

**Μάθημα:** Ρομποτική: Έλεγχος Ρομποτικού οχήματος με τον οδηγό L298N και Τηλεχειριστήριο

### Βιβλιογραφία:

Το φύλλο έργου στηρίχτηκε σε πληροφορίες

1. Από την ηλεκτρονική διεύθυνση <https://www.instructables.com/id/How-to-Use-L298n-to-Control-Dc-Motor-With-Arduino>
2. Από την ηλεκτρονική διεύθυνση <https://electronics hobbyists.com/controlling-dc-motors-arduino-arduino-l298n-tutorial/>
3. Από την ηλεκτρονική διεύθυνση <https://www.circuitbasics.com/arduino-ir-remote-receiver-tutorial/>

### Στόχοι:

1. Κατανόηση του προγραμματισμού με κλάσεις-βιβλιοθήκες όπως η IRremote και με αντικείμενα της κλάσης
2. Κατανόηση της λειτουργίας των DC κινητήρων και έλεγχος μέσω των κατάλληλων οδηγών (drivers)
3. Κατανόηση της λειτουργίας δέκτη υπερύθρων και τηλεχειριστηρίου και χρήση με το Arduino

### Τα Υλικά που θα χρειαστούμε:

- 1) Arduino board UNO R3
- 2) Πλακέτα L298 motor driver
- 3) 2 DC κινητήρες 3-12V
- 4) Δέκτη υπερύθρων
- 5) Πομπό υπερύθρων (Τηλεχειριστήριο)
- 6) Ράστερ (Breadboard)
- 7) Καλώδια
- 8) Τροφοδοσία 9 Volts
- 9) Ρομποτικό όχημα (σασί)



### Τι πρέπει να γνωρίζουμε:

Αντικείμενο του μαθήματος είναι ο έλεγχος της ταχύτητας και της φοράς περιστροφής 2 DC κινητήρων με τον οδηγό L298N και τη χρήση τηλεχειριστηρίου και δέκτη υπερύθρων.

Ένας κινητήρας DC μετατρέπει την ηλεκτρική ενέργεια σε κινητική χάρη στην αλληλεπίδραση ανάμεσα στο μαγνητικό πεδίο των μαγνητών του στάτη με τα πηνία του ρότορα, πράγμα το οποίο παράγει δύναμη και, κατά συνέπεια, ροπή, στον άξονα του κινητήρα.

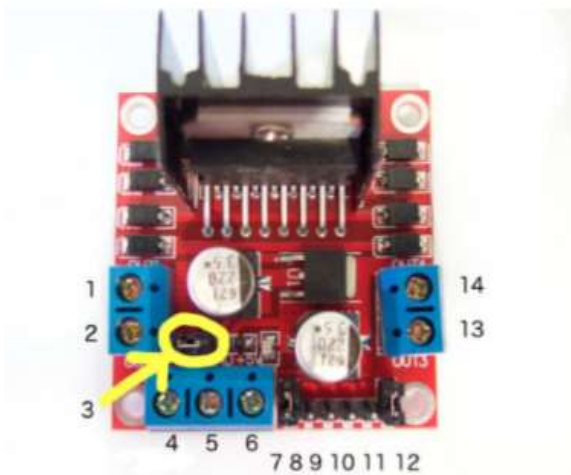
Επειδή οι κινητήρες αυτοί απαιτούν ρεύμα της τάξης άνω των 50mA, δεν μπορούμε να οδηγήσουμε απευθείας αυτούς τους κινητήρες από τους ακροδέκτες ενός μικροελεγκτή. Για το σκοπό αυτό χρησιμοποιούνται κατάλληλοι οδηγοί, που είναι στην ουσία ενισχυτές έντασης, δηλαδή παίρνουν στην είσοδο ένα σήμα ελέγχου χαμηλής έντασης και το μετατρέπουν σε ένα υψηλότερης έντασης σήμα ικανό να οδηγήσει τον κινητήρα. Εμείς εδώ θα χρησιμοποιήσουμε την πλακέτα οδηγού L298N που ονομάζεται και H-bridge με την οποία μπορούμε να ελέγξουμε 2 DC κινητήρες ή 1 βηματικό κινητήρα με ρεύμα αιχμής μέχρι 2A.

