κάποιες από τις σχεδιαστικές επιλογές του κώδικά μας, κρίνουμε δέον να κάνουμε μία μικρή παρένθετη αναφορά στην κυριότερη ειδοποίησή διαφορά της Rust: Την *ιδιοκτησία*.

#### Ιδιοκτησία και δανεισμός

Στη Rust αναγνωρίζεται ότι, πολύ συχνά, όταν ένα αντικείμενο δημιουργείται, υπάρχουν και κάποιες αρμοδιότητες που το συνοδεύουν. Ακριβώς όπως ένα κλειδί ενσωματώνει και συγκεκριμένες αρμοδιότητες για το σπίτι του οποίου την πόρτα ανοίγει, έτσι και τα αντικείμενα ενσωματώνουν διάφορες παρόμοιες αρμοδιότητες. Για παράδειγμα:

- Ένα αντικείμενο που αντιστοιχεί σε κάποια ενδεχόμενη δυναμική δέσμευση μνήμης έχει την αρμοδιότητα να δεσμεύει/αποδεσμεύει/αναδεσμεύει την εν λόγω μνήμη.
- Ένα αντικείμενο που αντιστοιχεί σε μία γραμμή I/O έχει την αρμοδιότητα της εγγραφής ή ανάγνωσης, αντιστοίχως.
- Ένα αντικείμενο που αντιστοιχεί σε έναν κινητήρα έχει την αρμοδιότητα της ρύθμισής του
- ...και πάει λέγοντας.

Δεδομένου λοιπόν ότι οι αρμοδιότητες αυτές δεν πρέπει ποτέ να *αντιγράφονται*, αλλά μπορούν μόνο να μεταβιβάζονται είτε προσωρινά είτε μόνιμα, η Rust αντιστοίχως καθορίζει ότι κάποια αντικείμενα δεν μπορούν να αντιγραφούν, παρά μόνο να μεταβιβαστούν είτε προσωρινά είτε μόνιμα. Αυτό ονομάζεται *ιδιοκτησία των εν λόγω αντικειμένων*, και η προσωρινή μεταβίβαση ονομάζεται *δανεισμός*.

---

<sup>2</sup>Στην ορολογία της Rust, «πανικός» σημαίνει ότι το πρόγραμμα «κρασάρει». Η ειδοποίησή διαφορά είναι ότι, «πανικός» αποκαλείται όταν το ίδιο το πρόγραμμα διαπιστώσει την ανάγκη να «κρασάρει», ενώ όταν αυτό διαπιστωθεί από το λειτουργικό σύστημα αυτό ονομάζεται «σφάλμα σελιδοποίησης» (“segmentation fault”).

### Κτήση περιφερειακών

Το σύστημα της ιδιοκτησίας και του δανεισμού αξιοποιείται κατά κόρον στη διεπαφή με τους μικροελεγκτές. Χάρη σ' αυτό, η αρμοδιότητα για το κάθε περιφερειακό (πχ τον ADC, την EEPROM, την κάθε γραμμή ξεχωριστά) δίδεται σε διαφορετικά τμήματα κώδικα, βεβαιώνοντας έτσι στατικά ότι δε θα υπάρξει ποτέ σύγκρουση στη χρήση τους.

Αυτό που δεν ήταν προφανές εκ των προτέρων, όμως, είναι ότι κάθε γραμμή εισόδου/εξόδου θεωρείται δικός της ξεχωριστός τύπος δεδομένων. Άμεση συνέπεια αυτού, είναι ότι κάθε έδρα του κυβιδίου θεωρείται κι αυτή τελίως ξεχωριστός τύπος δεδομένων, αφού «λαμβάνει την ιδιοκτησία» διαφορετικών γραμμών! Αν και αυτό θα μπορούσαμε να το είχαμε παρακάμψει κάνοντας το τελικό εκτελέσιμο ελαφρώς πιο αργό, εν τέλει αποφασίσαμε να δομήσουμε τον κώδικα ακριβώς έτσι ώστε να μην κάνουμε εκπτώσεις στην απόδοση.

Η επιλογή αυτή, παρ' ότι μας εγγυήθηκε τη μέγιστη απόδοση, είχε και ένα κόστος: Από τη στιγμή που ο κώδικας έπρεπε να μεταφραστεί για έξι διαφορετικούς τύπους δεδομένων, χρειάστηκε αναγκαστικά να εμφανιστεί στο τελικό εκτελέσιμο σε έξι διαφορετικά αντίγραφα, ένα για κάθε έδρα. Αυτό βέβαια δεν αποτέλεσε πρόβλημα εν τέλει, καθώς η μνήμη προγράμματος του Arduino Nano ήταν αρκετά μεγάλη από μόνη της.

#### 3.4.4 Πρωτόκολλο επικοινωνίας

Δεδομένου ότι οι πάγιες βιβλιοθήκες του Arduino δεν μπορούσαν να χρησιμοποιηθούν μέσω της Rust, εν τέλει επιλέξαμε να υλοποιήσουμε δικό μας πρωτόκολλο επικοινωνίας. Υλοποιήσαμε τόσο μία εκδοχή του πρωτοκόλλου RS-232 –όπου το κάθε bit κωδικοποιείται σε χαμηλή ή υψηλή *στάθμη*– όσο και μία εκδοχή του πρωτοκόλλου Manchester, όπου το κάθε bit κωδικοποιείται σε ανερχόμενη ή κατερχόμενη *παρυφή*. Εν τέλει το πρωτόκολλο Manchester αποδείχτηκε ταχύτερο, ενδεχομένως λόγω της αποφυγής αθροιστικών σφαλμάτων χρονισμού. Έτσι, επιλέξαμε να το χρησιμοποιήσουμε.

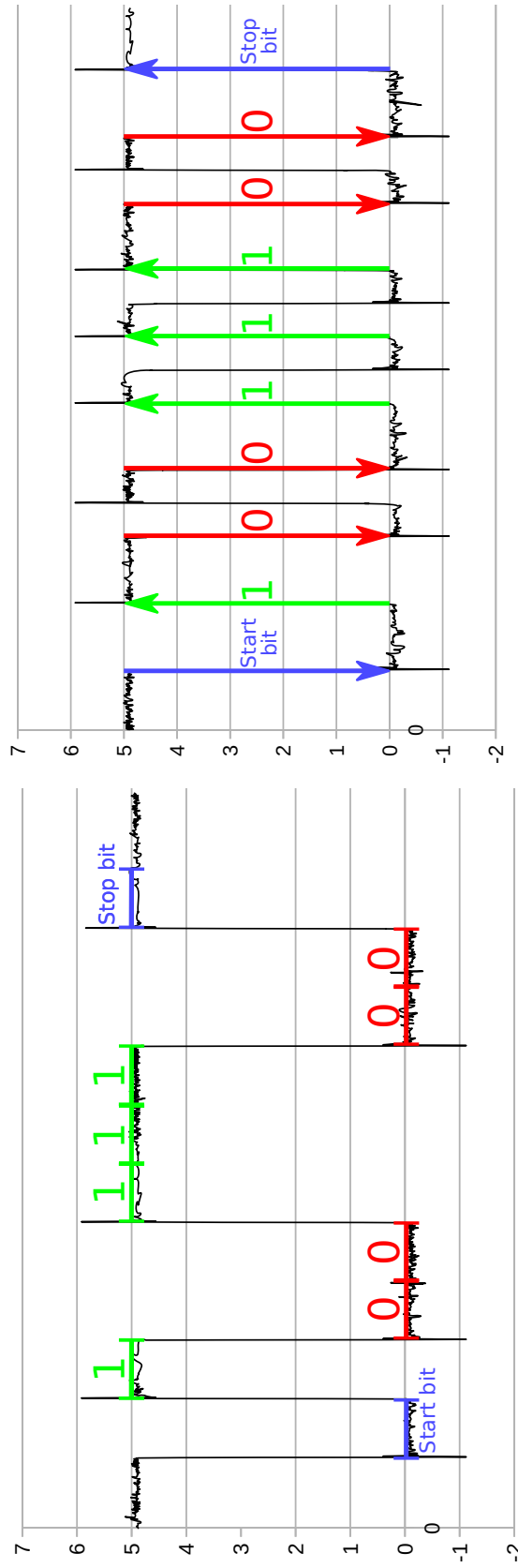
Παλμογραφήματα του ίδιου byte με τα 2 αυτά διαφορετικά πρωτόκολλα μπορούν να βρεθούν στο σχήμα 3.1. Προσοχή εφιστούμε στα εξής:

- Τα bit μεταδίδονται με ανάστροφη σειρά, ήτοι ξεκινώντας με το bit μικρότερης σημασίας.
- Πριν και μετά τη μετάδοση των καθ' αυτών bit δεδομένων, υπάρχει στην αρχή το bit έναρξης (start bit) το οποίο τοποθετεί τη γραμμή στο ενεργό 0, και το bit λήξης (stop bit) το οποίο τοποθετεί τη γραμμή στο αδρανές 1.

Το ωφέλιμο εύρος ζώνης (throughput) υπολογίζεται περίπου στα 50kbps-60kbps, αν και κάποιες πρωταρχικές δοκιμές μπόρεσαν να υπερβούν τα 200kbps.

#### 3.4.5 Χρήση της κάθε μνήμης

**Μνήμη προγράμματος:** Το τελικό εκτελέσιμο βγήκε περίπου ίσο με 20kiB, το οποίο δεν πλησιάζει καν το όριο των προγραμμάτων που μπορούν να χωρέσουν στη μνήμη flash του μικροελεγκτή, αφού αφήνει ελαφρώς πάνω από 10kiB ελεύθερα. Σημειωτέον ότι μέσα στα 20kiB αυτά συμπεριλαμβάνεται και η συνάρτηση πανικού, η οποία χρειάζεται μόνο για την ανάπτυξη του προγράμματος.



(Α) RS-232: κωδικοποίηση σε στάθμες

(Β) Manchester: κωδικοποίηση σε παραρφές

ΣΧΗΜΑ 3.1: Η μετάδοση του byte 0b00111001 με τα πρωτόκολλα RS-232 και Manchester.

**Μνήμη SRAM:** Επιλέξαμε να μη χρησιμοποιούμε καθ' όλου δυναμική δέσμευση μνήμης στο πρόγραμμά μας, ώστε να μπορούμε να έχουμε στατική βεβαιότητα για τη χρήση της μνήμης SRAM. Έτσι, μπορούμε να πούμε ότι η μέγιστη χρήση της ανέρχεται περίπου στα  $2/3$  της χωρητικότητάς της, κι αυτό με στατική δέσμευση αρκετής μνήμης για ένα σχέδιο 256 κυβιδίων.

**Μνήμη EEPROM:** Δεν αξιοποιήθηκε.

### 3.5 Προγραμματισμός θεμελίου

Έχοντας συγγράψει τον κώδικα που θα χρησιμοποιηθεί από τα κυβίδια, ο προγραμματισμός του θεμελίου είναι μάλλον εύκολος. Η μόνη διαφορά στον κώδικά του είναι ότι πρέπει με κάποιον τρόπο ο χρήστης να μπορέσει να επιλέξει το ποιο σχέδιο θα κατασκευάσει. Κατά τα άλλα, η ανίχνευση σύνδεσης σε κάποια βάση και η μετάδοση των απαραίτητων πληροφοριών γίνεται με τον ίδιο ακριβώς τρόπο όπως και στα κυβίδια. Μάλιστα, καθώς το θεμέλιο δε χρειάζεται ποτέ να λάβει αυτές τις πληροφορίες παρά μόνο να τις μεταδώσει, ο προγραμματισμός του καταλήγει αισθητά απλούστερος απ' αυτόν των κυβιδίων.

### 3.6 Διαγράμματα ροής

Τέλος, παρουσιάζουμε τα διαγράμματα ροής των κυριότερων ρουτινών που συγγράψαμε.

**Στο σχήμα 3.2** φαίνεται ο τρόπος μετάδοσης μεμονωμένων bit.

**Στο σχήμα 3.3** φαίνεται το πώς η μετάδοση bit αξιοποιείται για τη μετάδοση byte.

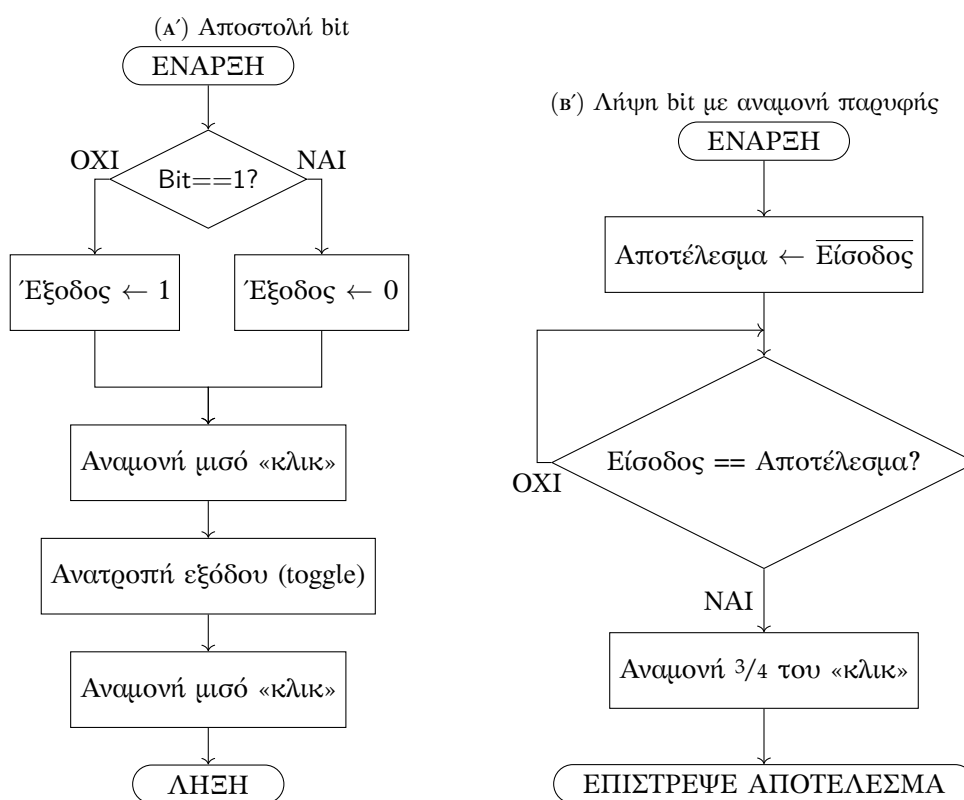
**Στο σχήμα 3.4** φαίνεται το πώς η μετάδοση των byte αξιοποιείται για τη μετάδοση όλων των απαραίτητων πληροφοριών.

**Στο σχήμα 3.5** φαίνονται τα διαγράμματα ροής του θεμελίου και του κυβιδίου.

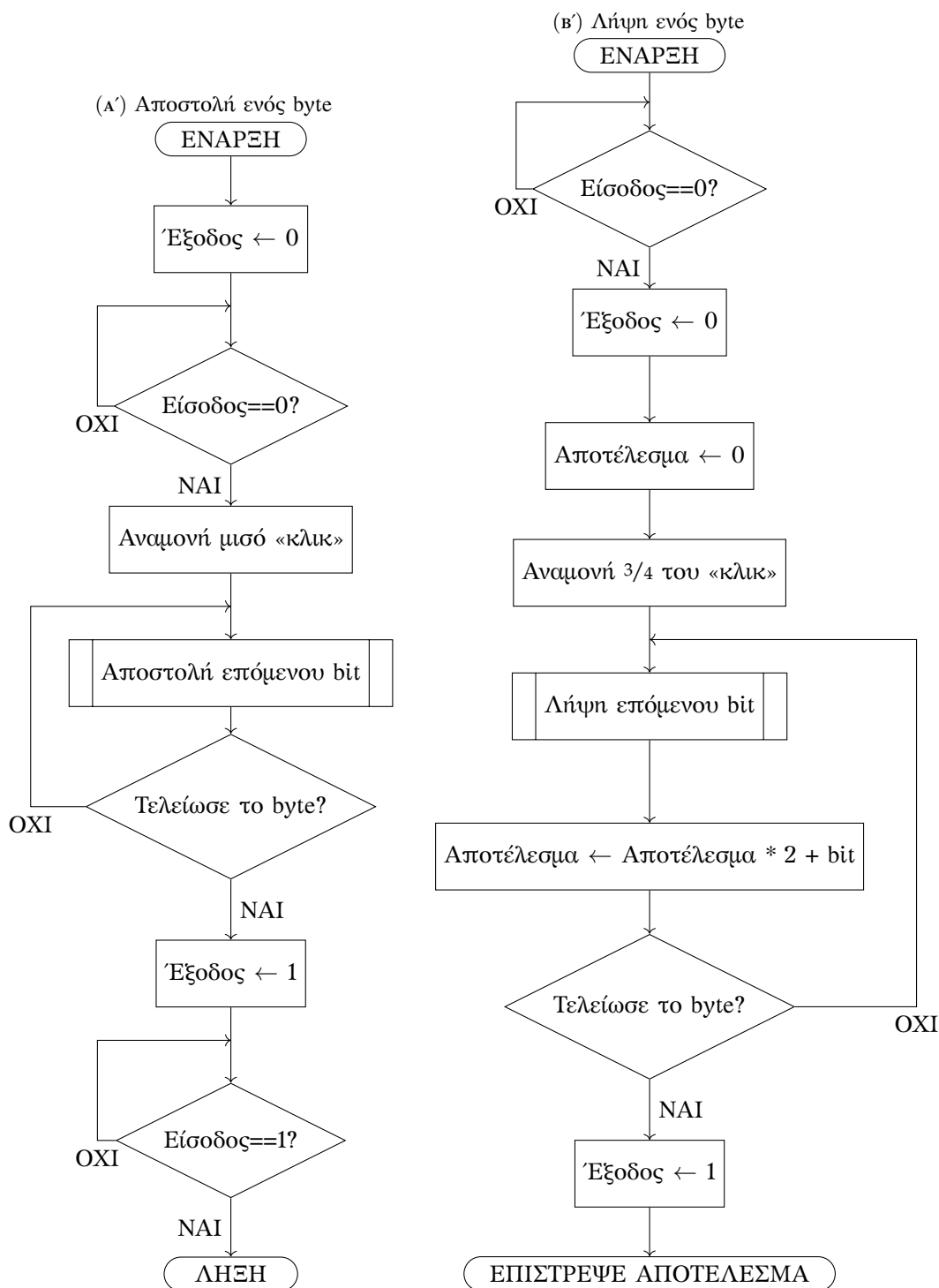
Δεν περιγράφηκαν αναλυτικά τα εξής:

- Το πώς δουλεύει ακριβέστερα η κωδικοποίηση των προσανατολισμών και των περιστροφών
- Το πώς ακριβώς η αναλογική ανάγνωση της εισόδου ανιχνεύει τη συνδεδεμένη ακμή (κάπως τετραμμένο)
- Το πώς λειτουργεί η συνάρτηση πανικού
- Το πώς μοντελοποιούνται στον κώδικα οι συντεταγμένες, το σχέδιο, και οι πλακέτες

Κάθε ενδιαφερόμενος μπορεί να ανατρέξει στην αναλυτικότερη τεκμηρίωση που γράφτηκε για τον κώδικα.

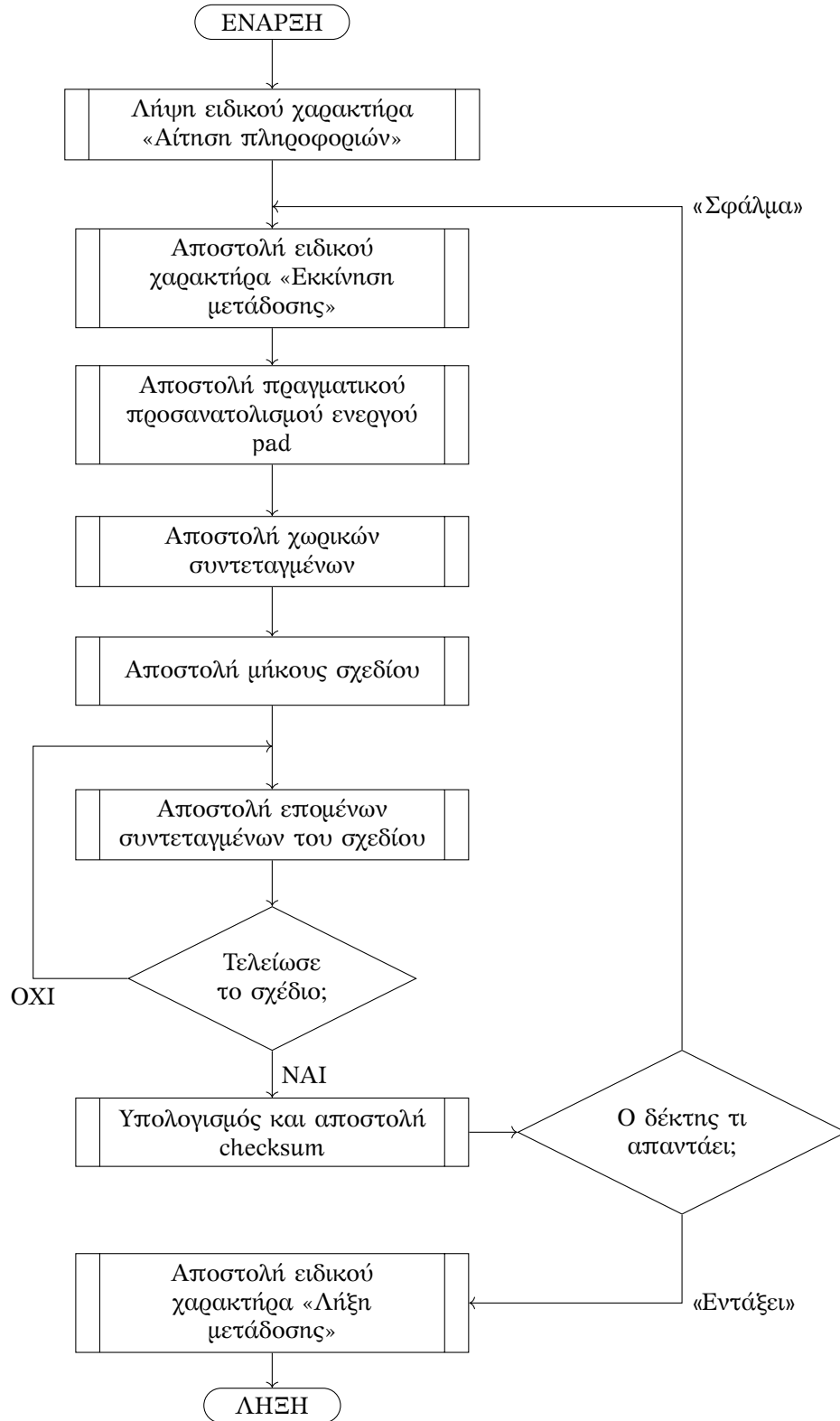


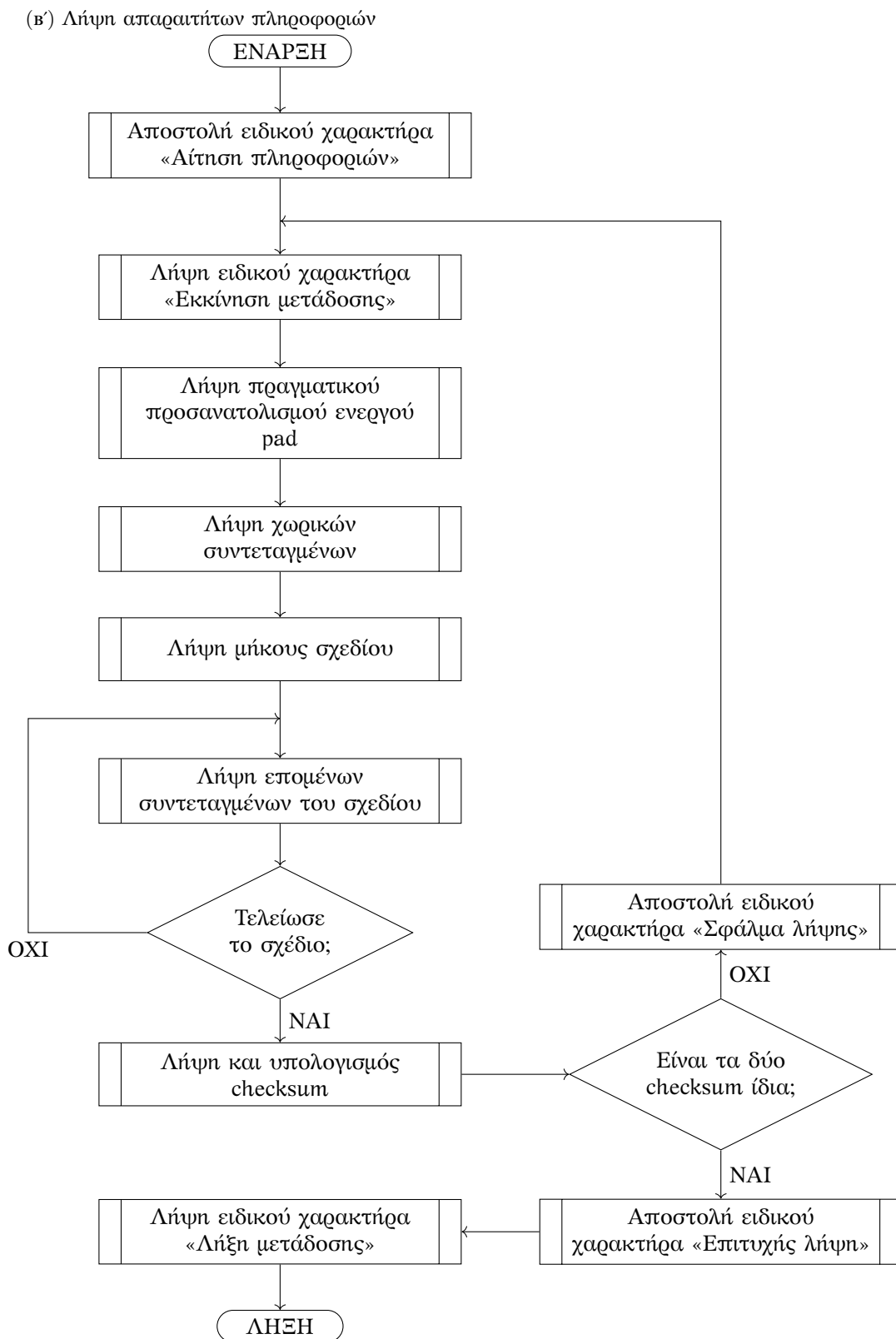
ΣΧΗΜΑ 3.2: Αποστολή και λήψη μεμονωμένων bit με βάση το πρωτόκολλο Manchester που εν τέλει χρησιμοποιήσαμε.



ΣΧΗΜΑ 3.3: Το πρωτόκολλο χαμηλού επιπέδου (bit-προς-bit για τη μετάδοση byte) που χρησιμοποιήθηκε.