Ο οδηγός μπορεί να λειτουργήσει με τάσεις 5V έως 35V, ενώ διαθέτει και μία παροχή σταθεροποιημένης τάσης στα 5V (regulator) που μπορεί να χρησιμοποιηθεί για τροφοδοσία του Arduino. Η τάση των 5V μπορεί να χρησιμοποιηθεί εάν η παρεχόμενη τάση οδήγησης είναι έως και 12V.

Τα pin της πλακέτας L298N είναι: (βλ. εικ. 1)

1. DC Κινητήρας 1 παροχή ρεύματος +, ή βηματικός κινητήρας A +
2. DC Κινητήρας 1 παροχή ρεύματος -, ή βηματικός κινητήρας A –

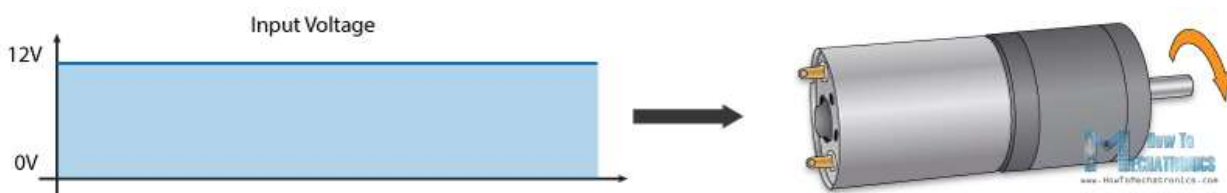
3. Jumper παροχής έως 12V. Βραχυκυκλώνουμε όταν η τάση τροφοδοσίας των μοτέρ (σύνδεση 4) είναι έως 12V. Σε αυτή την περίπτωση παρέχει σταθεροποιημένη τάση 5V (σύνδεση 6) μέσω ρυθμιστή, για τροφοδοσία του Arduino.
4. Σύνδεση της τάσης τροφοδοσίας των κινητήρων με μέγιστο τα 35V.
5. Γείωση
6. Παροχή σταθεροποιημένης τάσης 5V.
7. Σύνδεση της PWM εξόδου για έλεγχο της ταχύτητας του DC κινητήρα 1. Διαθέτει και βραχυκυκλωτήρα (jumper) που πρέπει να τοποθετηθεί όταν χρησιμοποιείται βηματικός κινητήρας και να αφαιρείται όταν χρησιμοποιείται DC κινητήρας.
8. Είσοδος 1 (καθορίζει την φορά κίνησης του κινητήρα 1)
9. Είσοδος 2 (καθορίζει την φορά κίνησης του κινητήρα 1)
10. Είσοδος 3 (καθορίζει την φορά κίνησης του κινητήρα 2)
11. Είσοδος 4 (καθορίζει την φορά κίνησης του κινητήρα 2)
12. Σύνδεση της PWM εξόδου για έλεγχο της ταχύτητας του DC κινητήρα 2. Διαθέτει και βραχυκυκλωτήρα (jumper) που πρέπει να τοποθετηθεί όταν χρησιμοποιείται βηματικός κινητήρας και να αφαιρείται όταν χρησιμοποιείται DC κινητήρας.
13. DC Κινητήρας 2 παροχή ρεύματος +, ή βηματικός κινητήρας B +
14. DC Κινητήρας 2 παροχή ρεύματος -, ή βηματικός κινητήρας B -