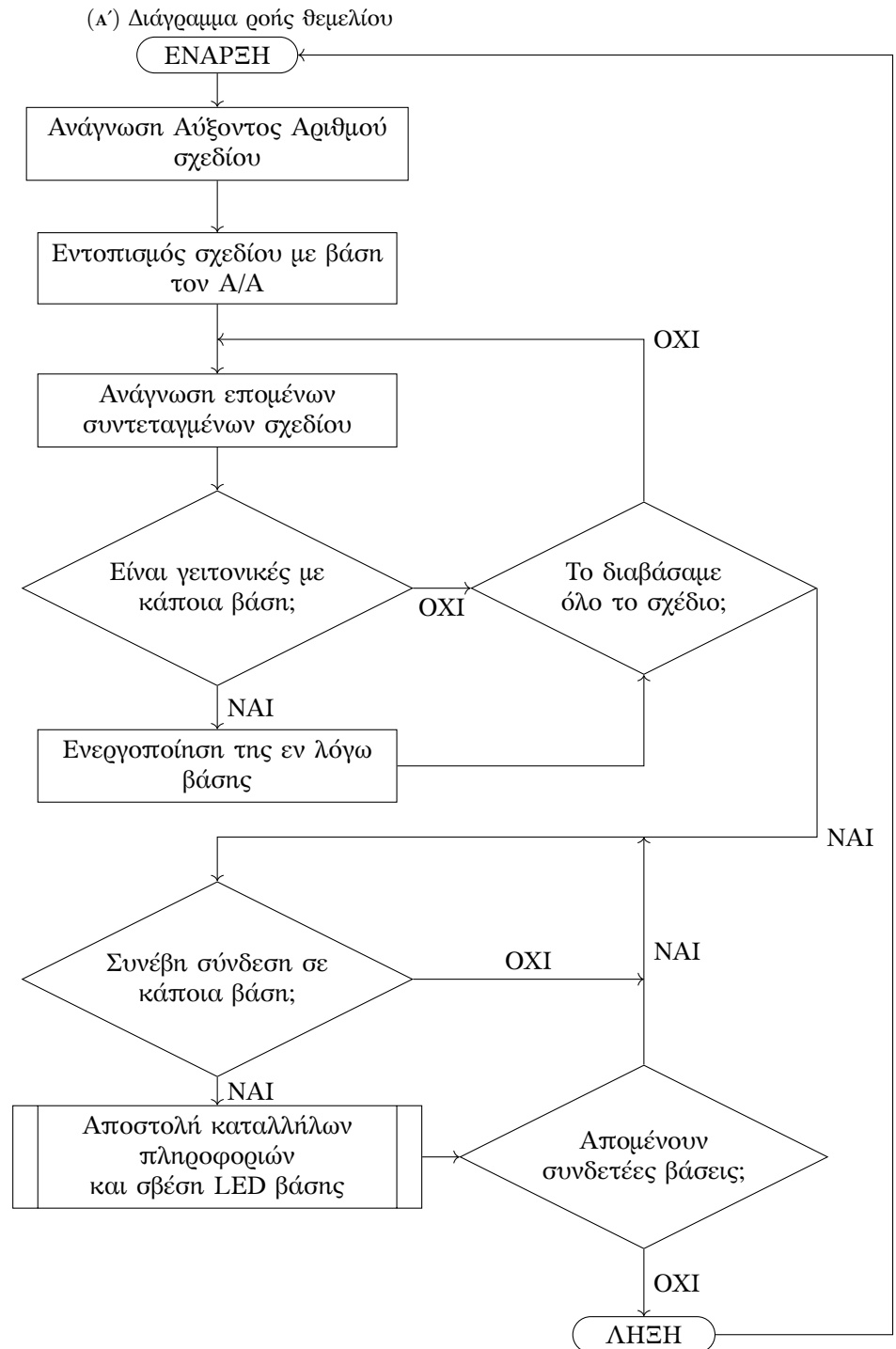
(Α') Αποστολή απαραίτητων πληροφοριών

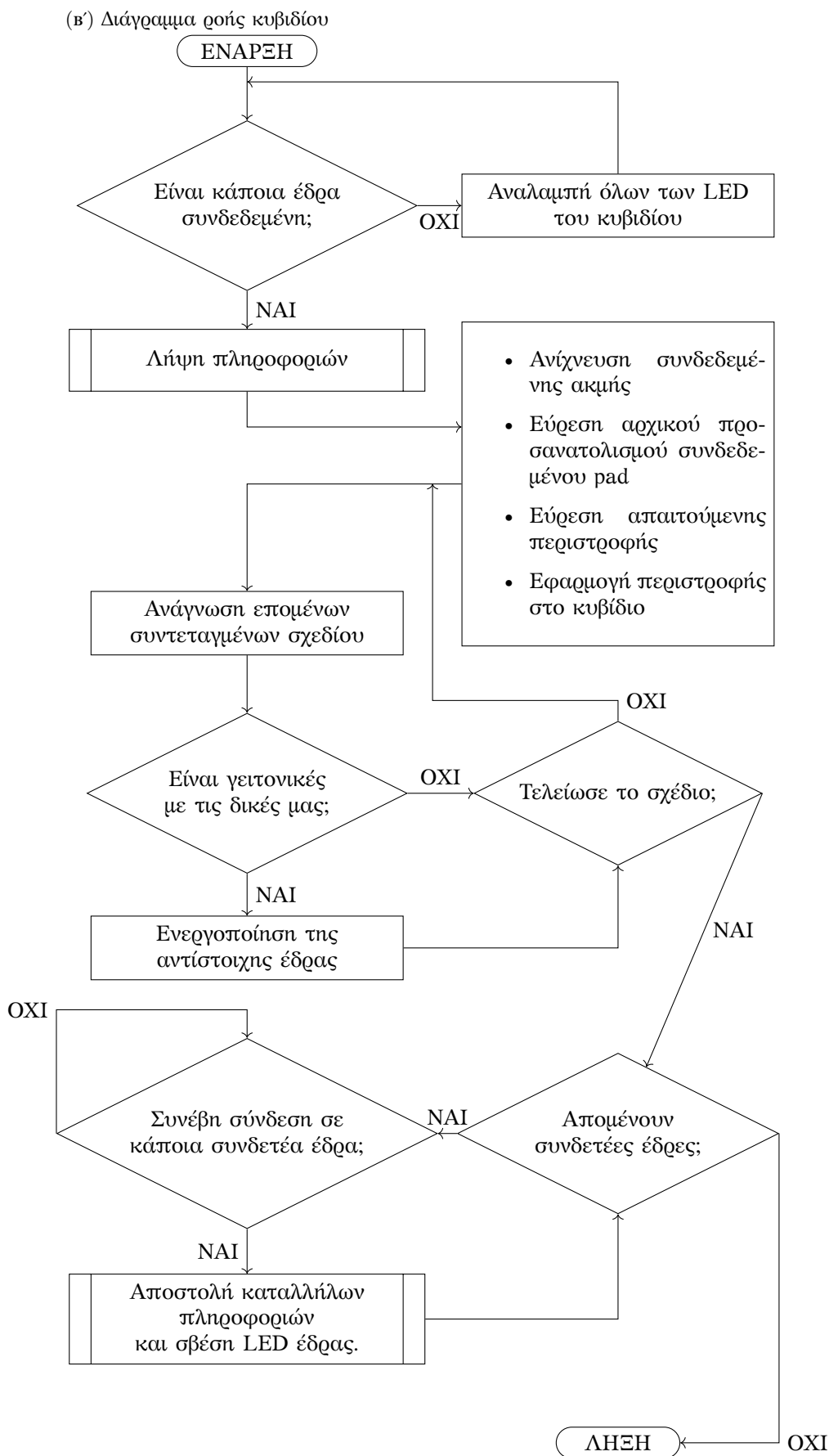




ΣΧΗΜΑ 3.4: Το πρωτόκολλο υψηλού επιπέδου (byte-προς-byte για τη μετάδοση όλων των δεδομένων) που χρησιμοποιήθηκε.







ΣΧΗΜΑ 3.5: Διαγράμματα ροής θεμελίου και κυβιδίων.

## Κεφάλαιο 4

# Κατασκευή, πειράματα, αξιολόγηση

### 4.1 Αρχικές δοκιμές

Η πρώτη κατασκευαστική μέθοδος που εξετάστηκε ήταν ο τεμαχισμός διάτρητων πλακετών, η προσαρμογή των καταλλήλων εξαρτημάτων σε αυτές, και η συναρμολόγησή τους. Αυτή η ιδέα είχε το πλεονέκτημα του ελάχιστου δυνατού παγίου κόστους, το οποίο συμφέρει για λίγα τεμάχια. Δυστυχώς, αυτό αποδείχτηκε δυσανάλογα χρονοβόρο και επιρρεπές σε σφάλματα συναρμολόγησης, και οδηγούσε σε πολύ κακής ποιότητας ηλεκτρικές συζεύξεις μεταξύ κυβιδίων. Έτσι επιλέξαμε να παραγγείλουμε την κατασκευή τους από ειδικό.

Μερικές φωτογραφίες από τις εν λόγω δοκιμές με διάτρητες πλακέτες μπορούν να φανούν στην ενότητα Α.3.

### 4.2 Κατασκευαστικές λεπτομέρειες

Όπως σκεπτόμασταν και αρχικά, κατασκευάστηκαν εν τέλει κάποιες μεμονωμένες πλακέτες, οι οποίες συναρμολογούμενες μεταξύ τους σχηματίζουν το εκάστοτε κυβίδιο· αυτό μας επιτρέπει να χρησιμοποιήσουμε και κάποιο αυθαίρετο πλήθος απ' αυτές ως βάσεις για το θεμέλιο.

#### 4.2.1 Εσωτερική πλευρά πλακετών

Ως τώρα, το μόνο που περιγράψαμε ήταν το πώς θα κατέλγαν να μοιάζουν οι πλακέτες εξωτερικά, καθώς αυτό ήταν μακράν το σημαντικότερο για τη λειτουργικότητα των κυβιδίων. Το τελικό προϊόν όμως αξιοποιεί αμφότερες τις πλευρές των πλακετών: Εξωτερικά τοποθετούνται οι μηχανικοί και ηλεκτρικοί συζευκτήρες, όπως και το LED· εσωτερικά τοποθετούνται όλα τα υπόλοιπα, όπως αντιστάτες, καλώδια, και προφανώς ο μικροελεγκτής.

Με βάση αυτό, πρέπει να τονιστεί το εξής: Ενώ οι έξι πλακέτες του κυβιδίου είναι πανομοιότυπες εξωτερικά, είναι κάθε μία διαφορετική εσωτερικά. Το εσωτερικό ανάπτυγμα του κυβιδίου φαίνεται στο σχήμα 4.1.

Όπως φαίνεται και στο σχήμα, λοιπόν, η πλακέτα με σήμανση «πάνω» είναι αυτή στην οποία τοποθετείται ο μικροελεγκτής, και συζευγνύεται τόσο ηλεκτρικά όσο και μηχανικά με τις υπόλοιπες πλακέτες μέσω ακιδοσειρών. Κάθε παλαιή πλακέτα χρειάζεται να συνδεθεί με την «πάνω» με τέσσερις γραμμές: τροφοδοσία, γείωση, είσοδος, έξοδος. Μόνη εξαίρεση είναι η πλακέτα με σήμανση «δεξιά», μέσω της οποίας περνούν και η είσοδος/έξοδος της πλακέτας με σήμανση «κάτω». Αυτό μπορεί να διαπιστωθεί και από το γεγονός ότι η

πλακέτα «δεξιά» είναι η μόνη που συνδέεται με την «πάνω» με πάνω από 4 ακίδες.

### 4.3 Πρώτα πειράματα

Το πρώτο βήμα, προφανώς, ήταν η συναρμολόγηση των κυβιδίων και του θεμελίου, και η επαλήθευση των ηλεκτρικών και μηχανικών συζεύξεων. Μία φωτογραφία δύο συναρμολογημένων κυβιδίων και ενός θεμελίου φαίνεται στο σχήμα 4.2.

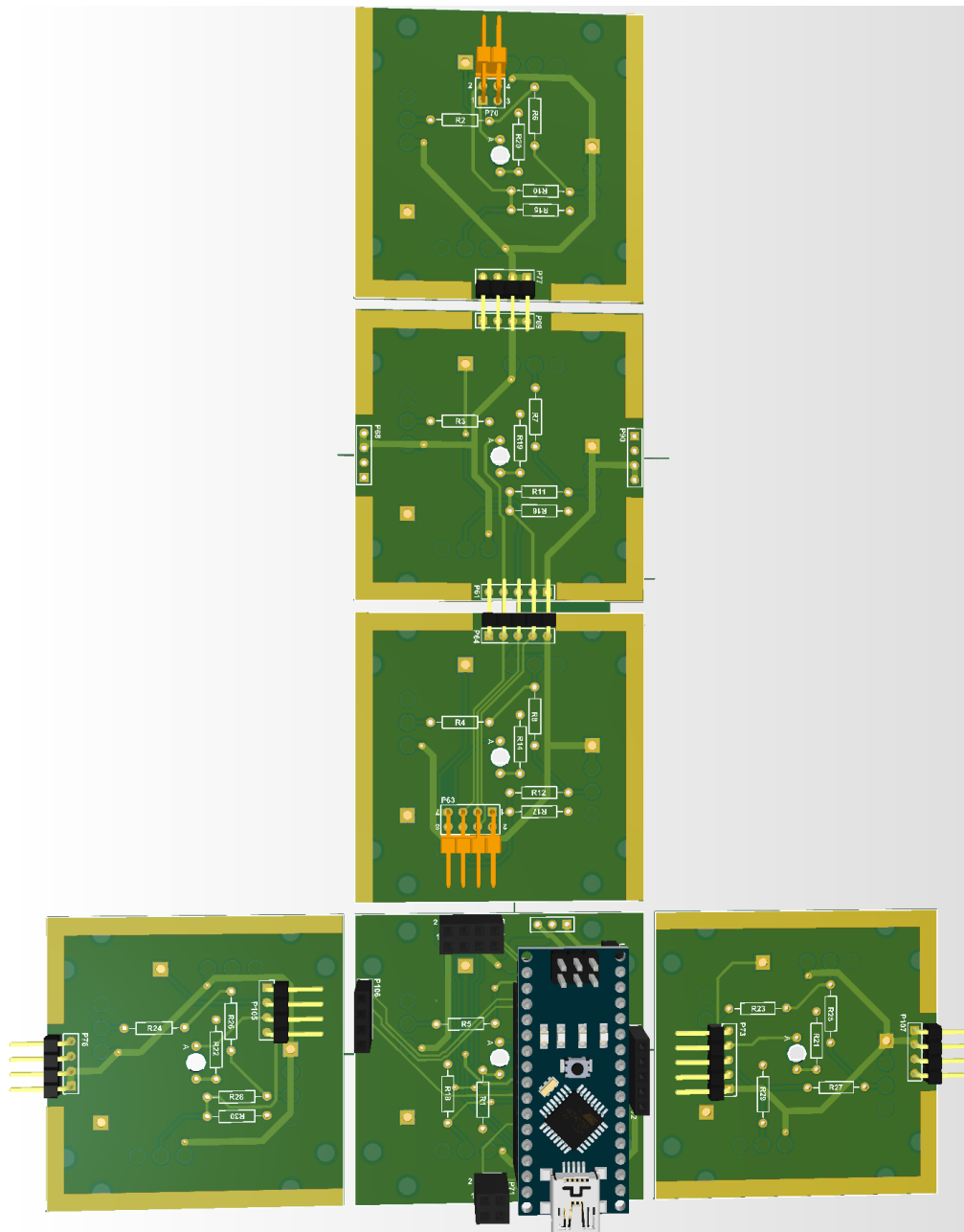
Προφανώς, από ένα πλήρως συναρμολογημένο κυβίδιο δεν μπορούσε να προεξέχει ο ενσωματωμένος Arduino Nano. Αυτό όμως σήμαινε ότι η θύρα USB του δε θα ήταν προσβάσιμη, και άρα ότι δε θα μπορούσε κανείς να το προγραμματίσει. Για το λόγο αυτό, τοποθετήθηκαν κάποιες ακιδοσειρές-προεκτάσεις που κρατούσαν την πάνω έδρα του κάθε κυβιδίου σε μία κάποια απόσταση απ' τις υπόλοιπες, ώστε η θύρα USB να είναι προσβάσιμη. Στο σχήμα 4.2 φαίνονται δύο κυβίδια, ένα «κλειστό» και ένα «ανοικτό» κατ' αυτόν τον τρόπο.

Στο ίδιο σχήμα φαίνεται και το θεμέλιο, το οποίο αποτελείται απλώς από έναν Arduino Mega 2560 τοποθετημένο σε ένα raster/breadboard. Σ' αυτόν έχει συνδεθεί μία έδρα ενός αποσυναρμολογημένου κυβιδίου, η οποία και αποτελεί τη μία και μοναδική βάση αυτού του θεμελίου.

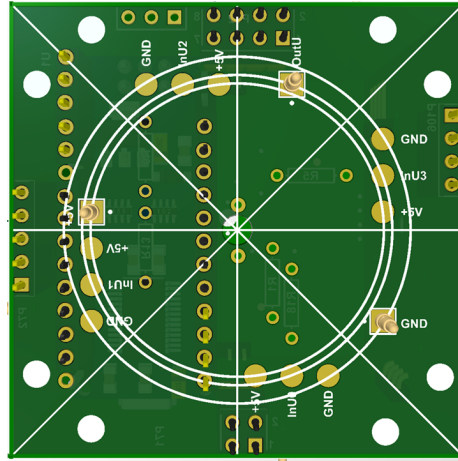
Αφότου η συναρμολόγηση των κυβιδίων και η επαλήθευση των συζεύξεων έγινε επιτυχώς, επόμενο βήμα ήταν η επαλήθευση του κώδικα που είχε γραφτεί.

Για το σκοπό αυτό, τα πειράματα που έγιναν ήταν με τη σειρά τα εξής:

1. Μετάδοση δεδομένων από το θεμέλιο προς το κυβίδιο, με προϋπάρχουσα σύνδεση
2. Ως άνω, αλλά με αναμονή μέχρι να συμβεί η σύνδεση.
3. Μετάδοση δεδομένων από το κυβίδιο προς το θεμέλιο, με προϋπάρχουσα σύνδεση
4. Ως άνω, αλλά με αναμονή μέχρι να συμβεί η σύνδεση.
5. Ηχώ: Αποστολή δεδομένων από το θεμέλιο στο κυβίδιο, και άμεση ανταποστολή τους από το κυβίδιο προς το θεμέλιο.
6. Εξερεύνηση της μέγιστης ταχύτητας με την οποία εξακολουθεί να δουλεύει η ηχώ, σύγκριση των πρωτοκόλλων μεταξύ τους, επιλογή του Manchester.
7. Ως άνω, αλλά με το κυβίδιο αποσυνδεδεμένο ώστε να μην τροφοδοτείται προτού συνδεθεί.
8. Μετάδοση σχεδίου και άλλων πληροφοριών, μόνο από το θεμέλιο προς το ένα κυβίδιο.
9. Μεταφορά του κώδικα ώστε να μπορεί να μεταδώσει τις πληροφορίες αυτές και από κυβίδιο σε κυβίδιο.

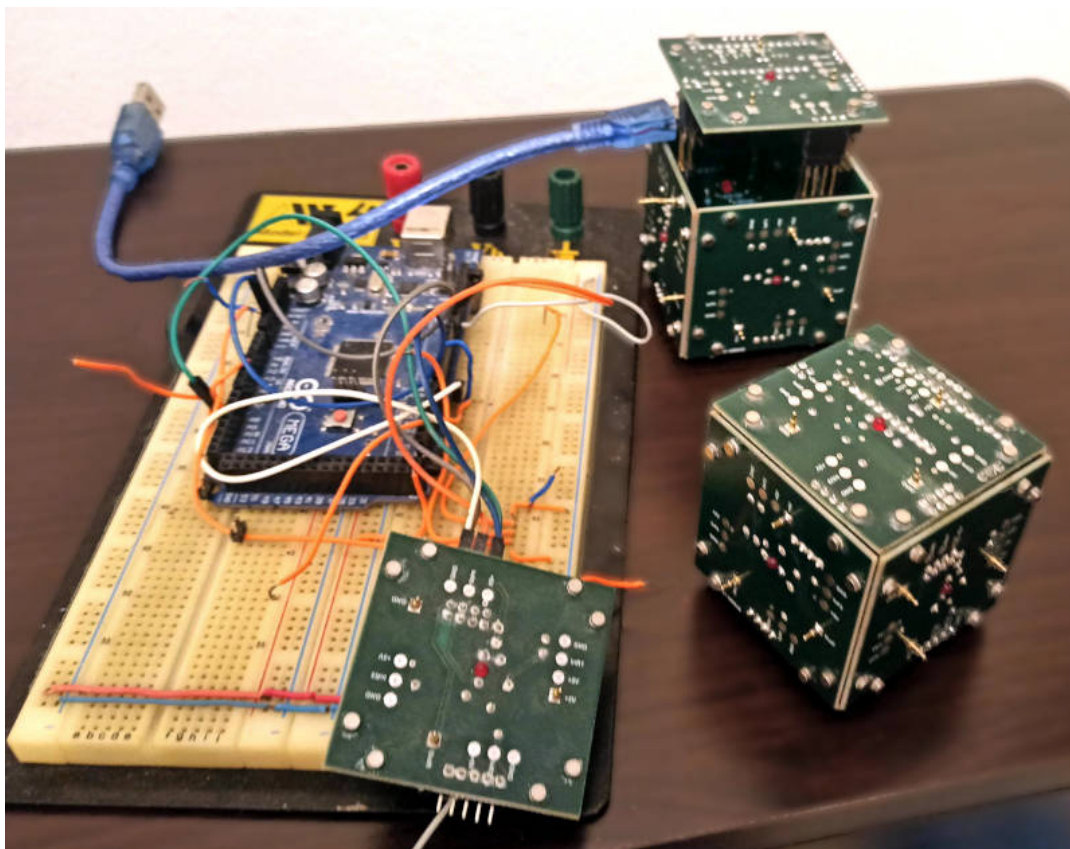


(Α') Εσωτερικό ανάπτυγμα



(Β') Εξωτερική πλευρά πλακέτας «πάνω»

ΣΧΗΜΑ 4.1: Αναπτύγματα κυβιδίου (εσωτερικά και εξωτερικά)



ΣΧΗΜΑ 4.2: Φωτογραφία δύο κυβιδίων και ενός θεμελίου

## 4.4 Τελική αποτίμηση επιλογών

### 4.4.1 Συμμετρία κυβιδίων

Όπως αναλύσαμε και νωρίτερα στην ενότητα 1.5, εν τέλει επιλέξαμε τα κυβίδια που θα κατασκευάσουμε να είναι πλήρως συμμετρικά. Αυτό οδήγησε σε μεγάλη πολυπλοκότητα στην κατασκευή, αλλά είχε τα εξής θετικά:

- Δεν απαιτείται πλέον από το παιδί να ξέρει αριθμούς ή χρώματα
- Δεν απαιτείται από το παιδί να διαβάζει κάποιο ξεχωριστό σχέδιο, καθώς η ίδια η κατασκευή το καθοδηγεί
- Αν χαθεί κάποιο κυβίδιο, αυτό δεν ενοχλεί καθ' όλου. Στην έσχατη, μπορεί ν' αντικατασταθεί με άλλο κυβίδιο από κάποιο άλλο kit
- Η αναγκαία καθοδήγηση από γονέα/δάσκαλο ελαχιστοποιείται

Με βάση αυτά, τείνουμε να αποφανθούμε ότι η επιλογή αυτή αποδείχτηκε επιτυχημένη.

### 4.4.2 Κατασκευαστικές επιλογές

#### Πλακέτες

Όπως προαναφέραμε νωρίτερα, εν τέλει απορρίψαμε ως ιδέα την τρισδιάστατη εκτύπωση ειδικών πλαστικών επιφανειών ως δομικό υλικό της κάθε έδρας. Αντ'

αυτού, οι ίδιες οι πλακέτες που εκτυπώθηκαν εξυπηρετήσαν αυτόν το σκοπό. Αυτό είχε τα εξής αποτελέσματα:

**Πλεονεκτήματα:**

- (+) Εξαιρετικά φτηνές στην ανάπτυξη και το σχεδιασμό
- (+) Επιτρέπουν την εύκολη πρόσβαση του πολυμέτρου στις επαφές

**Μειονεκτήματα:**

- (-) Οι πλακέτες δεν αντέχουν πολύ σε μηχανικές στρεβλώσεις, κάποιες φορές δεν ενώνονταν όλοι οι μαγνήτες μεταξύ τους
- (-) Οδηγούν σε πολλές αιχμηρές προεξοχές, κάτι ακατάλληλο για μικρά παιδιά
- (-) Δεν είναι ιδιαίτερα θελκτικές οπτικά
- (-) Στην κατασκευή είναι ακριβές χωρίς λόγο

Προτείνουμε λοιπόν για την τελική κατασκευή τη χρήση κάποιου καταλλήλου πλαστικού κελύφους.

### Μαγνήτες

Πέραν της προϋπάρχουσας ανάλυσης στην υπο-ενότητα 1.5, διαπιστώθηκαν για τους μαγνήτες τα εξής:

**Πλεονεκτήματα:**

- (+) Καθοδηγούν από μόνοι τους τη σωστή συναρμολόγηση

**Μειονεκτήματα:**

- (-) Οι μαγνήτες που επιλέξαμε παραήταν αδύναμοι για τη χρήση που θέλαμε
- (-) Δεδομένου ότι οι μαγνήτες προεξείχαν από τις πλακέτες και ήταν κολλημένοι σ' αυτές, η αποσύνδεση πλακετών μεταξύ τους οδηγούσε σε έντονες εφελκυστικές τάσεις στις κολλήσεις, με αποτέλεσμα συχνά οι μαγνήτες να ξεκολλούν από τις πλακέτες

Το πρώτο πρόβλημα θα μπορούσε προφανώς να επιλυθεί επιλέγοντας δυνατότερους μαγνήτες. Το δεύτερο πρόβλημα θα μπορούσε να επιλυθεί είτε χρησιμοποιώντας δυνατότερη κόλλα, είτε τοποθετώντας τους μαγνήτες πίσω από την πλακέτα ώστε οι τάσεις να είναι θλιπτικές, είτε φυσικά και τα δύο.

Πέραν τούτου, αν πρόκειται ως δομικό υλικό της κάθε έδρας να μη χρησιμοποιείται μόνο η ηλεκτρική πλακέτα, αλλά πχ ειδικά διαμορφωμένες πλαστικές επιφάνειες, θα μπορούσαν να έχουν εσοχές και εξοχές για τη μηχανική σύζευξη, καθιστώντας τους μαγνήτες καθαρά προαιρετικούς.

### Ακροδέκτες με ελατήρια (Pogo pins)

Τα μειονεκτήματα των pogo pins που διαπιστώθηκαν κατά τη χρήση ήταν τα εξής:

- (-) Δεν είναι πολύ στιβαρά, κάποια μεγάλη καταπόνηση μπορεί να τα στραβώσει
- (-) Αποτελούν μία ακόμα αιχμηρή προεξοχή

- (-) Έχουν έναν περιορισμό (έστω και μικρό) στο πόσο μεγάλο ρεύμα μπορούν να μεταδώσουν

Παρ' ότι υπάρχουν κάπως πιο κοντόχοντρα rogo pin, και άρα κάπως πιο φτηνά και στιβαρά, οφείλουμε να ομολογήσουμε ότι το κυριότερο κριτήριο για τη χρήση των rogo pins ως ηλεκτρικών συζευκτών είναι οι μεγάλες ανοχές στις κατασκευαστικές ατέλειες της κάθε έδρας. Εάν επρόκειτο να κατασκευαστούν οι έδρες με μικρότερες ανοχές, και άρα μεγαλύτερη κατασκευαστική τελειότητα, ενδέχεται τα rogo pin να μη χρειάζονταν καν.

### Δικτύωμα διαίρεσης τάσης

Όπως προαναφέραμε στην υπο-ενότητα 2.5.1, οι τιμές των αντιστατών για το δικτύωμα διαίρεσης τάσης επιλέχθηκαν περισσότερο με βάση την ελάχιστη δυνατή κατανάλωση ρεύματος στη μόνιμη κατάσταση. Αυτό μειώνει την ευρωστία της κατασκευής στο θόρυβο, το οποίο εν γνώσει μας αποδεχθήκαμε. Υπήρξε όμως και ένα μεγάλο μειονέκτημα σχετικό με τα μεταβατικά φαινόμενα, το οποίο δεν είχαμε προβλέψει.

Οι γραμμές που χρησιμοποιούσαμε είχαν κάποιες παράσιτες χωρητικότητες και αλληλεπαγωγές, οι οποίες δεν ήταν αμελητέες. Όταν αυτές μπορούσαν να φορτιστούν απ' ευθείας από τις εξόδους του μικροελεγκτή, το ρεύμα ήταν αρκετά μεγάλο ώστε να μη δημιουργούν προβλήματα. Όταν αυτές φορτίζονταν από την έξοδο του δικτυώματος, όμως, το ρεύμα θα μπορούσε να είναι μέχρι και 120 φορές μικρότερο. Αυτό δυσχέραινε πολύ την επικοινωνία μεταξύ κυβιδίων, όπως φαίνεται και στο παλμογράφημα του σχήματος 4.3.

Στο παλμογράφημα αυτό φαίνεται η μετάδοση ενός byte από το θεμέλιο προς ένα κυβίδιο, με το πρωτόκολλο Manchester. Με γαλανό χρώμα φαίνεται η τάση της εξόδου του θεμελίου, και με κόκκινο φαίνεται η τάση της εξόδου του κυβιδίου. Με πράσινο διακεκομμένο φαίνεται το σχήμα που θα έπρεπε ιδανικά να έχει η κόκκινη κυματομορφή.

Στο σημείο (α) φαίνεται η επαλήθευση από το δέκτη της λήψης του προηγούμενου byte. Στο σημείο (β) φαίνεται το bit αίτησης του δέκτη, και στο σημείο (γ) το bit έναρξης (start bit) του πομπού. Αντιστοίχως, στο σημείο (δ) είναι το bit λήξης (stop bit) του πομπού και στο (ε) το bit επαλήθευσης λήψης του δέκτη.

Παρατηρούμε ότι παρουσιάζονται έντονες καθυστερήσεις της κάθε παρυφής, όπως και παράσιτα της μίας γραμμής στην άλλη. Αυτό αποτέλεσε έναν περιοριστικό παράγοντα στη μέγιστη ταχύτητα μετάδοσης των δεδομένων, κάτι που διαπιστώθηκε πολύ μετά την κατασκευή των κυβιδίων.

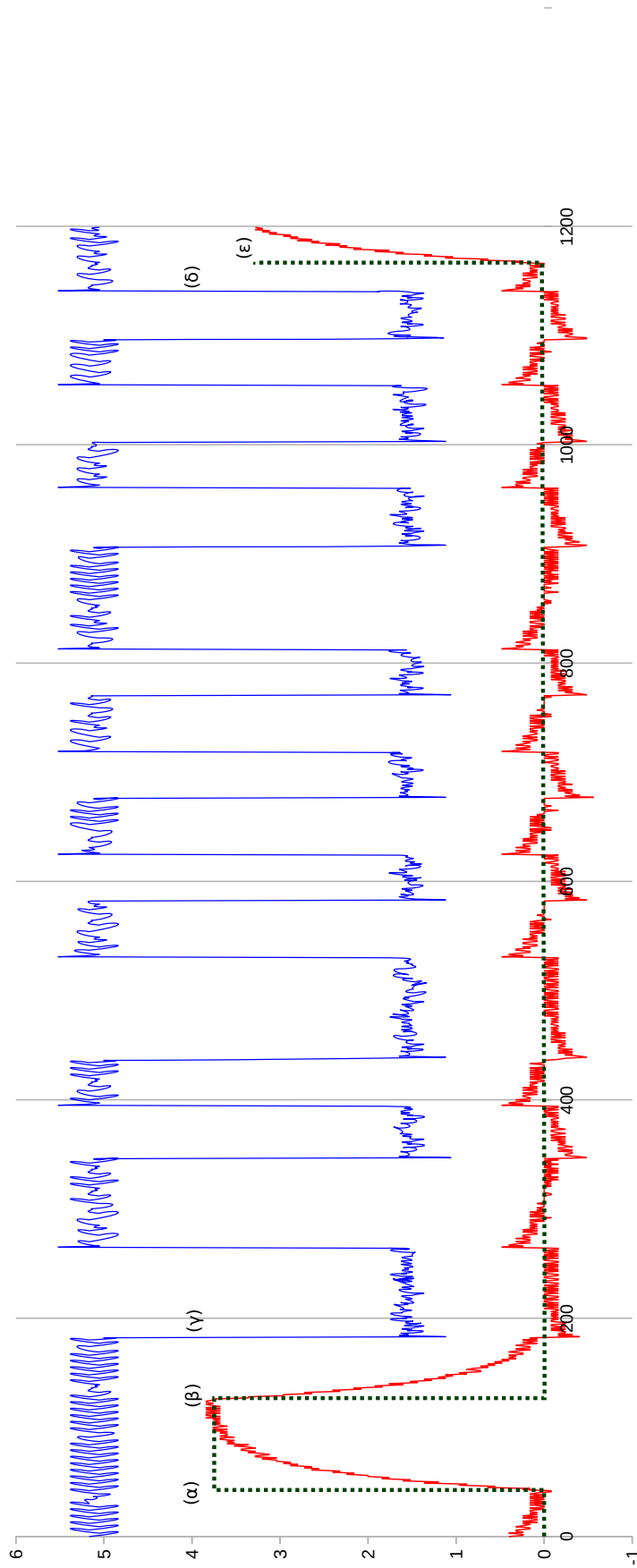
Με βάση όλα αυτά, λοιπόν, θεωρούμε ότι απ' ενός θα έπρεπε όλες οι αντιστάσεις να υποεικοσαπλασιαστούν, απ' ετέρου η τάση να διαιρείται λιγότερο ώστε το ρεύμα στην έξοδο του δικτυώματος να είναι μεγαλύτερο. (Φυσικά, για τη δική μας κατασκευή, όταν το πρόβλημα διαπιστώθηκε παραίταν αργά για τη διόρθωσή του μέσω hardware.)