**Εικ. 1** Πλακέτα οδηγού L298N

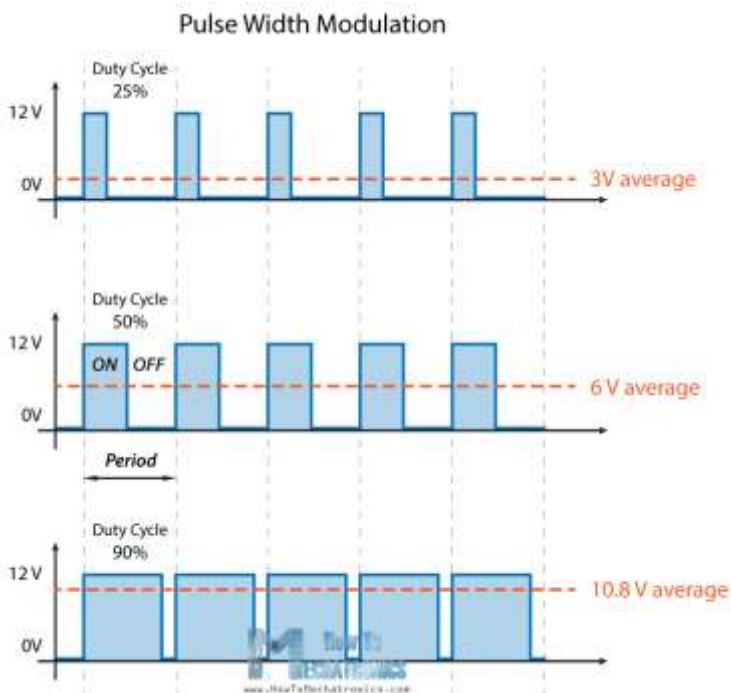
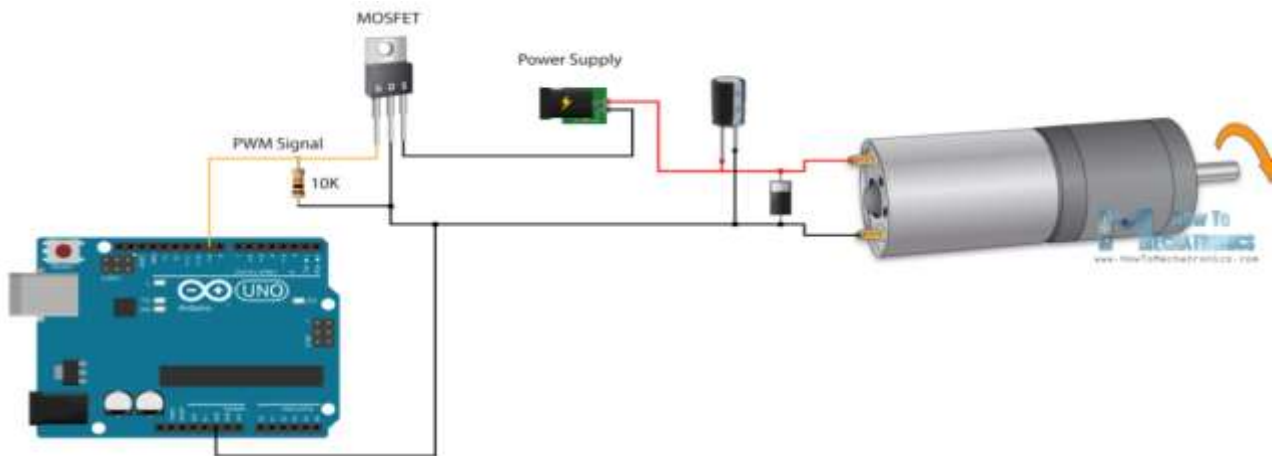
### Λειτουργία και έλεγχος του κινητήρα

Μπορούμε να ελέγξουμε την ταχύτητα του κινητήρα συνεχούς ρεύματος απλά ελέγχοντας την τάση εισόδου στο μοτέρ χρησιμοποιώντας σήμα **PWM**. PWM ή διαμόρφωση εύρους παλμών είναι μια τεχνική που μας επιτρέπει να ρυθμίζουμε τη μέση τιμή της τάσης που παρέχεται στον κινητήρα ενεργοποιώντας και αποσυνδέοντας την ισχύ με γρήγορο ρυθμό. Η μέση τάση εξαρτάται από τον κύκλο λειτουργίας ή από το χρονικό διάστημα που το σήμα είναι ενεργοποιημένο σε σχέση με το χρόνο που το σήμα είναι απενεργοποιημένο, σε μια μόνο χρονική περίοδο. (βλ. Εικ. 2)



Έτσι, ανάλογα με το μέγεθος του κινητήρα, μπορούμε απλά να συνδέσουμε μια έξοδο Arduino PWM στη βάση του τρανζίστορ ή την πύλη ενός MOSFET και να ελέγξουμε την ταχύτητα του κινητήρα ελέγχοντας την έξοδο PWM. Το σήμα χαμηλής ισχύος Arduino PWM ενεργοποιεί και απενεργοποιεί την πύλη στο

MOSFET μέσω του οποίου οδηγείται ο κινητήρας υψηλής ισχύος. Επομένως, αν συνδυάσουμε αυτές τις δύο μεθόδους, το PWM και τη γέφυρα H, μπορούμε να έχουμε πλήρη έλεγχο του κινητήρα συνεχούς ρεύματος.