### 4.4.3 Προγραμματισμός

#### Γλώσσα προγραμματισμού

Η επιλογή της γλώσσας προγραμματισμού Rust είναι αυτή για την οποία δεν είμαστε σίγουροι αν εν τέλει αποδείχτηκε εύστοχη ή άστοχη. Ο κυριότερος λόγος γι' αυτό είναι ότι γι' αυτήν αναγκαστήκαμε να γράψουμε δικό μας πρωτόκολλο





ΣΧΗΜΑ 4.3: Καθυστέρηση παρυφής και θόρυβος λόγω μικρών ρευμάτων

επικοινωνίας μεταξύ κυβιδίων, ενώ σε C θα μπορούσαμε να είχαμε χρησιμοποιήσει έτοιμες βιβλιοθήκες όπως τη SoftwareSerial. Παρ' όλ' αυτά, δεδομένου ότι η Rust γενικότερα οδηγεί σε πολύ λιγότερες δυσάρεστες εκπλήξεις σε σύγκριση με τη C, δεν μπορούμε να αποφανθούμε με βεβαιότητα για το αν εν τέλει μας εξυπηρετήσε καλύτερα ή όχι.

- (+) Τα γραφέντα προγράμματα μπορούν να μεταφερθούν σε πολλές άλλες αρχιτεκτονικές με ελάχιστες αλλαγές
- (+) Το κυβίδιο κατάφερε να «ακούει» όλες τις έδρες ταυτόχρονα (ενώ η βιβλιοθήκη SoftwareSerial θα μπορούσε να «ακούει» μόνο μία έδρα κάθε φορά)
- (=) Εξ ίσου ταχεία με τη C
- (=) Εκτελέσιμα εξ ίσου μικρά με αυτά της C
- (-) Πλήρης αδυναμία χρήσης των πάγιων βιβλιοθηκών που προσφέρονται έτοιμες για τους Arduino
- (-) Η αναλογική ανάγνωση μίας γραμμής αποκλείει κάθε μετέπειτα ψηφιακή χρήση της (πρόβλημα που εν τέλει απλώς παρακάμψαμε αντί να λύσουμε)

### Πρωτόκολλο επικοινωνίας

Η ανάγκη για την υλοποίηση του δικού μας πρωτοκόλλου επικοινωνίας, παρ' ότι οδήγησε μακράν στη μεγαλύτερη καθυστέρηση του χρονικού προϋπολογισμού, είχε και τα θετικά της. Χαρακτηριστικά, υπήρξαν δοκιμές στις οποίες το ωφέλιμο εύρος ζώνης στη μετάδοση των δεδομένων ήταν 200kbps. Θεωρούμε ότι, με τη χρήση κάποιου καταλληλότερου δικτύωματος διαίρεσης τάσης, η ταχύτητα θα μπορούσε να ξαναφτάσει σε αυτό το επίπεδο, και –γιατί όχι– να το υπερβεί.

## Κεφάλαιο 5

# Μελλοντικές βλέψεις

### 5.1 Κόστος κατασκευής

Δεδομένου ότι η εργασία αυτή προσβλέπει στην ανάπτυξη κάποιου προϊόντος που θα μπορούσε να διατεθεί στην αγορά, κρίνουμε σκόπιμη τη σύντομη ανάλυση του πιθανού κοστολογίου που θα χρειαζόταν για την υλοποίηση ενός τέτοιου προϊόντος με βάση τις σχεδιαστικές επιλογές που έχουν γίνει ως τώρα.

Για τους υπολογισμούς μας, θα υποθέσουμε ένα kit των 50 κυβιδίων. Οι τιμές λήφθηκαν από την ιστοσελίδα Mouser. Η υπο-ενότητα 5.1.5 περιέχει συνοπτικά τις πληροφορίες.

Η ανάλυση του κοστολογίου των πρωτοτύπων αυτών καθ' αυτών κρίνεται περιττή, καθώς είναι για πολλούς λόγους πολύ πιο ακριβά απ' όσο χρειάζεται. Για παράδειγμα, η χρήση εξαρτημάτων διάτρησης (through-hole) αντί για επιφανειακής στήριξης (surface-mounted) θα ήταν άσκοπη για την παρασκευή ενός τελικού προϊόντος, καθώς απαιτεί πολύ περισσότερο εμβαδό. Παρόμοιες διαφορές θα τονίζουμε και όπου αλλού εμφανίζονται.

#### 5.1.1 Μικροελεγκτές

Οι Arduino Nano κοστίζουν περίπου 6€-7€ ανά τεμάχιο, και άρα παρείναι ακριβοί για το σκοπό αυτό. Πέραν τούτου, είναι και πολύ ογκώδεις, και έτσι οδηγούν σε πολύ μεγάλα κυβίδια.

Το θέμα όμως είναι ότι οι Arduino Nano είναι μία πλακέτα με πολλά εξαρτήματα, τα περισσότερα εκ των οποίων θα μας ήταν άχρηστα. Απ' όλην την πλακέτα, τα μόνα που θα χρειαζόμασταν όντως θα ήταν τα εξής:

**Μικροελεγκτή ATmega328p**, με κόστος περίπου 67€ για το kit.

**Κρύσταλλο 16MHz**, προαιρετικά, με κόστος περίπου 8€ για το kit.

Ο κρύσταλλος χρειάζεται μόνο για την επιτάχυνση της επικοινωνίας μεταξύ κυβιδίων, και μπορεί να παραλειφθεί αν η ταχύτητα αυτή είναι ικανοποιητική ακόμα και με τη χρήση του εσωτερικού ταλαντωτή RC.

#### 5.1.2 Μαγνήτες

Στην παρούσα κατασκευή, οι μαγνήτες χρειάστηκε να τοποθετηθούν δύο-προς-δύο, επειδή τα rogo pin παραίταν μακριά και έτσι το ύψος των μαγνητών δεν αρκούσε. (Βλέπε σχήμα 2.5.) Αν υποθέσουμε ότι στην τελική κατασκευή θα τοποθετούνται μονοί, χρειάζονται 8 μαγνήτες ανά έδρα, \* 6 έδρες ανά κυβίδιο, \* 50 κυβίδια = 2400 μαγνήτες, προς 18€ περίπου.

### 5.1.3 Ακίδες

Παρά τις φιλότιμες προσπάθειες μείωσης της αναγκαίας ποσότητας, καταλήξαμε να χρειαζόμαστε 18 ακίδες για κάθε κυβίδιο. Για 50 κυβίδια, αυτό χρειάζεται 90 ακίδες, προς 22€ συνολικά.

### 5.1.4 Πλακέτες και πλαστικά κελύφη

Δυστυχώς, για τις παρούσες ανάγκες, το κόστος των δομικών στοιχείων των κυβιδίων δεν μπορεί να εκτιμηθεί. Για να γίνει μια τέτοια εκτίμηση, θα έπρεπε να γνωρίζει κανείς το ακριβές μέγεθος των κυβιδίων, για το οποίο δεν αποφαινόμαστε καθ' όλου.

Πέραν τούτου, είναι μία ακόμα μελλοντική κατασκευαστική επιλογή το αν πίσω από κάθε έδρα θα υπάρχει μία ξεχωριστή πλακέτα, ή αν θα υπάρχει μόνο μία πλακέτα η οποία θα συνδέεται με τις έξι έδρες μέσω αγωγών. Αυτό αποτελεί έναν ακόμα παράγοντα που δεν μπορούμε να συνυπολογίσουμε.

### 5.1.5 Σύνοψη

Ιδού η συνοπτική παρουσίαση του κοστολογίου που θα χρειαζόταν για την παρούσα κατασκευή:

Εξάρτημα	Πλήθος	Κόστος
Μικροελεγκτές	50	67€
Κρυσταλλικοί ταλαντωτές	50	8€
Μαγνήτες	2400	18€
Ακίδες	90	22€
Kit (κατ' ελάχιστον)	1	107€
Kit (κατά μέγιστον)	1	115€

Στην υπο-ενότητα 5.2.6 θα παρουσιαστεί ένας τρόπος μείωσης του κόστους κατά 27€ επί πλέον.

## 5.2 Πιθανές βελτιώσεις

Η παρούσα εργασία εξ αρχής στόχευε στην κατασκευή ενός πρωτοτύπου, όχι ενός προϊόντος ικανού να βγει αυτούσιο στην αγορά. Ως εκ τούτου, έχουν αφεθεί πολλές πιθανές μελλοντικές προεκτάσεις που δεν αξιοποιήθηκαν. Κάποιες απ' αυτές αναφέρθηκαν συνοπτικά νωρίτερα, αλλά εδώ θα αναφερθούν όλες μαζί συγκεντρωτικά.

### 5.2.1 Αναλογική ανάγνωση της γραμμής χωρίς περιορισμούς

Οι προεπιλεγμένες μέθοδοι αναλογικής ανάγνωσης των γραμμών I/O που διατίθεται στη γλώσσα Rust έχουν την παράπλευρη επίπτωση ότι, άπαξ και γίνει η αναλογική ανάγνωση, η γραμμή παραμένει συνδεδεμένη με τον ADC, και άρα μετά η γραμμή αυτή δεν μπορεί να ξαναχρησιμοποιηθεί ψηφιακά. Για το δικό μας πρόγραμμα, το μόνο που κάναμε επ' αυτού ήταν να μετακινήσουμε την αναλογική ανάγνωση στο τέλος του προγράμματος, ώστε να παρακάμψουμε αυτό το πρόβλημα.

Έχουμε κάθε λόγο να πιστεύουμε ότι, χρησιμοποιώντας Rust χαμηλού επιπέδου, μπορούμε το μετατροπέα analog-to-digital να τον χειριστούμε καταχωρητή-προς-καταχωρητή και bit-προς-bit, επιλύοντας έτσι αυτό το πρόβλημα ριζικά. Αυτό θα ήταν απαραίτητη προϋπόθεση για διάφορες από τις υπόλοιπες βελτιστοποιήσεις που παρατίθενται στην ενότητα αυτή.

## 5.2.2 Βελτιώσεις δικτύωματος διαίρεσης τάσης

### Επιτάχυνση επικοινωνίας

Η χρήση μεγάλων αντιστατών στο κύκλωμα διαίρεσης τάσης οδηγεί σε μεγάλα προβλήματα στις μεγάλες ταχύτητες μετάδοσης. Μειώνοντας τους αντιστάτες, τόσο στο άθροισμα όσο και στην αναλογία τους, θα μπορούσαμε να επιτύχουμε σημαντική αύξηση της ταχύτητας μετάδοσης των δεδομένων. Αυτό θα μπορούσε ενδεχομένως να καταστήσει περιττή τη χρήση των κρυστάλλων.

### Αναγνώριση αποσύνδεσης

Αναφέραμε στη σελίδα 18 ότι, με το παρόν δίκτυωμα διαίρεσης τάσης, δεν μπορεί να γίνει διαχωρισμός ανάμεσα στις καταστάσεις «έχει γίνει σύνδεση, αλλά διαβάζουμε λογικό 0» και «δεν υπάρχει καμία σύνδεση». Αυτό οφείλεται στο ότι η είσοδος προσδέεται στη γείωση.

Αν όμως προσδέσουμε το όλο δίκτυωμα και στην τροφοδοσία και στη γείωση, όπως φαίνεται στο σχήμα 5.1α', τότε γίνεται να διαχωρίσουμε την αποσυνδεδεμένη κατάσταση διαβάζοντας αναλογικά την είσοδο. Οι καταστάσεις θα είναι ως εξής:

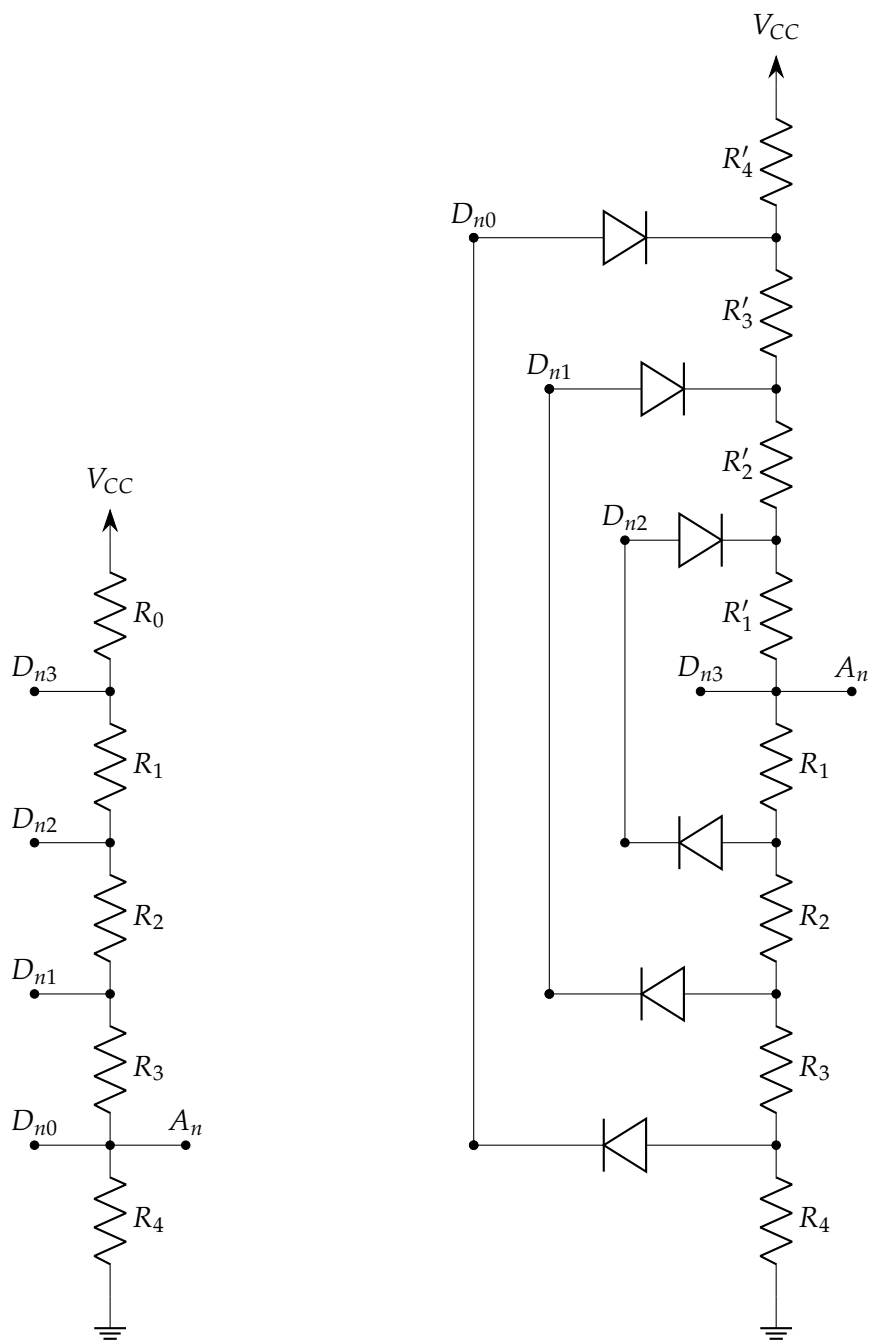
- Εάν η πλακέτα είναι αποσυνδεδεμένη, η τάση θα είναι 3.36V.
- Εάν η πλακέτα είναι συνδεδεμένη σε ακροδέκτη που δίνει λογικό 0, η τάση θα είναι 0V όπως και πριν. Θα αναγνωρίζεται ότι υπάρχει σύνδεση, αλλά όχι το ποιο pad είναι συνδεδεμένο.
- Εάν η πλακέτα είναι συνδεδεμένη σε ακροδέκτη που δίνει λογικό 1, η τάση θα κυμαίνεται όπως και πριν ανάμεσα στις τιμές 3.5V και 5V ανάλογα με τη συνδεδεμένη ακμή.