**Εικ. 2** Τεχνική PWM

### Έλεγχος φοράς περιστροφής με τον οδηγό L298N

Η φορά περιστροφής του κινητήρα 1 ελέγχεται από τις τιμές στις εισόδους **8** και **9** της πλακέτας L298N. Εάν δώσουμε HIGH στη μία και LOW στην άλλη ο κινητήρας 1 περιστρέφεται προς την μία κατεύθυνση, ενώ αν δώσουμε LOW στην πρώτη και HIGH στην δεύτερη τότε περιστρέφεται προς την άλλη κατεύθυνση. Βέβαια ο κινητήρας δεν ξεκινάει αν δεν πάρει θετικό παλμό (HIGH) στην είσοδο **7**, όπου και θα πρέπει να δώσουμε ένα PWM σήμα για να ελέγξουμε την ταχύτητα του κινητήρα. Εάν το PWM σήμα ρυθμιστεί με διάρκεια παλμού 1% ο κινητήρας έχει την ελάχιστη ταχύτητα περιστροφής, ενώ αν δοθεί σήμα PWM με διάρκεια παλμού 100% τότε ο κινητήρας θα έχει την μέγιστη ταχύτητα περιστροφής. Για να σταματήσουμε τον κινητήρα αρκεί να δώσουμε χαμηλή στάθμη (LOW) και στην σύνδεση 8 αλλά και στην σύνδεση 9. Αντίστοιχα, η φορά περιστροφής του κινητήρα 2 ελέγχεται από τις τιμές στις εισόδους **10** και **11** της πλακέτας L298N. Εάν δώσουμε HIGH στη μία και LOW στην άλλη ο κινητήρας 2 περιστρέφεται προς

την μία κατεύθυνση, ενώ αν δώσουμε LOW στην πρώτη και HIGH στην δεύτερη τότε περιστρέφεται προς την άλλη κατεύθυνση. Βέβαια ο κινητήρας 2 δεν ξεκινάει αν δεν πάρει θετικό παλμό (HIGH) στην είσοδο **12**, όπου και θα πρέπει να δώσουμε ένα PWM σήμα για να ελέγξουμε την ταχύτητα του κινητήρα 2. Εάν το PWM σήμα ρυθμιστεί με διάρκεια παλμού 1% ο κινητήρας 2 θα έχει την ελάχιστη ταχύτητα περιστροφής, ενώ αν δοθεί σήμα PWM με διάρκεια παλμού 100% τότε ο κινητήρας 2 θα έχει την μέγιστη ταχύτητα περιστροφής. Για να σταματήσουμε τον κινητήρα 2 αρκεί να δώσουμε χαμηλή στάθμη (LOW) και στην σύνδεση 10 αλλά και στην σύνδεση 11.

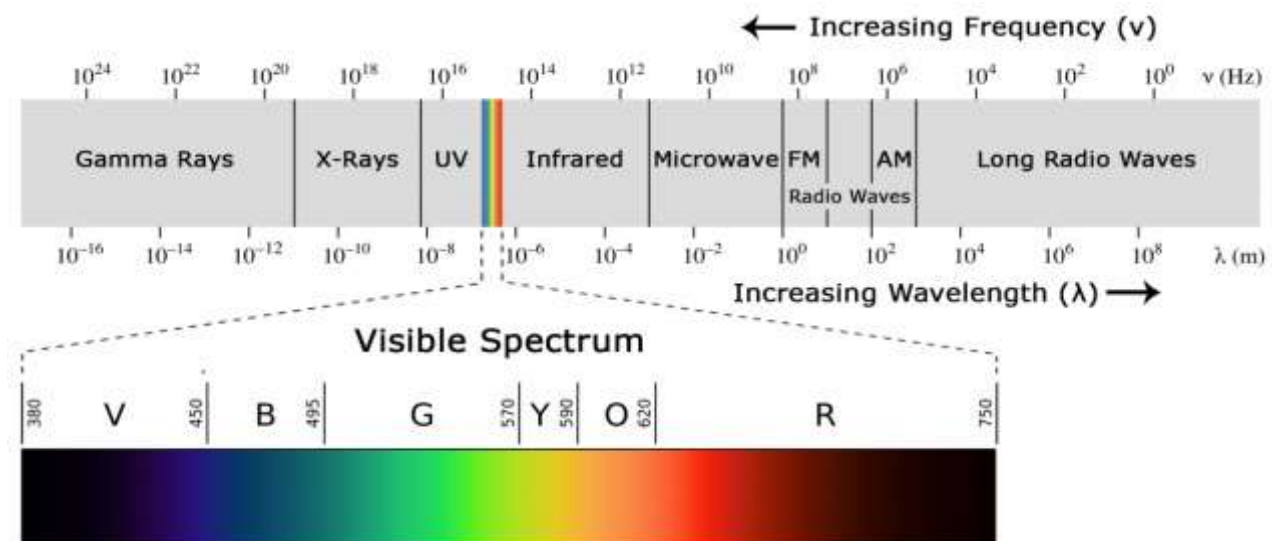
Όλα τα παραπάνω συνοψίζονται στον παρακάτω πίνακα:

Κινητήρας 1			
Περιστροφή προς την μία κατεύθυνση	Σύνδεση 8 (Είσοδος 1) = HIGH Σύνδεση 9 (Είσοδος 2) = LOW	Έλεγχος ταχύτητας:	Αργά: στη Σύνδεση 7 PWM σήμα <50% Γρήγορα: στη Σύνδεση 7 PWM σήμα >50%
Περιστροφή προς την άλλη κατεύθυνση	Σύνδεση 8 (Είσοδος 1) = LOW Σύνδεση 9 (Είσοδος 2) = HIGH	Έλεγχος ταχύτητας:	Αργά: στη Σύνδεση 7 PWM σήμα <50% Γρήγορα: στη Σύνδεση 7 PWM σήμα >50%
Σταμάτημα Κινητήρα	Σύνδεση 8 (Είσοδος 1) = LOW Σύνδεση 9 (Είσοδος 2) = LOW		
Κινητήρας 2			
Περιστροφή προς την μία κατεύθυνση	Σύνδεση 10 (Είσοδος 3) = HIGH Σύνδεση 11 (Είσοδος 4) = LOW	Έλεγχος ταχύτητας:	Αργά: στη Σύνδεση 12 PWM σήμα <50% Γρήγορα: στη Σύνδεση 12 PWM σήμα >50%
Περιστροφή προς την άλλη κατεύθυνση	Σύνδεση 10 (Είσοδος 3) = LOW Σύνδεση 11 (Είσοδος 4) = HIGH	Έλεγχος ταχύτητας:	Αργά: στη Σύνδεση 12 PWM σήμα <50% Γρήγορα: στη Σύνδεση 12 PWM σήμα >50%
Σταμάτημα Κινητήρα	Σύνδεση 10 (Είσοδος 1) = LOW Σύνδεση 11 (Είσοδος 2) = LOW		

### Δέκτης και πομπός υπέρυθρων (Τηλεχειριστήριο)

Η επικοινωνία μέσω υπέρυθρων (IR) είναι μια ευρέως χρησιμοποιούμενη και εύκολη στην εφαρμογή ασύρματη τεχνολογία που έχει πολλές χρήσιμες εφαρμογές. Τα πιο σημαντικά παραδείγματα στην καθημερινή ζωή είναι τα τηλεχειριστήρια τηλεόρασης / βίντεο, οι αισθητήρες κίνησης και τα υπέρυθρα θερμομέτρα.

Η υπέρυθρη ακτινοβολία είναι μια μορφή φωτός παρόμοια με το φως που βλέπουμε γύρω μας. Η μόνη διαφορά μεταξύ του φωτός IR και του ορατού φωτός είναι η συχνότητα και το μήκος κύματος. Η υπέρυθρη ακτινοβολία βρίσκεται έξω από το φάσμα του ορατού φωτός, έτσι ώστε οι άνθρωποι δεν μπορούν να τη δουν.