### Διαχωρισμός 9 καταστάσεων

Όταν μία έδρα διαβάζει αναλογικά την είσοδό της, οι πιθανές καταστάσεις είναι σύνολο εννέα: Τέσσερα pad επί 2 καταστάσεις, συν η πλήρης αποσύνδεση. Το αρχικό δίκτυωμα αναγνώριζε σύνολο 5 απ' αυτές, ενώ το άνω δίκτυωμα αναγνώριζε σύνολο 6.

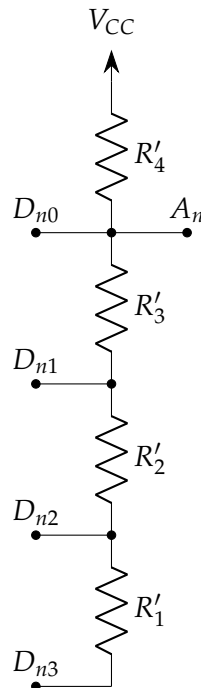
Υπάρχει όμως τρόπος να διαχωριστούν και οι εννέα: χρησιμοποιώντας ουσιαστικά εις διπλούν το διαιρέτη τάσης που αναλύθηκε στην υπο-ενότητα 2.5.1 στη σελίδα 17. Σ' αυτόν το διαιρέτη τάσης, όταν ένας συνδεδεμένος ακροδέκτης είναι σε υψηλή λογική στάθμη τότε μπορούμε να καταλάβουμε ποιος είναι, ενώ σε κάθε άλλη περίπτωση η αναγινωσκόμενη τάση είναι 0V.

Αντικατοπτρίζοντας αυτόν το διαιρέτη, κρατώντας τις τιμές των αντιστατών ακριβώς ίδιες, και συνδέοντάς τον στην τάση τροφοδοσίας  $V_{CC}$  αντί για τη γείωση, προκύπτει ο διαιρέτης τάσης που φαίνεται στο σχήμα 5.2. Εδώ, οι ρόλοι



(Α') Με αναγνώριση ακμής μόνο σε υ- (Β') Με αναγνώριση ακμής τόσο σε υ-  
ψηλή στάθμη

ΣΧΗΜΑ 5.1: Δικτυώματα διαίρεσης τάσης με αναγνώριση αποσύν-  
δεσης



ΣΧΗΜΑ 5.2: Αντικατοπτρισμός του διαιρέτη του σχήματος 2.10

υψηλής και χαμηλής στάθμης αντιστρέφονται: Όταν κάποιος συνδεδεμένος ακροδέκτης είναι σε χαμηλή λογική στάθμη τότε μπορούμε να καταλάβουμε ποιος είναι, ενώ σε κάθε άλλη περίπτωση η αναγινωσκόμενη τάση είναι 5V.

Παρατηρούμε ότι αμφότεροι οι διαιρέτες αυτοί είναι χρήσιμοι, καθ' ένας για διαφορετική περίπτωση. Το ιδανικό λοιπόν θα ήταν να μπορεί κανείς να θέτει τις υψηλές τάσεις στον ένα διαιρέτη και τις χαμηλές τάσεις στον άλλο, ώστε η αναγνώριση συνδεδεμένου ακροδέκτη να μπορεί να γίνεται ανεξαρτίτως λογικής στάθμης.

Μία πιθανή διαρρύθμιση για το πώς θα μπορούσε να επιτευχθεί αυτό φαίνεται στο σχήμα 5.1β'. Οι δίοδοι εξυπηρετούν ακριβώς τον προαναφερθέντα σκοπό, και οδηγούν την εκάστοτε λογική στάθμη στο σημείο που πρέπει ώστε να γίνει σωστά η αναλογική ανάγνωση. Συγκεκριμένα:

- Όταν η λογική στάθμη του ακροδέκτη είναι **υψηλή**, την οδηγούν στους **πάνω** αντιστάτες. Έτσι, οι αντιστάτες  $R_1$  έως και  $R_4$  καταλήγουν συνδεδεμένοι σε σειρά, λειτουργώντας ως ένας ενιαίος αντιστάτης πρόσδεσης στη γείωση. Αντιθέτως, οι αντιστάτες  $R'_1$  έως και  $R'_4$  λειτουργούν ως διαιρέτης τάσης, δίνοντας 4 διαφορετικές τιμές κοντά στα 5V.
- Όταν η λογική στάθμη του ακροδέκτη είναι **χαμηλή**, την οδηγούν στους **κάτω** αντιστάτες. Έτσι, οι αντιστάτες  $R'_1$  έως και  $R'_4$  καταλήγουν συνδεδεμένοι σε σειρά, λειτουργώντας ως ένας ενιαίος αντιστάτης πρόσδεσης στην τροφοδοσία. Αντιθέτως, οι αντιστάτες  $R_1$  έως και  $R_4$  λειτουργούν ως διαιρέτης τάσης, δίνοντας 4 διαφορετικές τιμές κοντά στα 0V.
- Όταν ο ακροδέκτης είναι εντελώς ασύνδετος, η τάση θα είναι 2.5V.

**Σημείωση:** Αν και ο μικροελεγκτής έχει ενσωματωμένους αντιστάτες πρόσδεσης προς την τροφοδοσία, δεν μπορούν να χρησιμοποιηθούν για τρεις λόγους:

1. Δε βολεύουν για τα παρόντα σχηματικά
2. Έχουν υπερβολικά μεγάλες τιμές (της τάξης των μυριάδων ohm)
3. Οι τιμές τους δεν είναι αυστηρά ορισμένες κατασκευαστικά, παρά μόνο «από 20kΩ έως και 50kΩ».

### 5.2.3 Διακοπές (interrupts)

Παρ' ότι κάποιες δοκιμές που κάναμε μπορούσαν να αξιοποιήσουν την ικανότητα του μικροελεγκτή να δέχεται διακοπές, το τελικό πρόγραμμα εν τέλει δεν τις χρησιμοποιεί καθ' όλου. Οι λόγοι ήταν κατά βάση δύο:

- Η έμφαση της Rust στο να έχει κάθε τμήμα δεδομένων έναν και μόνο έναν «ιδιοκτήτη» (βλ. 3.4.3), μαζί με την αδυναμία της ρουτίνας διακοπής να δέχεται ορίσματα ή να επιστρέφει αποτελέσματα, περιπλέκει απίστευτα την υλοποίηση του προγράμματος.
- Αν η ρουτίνα διακοπής δεν είναι πάρα πολύ μικρή, η κλήση/επιστροφή καταναλώνει πολύτιμο χρόνο τον οποίο δε βρήκαμε εν τέλει λόγο να διαθέσουμε.

Παρ' όλ' αυτά, η χρήση των διακοπών στο τελικό προϊόν θα μπορούσε να αξιοποιηθεί ώστε να επιτρέπει στο κυβίδιο να «κομάται» ενόσω δε χρειάζεται να λειτουργήσει και να «ξυπνά» μόλις συνδεθεί σ' αυτό κάποιο νέο κυβίδιο. Αυτό θα μείωνε πολύ την κατανάλωση ισχύος, και άρα και την καταπόνηση των ακροδεκτών τροφοδοσίας και γείωσης.

### 5.2.4 Πλαστικό περίβλημα

Η χρήση των Τυπωμένων Κυκλωματικών Πλακετών (PCB), παρ' ότι φτηνή, έχει σημαντικά μειονεκτήματα: Οδηγεί σε πολλές αιχμηρές προεξοχές, δεν είναι θελκτική οπτικά, και είναι ευάλωτες σε μηχανικές στρεβλώσεις οι οποίες ενδέχεται να μην επιτρέπουν τη σωστή μηχανική σύζευξη των κυβιδίων. Για όλους αυτούς τους λόγους, κρίνουμε σκόπιμη τη χρήση ενός πλαστικού περιβλήματος το οποίο θα επιλύει αυτά τα προβλήματα. Πέραν των προαναφερθέντων, θα επιτρέπει επί πλέον τη χρήση εσοχών και εξοχών ως μηχανικούς συζευκτήρες, είτε αντικαθιστώντας είτε συμπληρώνοντας τους μαγνήτες.

### 5.2.5 Ανίχνευση ολοκλήρωσης κατασκευής

Αυτήν τη στιγμή, δεν υπάρχει κάποιος τρόπος να επικοινωνηθεί στο χρήστη ότι όλη η κατασκευή έχει συναρμολογηθεί σωστά. Ο μόνος τρόπος να βγει αυτό το συμπέρασμα είναι να διαπιστώσει κανείς ότι κανένα από τα LED δεν είναι αναμμένο πλέον. Στο μέλλον, θα μπορούσε να αξιοποιηθεί η ικανότητα επικοινωνίας των κυβιδίων μεταξύ τους ώστε σιγά-σιγά να φτάσει στο θεμέλιο η πληροφορία της επιτυχούς κατασκευής, ώστε λχ να εμφανιστεί κάποιο κατάλληλο συγχαρητήριο μήνυμα.



### 5.2.6 Φτηνότεροι μικροελεγκτές

Νωρίτερα, υποθέσαμε ότι η κατασκευή θα αξιοποιήσει μικροελεγκτές ATmega328p. Αυτοί οι μικροελεγκτές έχουν διάφορα θετικά:

- Για τα αρχικά πειράματα, μπορούν να προγραμματιστούν εύκολα μέσω των Arduino Nano
- Η SRAM τους είναι αρκετά μεγάλη ώστε να χωρέσει μεγάλα σχέδια
- Έχουν πολλές γραμμές, οπότε κάθε έδρα μπορεί να συνδεθεί απ' ευθείας στο μικροελεγκτή

Το αρνητικό τους, όπως είδαμε και νωρίτερα, είναι ότι είναι μακράν το ακριβότερο εξάρτημα που χρειάζεται για την κατασκευή των κυβιδίων. Κατά συνέπεια, εάν θα γινόταν να αξιοποιηθούν φτηνότεροι μικροελεγκτές, αυτό θα μπορούσε να μειώσει δραματικά το κόστος.

#### Μικροελεγκτές ATtiny402

Οι φτηνότεροι μικροελεγκτές που βρήκαμε στην αγορά είναι αυτοί της σειράς ATtiny, και συγκεκριμένα ο ATtiny202 και ο ATtiny402. Αξιόλογη διαφορά στην τιμή δε φαίνεται να έχουν: Κοστίζουν αμφότεροι περίπου 0.41€ ανά τεμάχιο. Εφ' εξής θα λειτουργούμε με βάση τον ATtiny402. Για συντομία, ο ATmega328p θα αποκαλείται εφ' εξής "Mega" και ο ATtiny402 αντιστοίχως "Tiny".

Το πρώτο πρόβλημα που θα παρουσιαζόταν στη χρήση των Tiny είναι ότι δεν έχουν πολλές γραμμές I/O: Όλες μαζί είναι 8 τον αριθμό, οπότε δεν μπορεί κάθε έδρα να συνδεθεί στο μικροελεγκτή απ' ευθείας. Για να μπορέσουν να συνδεθούν, θα αξιοποιήσουμε απο/πολυπλέκτες.

#### Απο/πολυπλέκτες

Η μόνη λύση για να μπορέσουν να συνδεθούν 6 πλακέτες σε έναν Tiny είναι να συνδεθούν οι όμοιες γραμμές σε απο/πολυπλέκτες 8-προς-1: Αναλογικό πολυπλέκτη για τις εισόδους, αποπολυπλέκτες για τις εξόδους. Έτσι, θα χρειαζόμασταν έξι γραμμές: Μία για τα LED, μία για τα pin, μία για τα rad, και τρεις για την επιλογή της έδρας. Μαζί με την τροφοδοσία και τη γείωση, προκύπτουν ακριβώς οι 8 συνολικά γραμμές του μικροελεγκτή. Μάλιστα, η χρήση των γραμμών είναι τόσο οριακή που, ακόμα και να θέλαμε να βάλουμε εξωτερικό κρυσταλλικό ταλαντωτή, δεν περισσεύει γραμμή για να τον συνδέσουμε!

Η επιλογή «Tiny + από/πολυπλέκτες», αν επρόκειτο να υλοποιηθεί με διακριτά εξαρτήματα, θα χρειαζόταν περισσότερα ολοκληρωμένα. Αξιοσημείωτο είναι όμως ότι, ακόμα και σ' αυτήν την περίπτωση, η τιμή είναι σημαντικά μειωμένη σε σύγκριση με τους Mega: Αυτοί κοστίζουν ελαφρώς πάνω από 1.3€ ανά τεμάχιο, ενώ ένας Tiny και τρεις απο/πολυπλέκτες χρειάζονται  $0.41 + 3 * 0.13 = 0.8€$  ανά κυβίδιο, ήτοι 40€ για όλο το κιτ, ήτοι 27€ λιγότερα απ' τον προηγούμενο προϋπολογισμό.

Ας αναλυθεί τώρα η χρήση των μνημών.

#### Μνήμη προγράμματος (Flash)

Είδαμε νωρίτερα ότι η χρήση της μνήμης προγράμματος στους Mega ανερχόταν στα 20kiB, ένα νούμερο που δε χωράει με τίποτε στη μνήμη προγράμματος ενός Tiny. Αυτό όμως είναι παραπλανητικό.