Εικ. 3 Διάγραμμα ηλεκτρομαγνητικού φάσματος

Επειδή η υπέρυθρη ακτινοβολία είναι ένας τύπος φωτός, η επικοινωνία αυτή απαιτεί άμεση οπτική επαφή από τον δέκτη στον πομπό, δηλαδή δε μπορεί να μεταδοθεί μέσω τοίχων ή άλλων υλικών.

### Πως λειτουργεί ένα σύστημα επικοινωνίας με υπέρυθρες ακτίνες

Ένα τυπικό σύστημα επικοινωνίας υπέρυθρων αποτελείται από ένα πομπό IR και ένα δέκτη IR. Ο πομπός μοιάζει με ένα τυπικό LED, εκτός από το ότι παράγει φως στο φάσμα IR αντί του ορατού φάσματος. Αν κοιτάξετε στο μπροστινό μέρος του τηλεχειριστηρίου, θα δείτε το LED του πομπού IR (βλ. εικ. 4).



**Εικ. 4** Πομπός IR (Τηλεχειριστήριο)



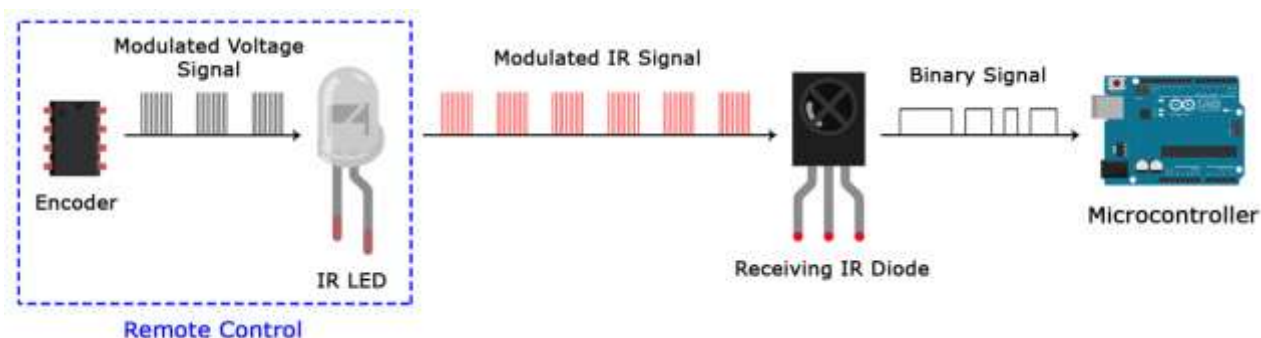
**Εικ. 5** Δέκτης IR

Ο δέκτης IR είναι μια φωτοδίοδος και ένας προενισχυτής που μετατρέπει το φως IR σε ένα ηλεκτρικό σήμα (βλ. εικ. 5).

### Διαμόρφωση σήματος IR

Το φως IR εκπέμπεται από τον ήλιο, τους λαμπτήρες και στιδήποτε άλλο παράγει θερμότητα. Αυτό σημαίνει ότι υπάρχει πολύς θόρυβος υπέρυθρης ακτινοβολίας γύρω από εμάς. Για να αποφευχθεί η παρεμβολή αυτού του θορύβου στο σήμα IR, χρησιμοποιείται μια τεχνική διαμόρφωσης σήματος.

Στη διαμόρφωση του σήματος IR, ένας κωδικοποιητής στο τηλεχειριστήριο IR μετατρέπει ένα δυαδικό σήμα σε ένα διαμορφωμένο ηλεκτρικό σήμα. Αυτό το ηλεκτρικό σήμα αποστέλλεται στο LED μετάδοσης. Το LED εκπομπής μετατρέπει το διαμορφωμένο ηλεκτρικό σήμα σε διαμορφωμένο σήμα φωτός IR. Ο δέκτης IR αποδιαμορφώνει αυτό το σήμα φωτός IR και το μετατρέπει σε δυαδική μορφή πριν μεταβιβάσει τις πληροφορίες σε ένα μικροελεγκτή (βλ. εικ. 6).