Ο λόγος που το πρόγραμμα βγήκε τόσο μεγάλο για τους Mega είναι επειδή εμείς επιλέξαμε να δημιουργήσουμε ουσιαστικά 6 αντίγραφα του, ένα για κάθε έδρα. Αν επρόκειτο οι έδρες να οδηγούνται από τις ίδιες ακριβώς γραμμές, θα χρειαζόταν αποκλειστικά και μόνο ένα αντίγραφο. Ακόμα καλύτερα, αν την επικοινωνία με τους άλλους μικροελεγκτές θα επρόκειτο να την αναλαμβάνουν κάθε φορά οι ίδιες γραμμές, θα μπορούσαμε να χρησιμοποιούμε την ενσωματωμένη θύρα USART και έτσι να έχουμε hardware πρωτόκολλο επικοινωνίας αντί για software. Αυτό θα υποπολλαπλασίαζε την αναγκαία μνήμη προγράμματος, σε σημείο να χωράει άνετα στα 4kiB του Tiny.

### Μνήμη SRAM

Λειτουργώντας με τους Mega, είχαμε αφιερώσει χώρο στην SRAM αρκετό για 256 κυβίδια. Κάθε κυβίδιο ήθελε μία τριάδα από προσημασμένους 8άbitους ακεραίους για να αναπαρασταθεί, οπότε συνολικά μόνο για το σχέδιο αφιερώσαμε 768 byte—από μόνο του, τρεις φορές μεγαλύτερο απ' ότι ολόκληρη η SRAM του Tiny.

Οι πιθανές λύσεις που μπορούμε να σκεφτούμε επ' αυτού είναι οι εξής:

**Μείωση μήκους σχεδίου:** Αφιερώνοντας αρκετό χώρο πχ για 50 κυβίδια αντί για 256, αμέσως υποπενταπλασιάζεται η απαραίτητη μνήμη.

**Καλύτερη συμπίεση συντεταγμένων:** Αφιερώνοντας 2 byte σε κάθε κυβίδιο αντί για 3, θα μπορούσαμε να μειώσουμε την αναγκαία μνήμη SRAM για το σχέδιο κατά το  $1/3$ .

**Αξιοποίηση της EEPROM:** Αν επρόκειτο η SRAM να υπερχειλίζει, θα μπορούσαμε απλώς να γράψουμε τις υπόλοιπες συντεταγμένες στην EEPROM. Αυτή είναι πολύ αργή, και άρα ελαφρώς ακατάλληλη για γρήγορη επικοινωνία. Αλλά, προσφέρει επί πλέον 128 διαθέσιμα byte.

**Επανασυγγραφή του προγράμματος σε Assembly:** Έχοντας αυστηρότερο έλεγχο του τι τοποθετείται στη στοίβα, μπορούμε να είμαστε σίγουροι ότι αυτή δε θα υπερχειλίζει.

Εφαρμόζοντας και τις τρεις αυτές λύσεις, καταλήγουμε να χρειαζόμαστε 100 + 128 byte για το σχέδιο και 156 για την υπόλοιπη στοίβα, το οποίο κρίνεται αρκετό αν και κάπως οριακό.

Οι βελτιώσεις που παρουσιάστηκαν στην υποενότητα αυτή κατεβάζουν σημαντικά το κόστος της κατασκευής. Παρ' όλ' αυτά, η υλοποίησή τους κρίθηκε εκτός θέματος για το παρόν πόνημα, καθώς θα ξέφευγε τόσο από οικονομικής πλευράς όσο και από χρονικής πλευράς από τα πλαίσια μιας διπλωματικής εργασίας.

### 5.2.7 Έλεγχος τροφοδοσίας κυβιδίων

Το παρόν πρόβλημα, αν και υπαρκτό, δεν είναι μεγάλο.

Στην παρούσα κατασκευή υποθέτουμε ότι, όταν το θεμέλιο επιλέξει το σχέδιο προς κατασκευή, δε θα είναι σε αυτό συνδεδεμένο κανένα κυβίδιο. Αν η υπόθεση αυτή αποδειχθεί ψευδής, όμως, το πρόγραμμα δε θα δουλεύει!

Αυτό θα μπορούσε να επιλυθεί πολύ κομψά απλώς επιτρέποντας στο θεμέλιο να κόβει την τροφοδοσία των βάσεων του για ένα μικρό χρονικό διάστημα, αναγκάζοντας έτσι τα κυβίδια να κάνουν reset θέλοντας και μη, και φέρνοντάς τα πάλι στην αρχή του προγράμματος ώστε η υπόλοιπη σύνδεση να γίνει ομαλά.

### 5.2.8 Μαζική παραγωγή

Ως τώρα, όλοι οι υπολογισμοί που κάναμε υπέθεταν ότι η όλη κατασκευή θα πρέπει να μπορεί να κατασκευαστεί από διακριτά εξαρτήματα. Αυτό δεν είναι απαραίτητο: αν επρόκειτο για μαζική παραγωγή, θα μπορούσαν ενδεχομένως να αξιοποιηθούν FPGA ή ακόμα και ASIC για τη μείωση του κόστους.



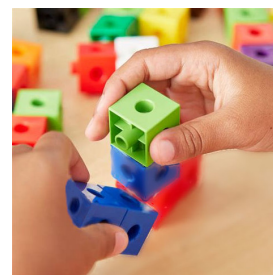
## Παράρτημα Α΄

# Φωτογραφίες άλλων λύσεων

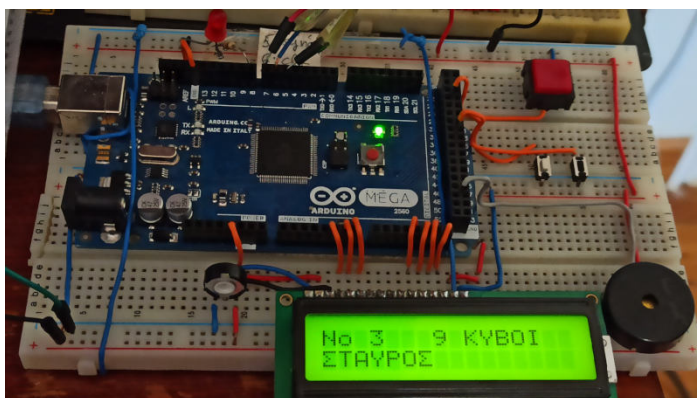
### Α΄.1 Μονοκονδυλιά



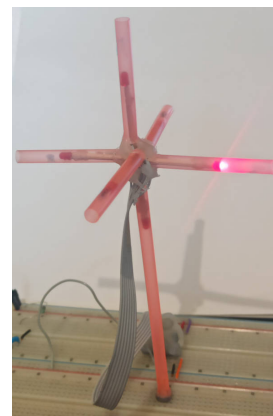
ΣΧΗΜΑ Α΄.1: Καθοδηγητικά μηνύματα στην οθόνη LCD



ΣΧΗΜΑ Α΄.2: Διαδικασία κατασκευής

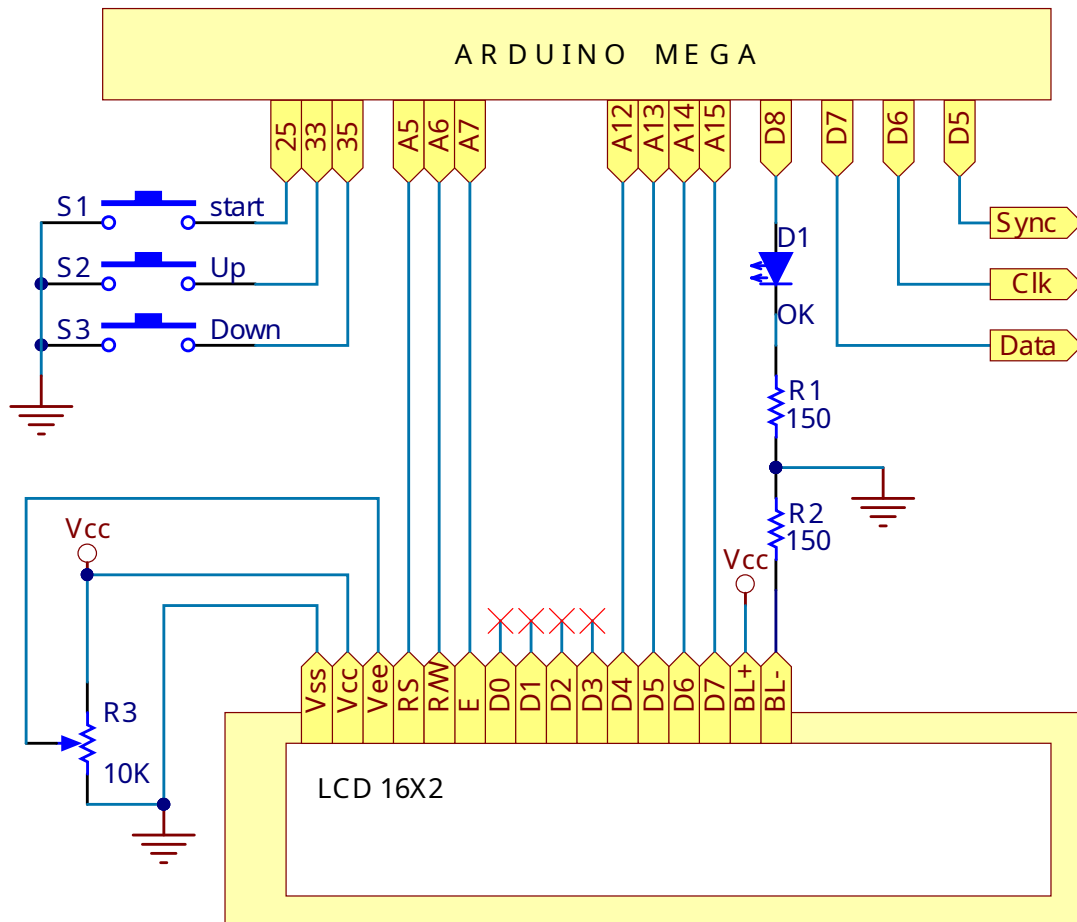


ΣΧΗΜΑ Α΄.3: Θεμέλιο

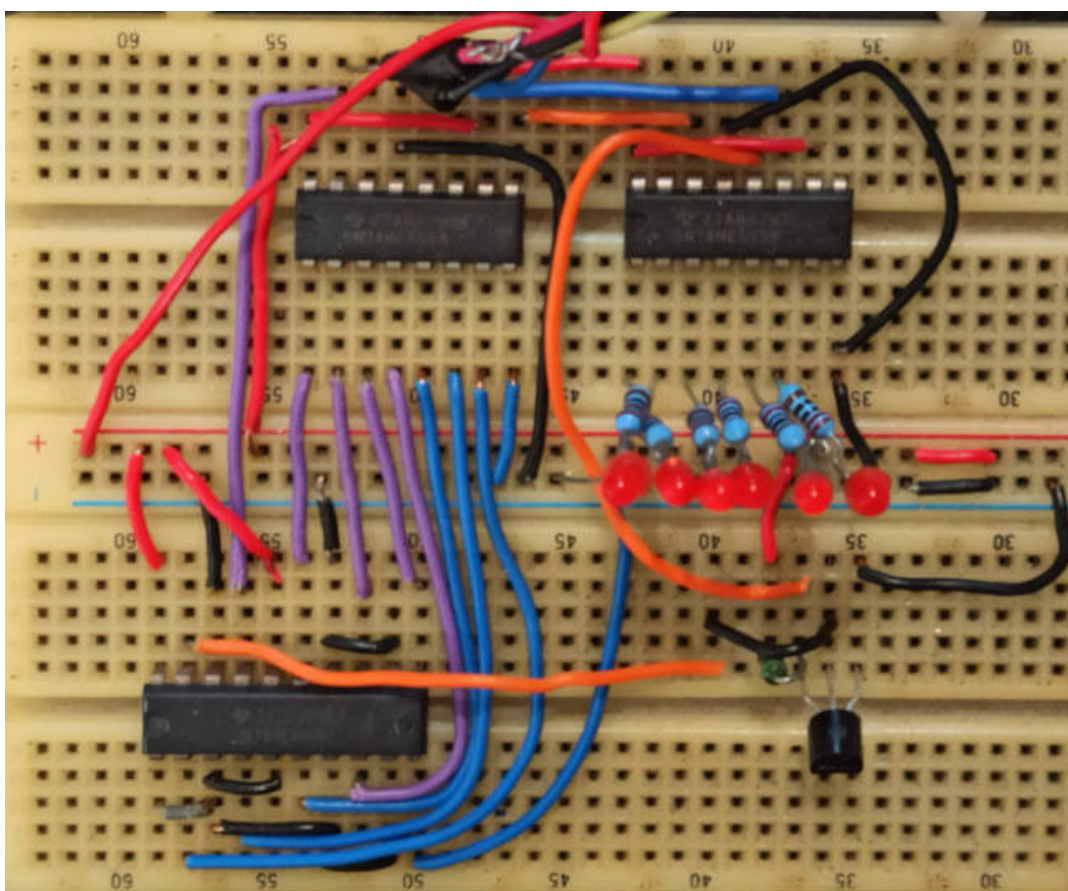


ΣΧΗΜΑ Α΄.4: Οδηγία «δεξιά»