**Εικ. 6** Διαμόρφωση σήματος IR

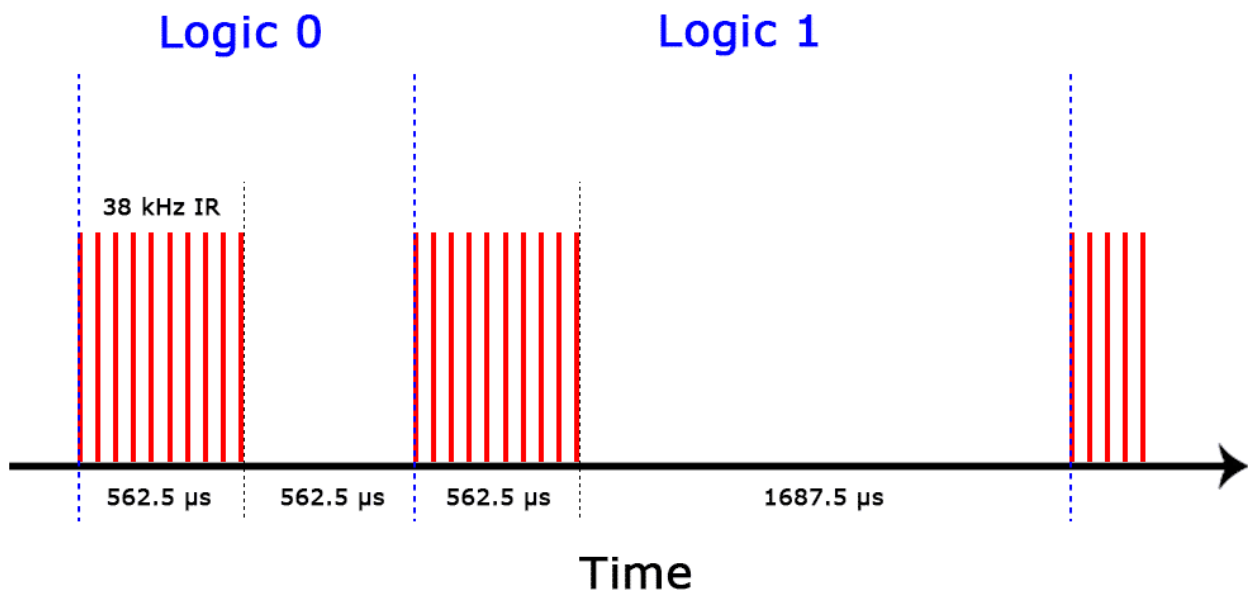
Το διαμορφωμένο σήμα IR είναι μία σειρά IR παλμών φωτός που ενεργοποιούνται και απενεργοποιούνται σε μια υψηλή συχνότητα γνωστή ως συχνότητα φορέα. Η φέρουσα συχνότητα που χρησιμοποιείται από τους περισσότερους πομπούς είναι 38 kHz, επειδή είναι σπάνια στη φύση και έτσι μπορεί να διακριθεί από τον θόρυβο του περιβάλλοντος. Με αυτό τον τρόπο ο δέκτης IR γνωρίζει ότι το σήμα 38 kHz έχει αποσταλεί από τον πομπό και όχι από άλλη πηγή του περιβάλλοντος.

Η φωτοδίοδος του δέκτη ανιχνεύει όλες τις συχνότητες του φωτός IR, αλλά έχει ένα φίλτρο ζώνης διέλευσης που επιτρέπει μόνο στο σήμα IR των 38 kHz να περάσει. Στη συνέχεια, ενισχύει το διαμορφωμένο σήμα με έναν προενισχυτή και το μετατρέπει σε δυαδικό σήμα πριν το αποστείλει σε ένα μικροελεγκτή.

### Πρωτόκολλα μεταφοράς σήματος IR

Το πρότυπο με το οποίο μετατρέπεται το διαμορφωμένο σήμα IR σε δυαδικό ορίζεται από ένα πρωτόκολλο μετάδοσης. Υπάρχουν πολλά πρωτόκολλα μετάδοσης σήματος IR όπως αυτά της Sony, Matsushita, NEC και RC5. Εμείς εδώ θα χρησιμοποιήσουμε το πρωτόκολλο της NEC που είναι ένα από τα πιο συνηθισμένα πρωτόκολλα.