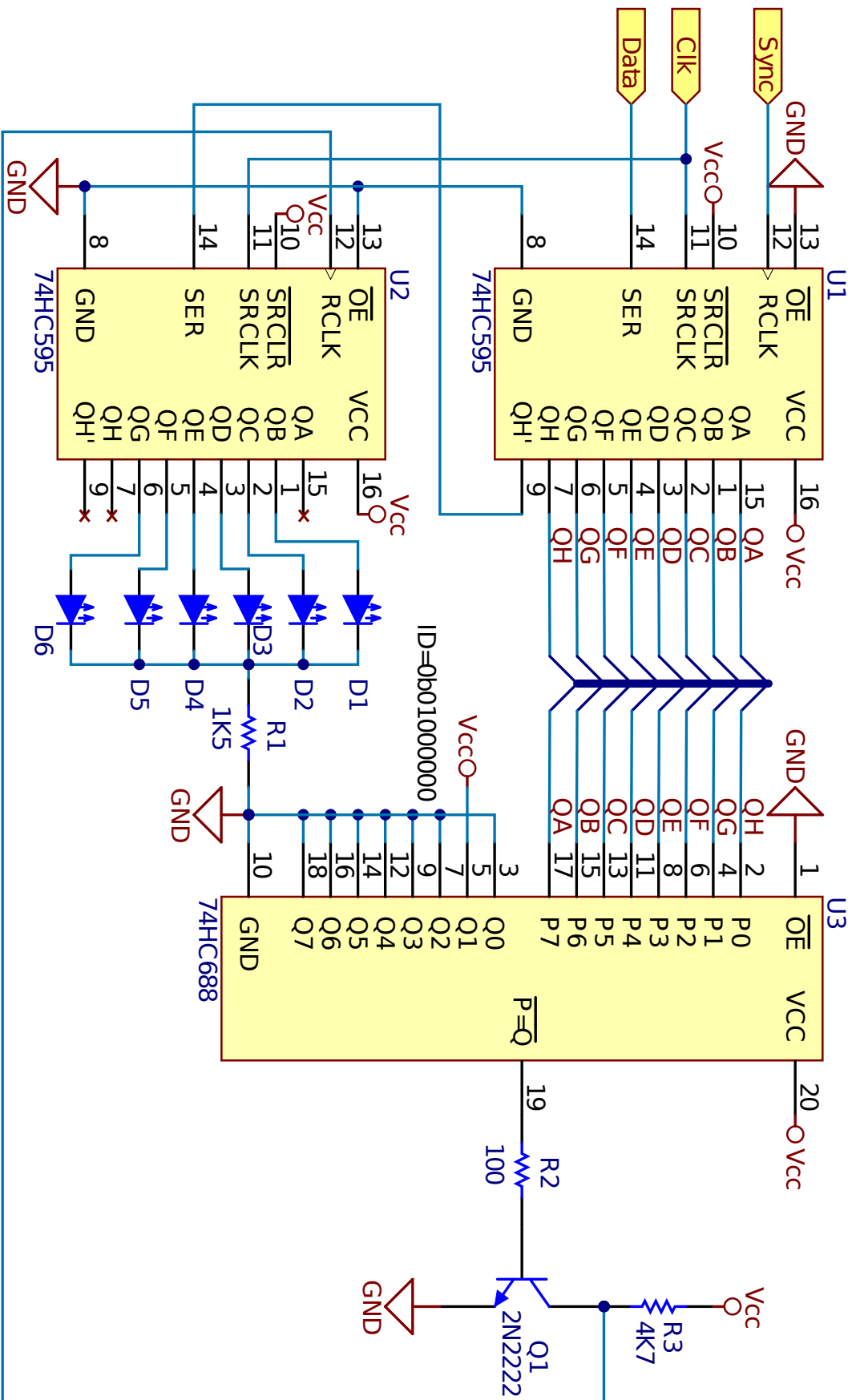
## Α'.2 Προσανατολισμένα



ΣΧΗΜΑ Α'.5: Κυκλωματικό διάγραμμα θεμελίου



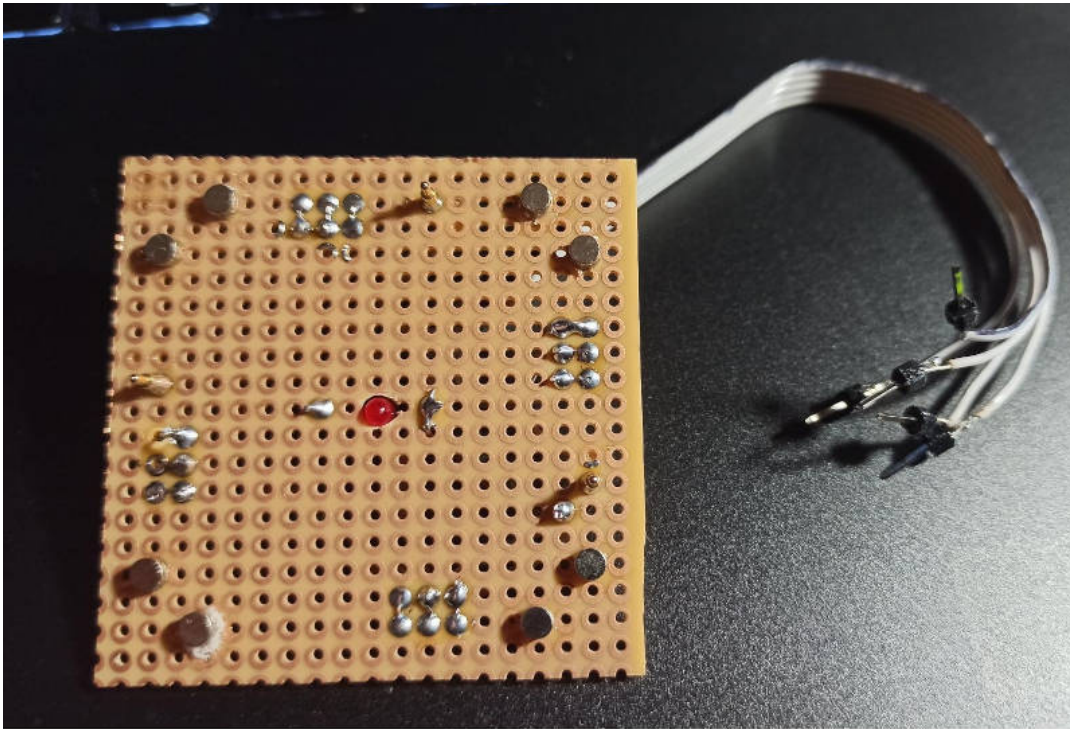
ΣΧΗΜΑ Α.6: Φωτογραφία δοκιμαστικού κυκλώματος στο raster/breadboard



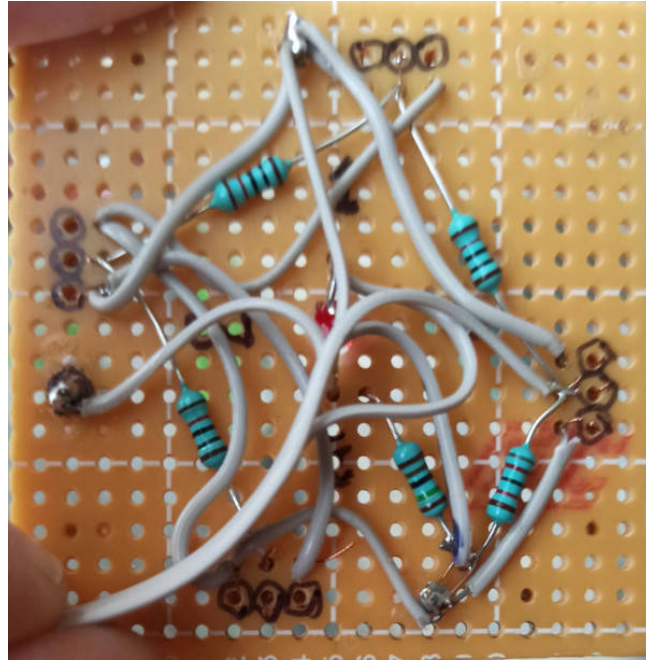
ΣΧΗΜΑ Α'7: Κυκλωματικό διάγραμμα κοβιδίου



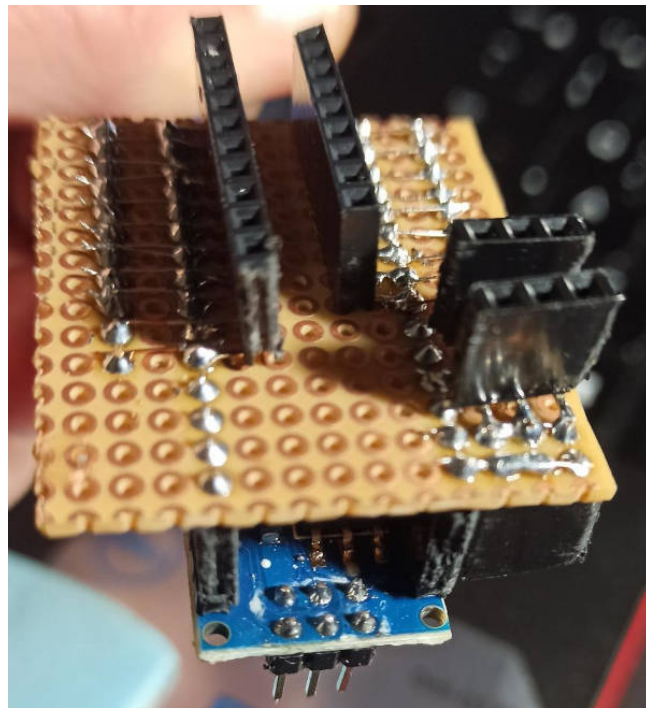
### Α.3 Συμμετρικά με διάτρητες πλακέτες



ΣΧΗΜΑ Α.8: Εξωτερική πλευρά έδρας

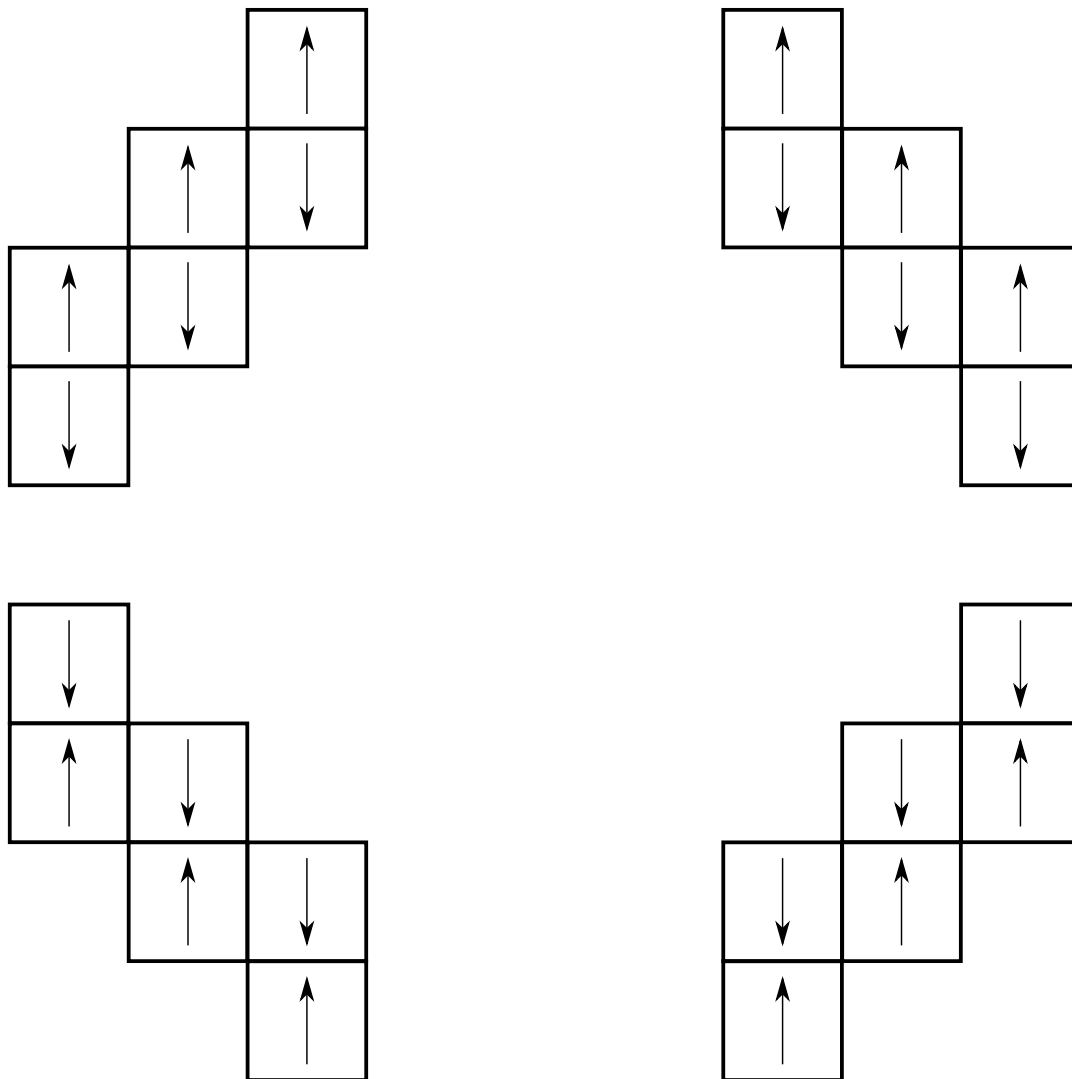


ΣΧΗΜΑ Α'.9: Ανάστροφη πλευρά έδρας



ΣΧΗΜΑ Α'.10: Εσωτερική συνδεσμολογία μικροελεγκτή

#### Α'.4 Διάφορα



ΣΧΗΜΑ Α.11: Τα 4 πιθανά αναπτύγματα



# Κατάλογος Σχημάτων

1.1 Ένα πιθανό ανάπτυγμα των μηχανικών συζευκτών των Προσανατολισμένων Κυβιδίων. . . . .	6
1.2 Κυκλωματικό διάγραμμα Προσανατολισμένων κυβιδίων . . . . .	8
2.1 Ένας συζευκτής σε μία έδρα . . . . .	12
2.2 Τετραπλασιασμός για συμμετρία . . . . .	12
2.3 Αντικατοπτρισμός των συμπληρωματικών συζευκτών . . . . .	12
2.4 Σύγκρουση συζευκτών . . . . .	12
2.5 Μία ακίδα με ελατήριο (αγγλιστί rogo pin) στην ίδια κλίμακα με τους μαγνήτες. . . . .	13
2.6 Διάταξη ηλεκτρικών ακροδεκτών. Με πράσινο τα pad και με κόκκινο το rogo pin. . . . .	15
2.7 Μία πλήρης έδρα κυβιδίου . . . . .	15
2.8 Ανάπτυγμα κυβιδίου . . . . .	16
2.9 Διάγραμμα ακροδεκτών Arduino Nano . . . . .	16
2.10 Συνδεσμολογία των ακροδεκτών δεδομένων . . . . .	19
2.11 Πλήρες κυκλωματικό διάγραμμα κυβιδίου . . . . .	20
2.12 Μία πιθανή διάταξη των βάσεων του θεμελίου . . . . .	21
2.13 Κυκλωματικό διάγραμμα θεμελίου . . . . .	22
3.1 Η μετάδοση του byte 0b00111001 με τα πρωτόκολλα RS-232 και Manchester. . . . .	29
3.2 Αποστολή και λήψη μεμονωμένων bit με βάση το πρωτόκολλο Manchester που εν τέλει χρησιμοποιήσαμε. . . . .	31
3.3 Το πρωτόκολλο χαμηλού επιπέδου (bit-προς-bit για τη μετάδοση byte) που χρησιμοποιήθηκε. . . . .	32
3.4 Το πρωτόκολλο υψηλού επιπέδου (byte-προς-byte για τη μετάδοση όλων των δεδομένων) που χρησιμοποιήθηκε. . . . .	34
3.5 Διαγράμματα ροής θεμελίου και κυβιδίων. . . . .	36
4.1 Αναπτύγματα κυβιδίου (εσωτερικά και εξωτερικά) . . . . .	39
4.2 Φωτογραφία δύο κυβιδίων και ενός θεμελίου . . . . .	40
4.3 Καθυστέρηση παρυφής και θόρυβος λόγω μικρών ρευμάτων . . . . .	43
5.1 Δικτυώματα διαίρεσης τάσης με αναγνώριση αποσύνδεσης . . . . .	48
5.2 Αντικατοπτρισμός του διαιρέτη του σχήματος 2.10 . . . . .	49
A.1 Καθοδηγητικά μηνύματα στην οθόνη LCD . . . . .	55
A.2 Διαδικασία κατασκευής . . . . .	55
A.3 Θεμέλιο . . . . .	55
A.4 Οδηγία «δεξιά» . . . . .	55
A.5 Κυκλωματικό διάγραμμα θεμελίου . . . . .	56
A.6 Φωτογραφία δοκιμαστικού κυκλώματος στο raster/breadboard . . . . .	57

---

A.7 Κυκλωματικό διάγραμμα κυβιδίου . . . . .	58
A.8 Εξωτερική πλευρά έδρας . . . . .	59
A.9 Ανάστροφη πλευρά έδρας . . . . .	60
A.10 Εσωτερική συνδεσμολογία μικροελεγκτή . . . . .	60
A.11 Τα 4 πιθανά αναπτύγματα . . . . .	61