Το λογικό '1' ξεκινά με έναν παλμό HIGH διάρκειας 562,5 μs συχνότητας 38 kHz που ακολουθείται από παλμό LOW διάρκειας 1,687,5 μs. Το λογικό '0' μεταδίδεται με παλμό HIGH διάρκειας 562,5 μs που ακολουθείται από παλμό LOW διάρκειας 562,5 μs (βλ. εικ. 7).



**Εικ. 7** Διαμορφωμένο σήμα IR

Αυτός είναι ο τρόπος με τον οποίο το πρωτόκολλο NEC κωδικοποιεί και αποκωδικοποιεί τα δυαδικά δεδομένα σε διαμορφωμένο σήμα. Άλλα πρωτόκολλα διαφέρουν μόνο στη διάρκεια των μεμονωμένων παλμών HIGH και LOW.

### Κωδικοί IR

Κάθε φορά που πατάμε ένα πλήκτρο στο τηλεχειριστήριο, δημιουργείται ένας μοναδικός δεκαεξαδικός κωδικός. Αυτός ο κωδικός αποτελεί τις πληροφορίες που διαμορφώνονται και αποστέλλονται μέσω IR στο δέκτη. Για να αποκρυπτογραφήσει ο μικροελεγκτής ποιο πλήκτρο έχει πατηθεί, πρέπει να γνωρίζει ποιος κωδικός αντιστοιχεί σε κάθε πλήκτρο στο τηλεχειριστήριο.

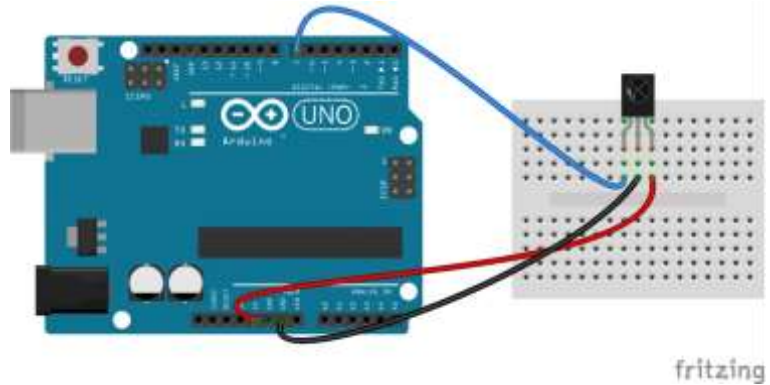
Διαφορετικά τηλεχειριστήρια στέλνουν διαφορετικούς κωδικούς για κάθε πλήκτρο, επομένως θα πρέπει να καθορίσουμε τον κώδικα που δημιουργήθηκε για κάθε πλήκτρο στο συγκεκριμένο τηλεχειριστήριο. Εάν μπορούμε να ανατρέξουμε στο τεχνικό εγχειρίδιο του τηλεχειριστηρίου που χρησιμοποιούμε, μπορούμε να βρούμε τους κωδικούς των πλήκτρων. Εάν όχι, θα χρησιμοποιήσουμε έναν κώδικα στη wiring C στο περιβάλλον IDE του Arduino που θα διαβάσει τα περισσότερα από τα δημοφιλή τηλεχειριστήρια και θα τυπώσει τους δεκαεξαδικούς κωδικούς στη σειριακή οθόνη μόλις πατήσουμε ένα πλήκτρο.

### Σύνδεση δέκτη IR στο Arduino

Όλοι οι δέκτες IR έχουν τρεις ακροδέκτες: σήμα, γείωση και τροφοδοσία (Vcc) (βλ. εικ. 8) και συνδέονται με το μικροελεγκτή με πολύ απλό τρόπο (βλ. εικ. 9).



**Εικ. 8**



**Εικ. 9** Σύνδεση δέκτη IR με το Arduino

Αφού συνδέσουμε τον δέκτη IR, πρέπει να εγκαταστήσουμε τη βιβλιοθήκη IRremote στο προγραμματιστικό περιβάλλον του Arduino και να ξεκινήσουμε τον προγραμματισμό για να βρούμε τους κωδικούς που αποστέλλονται από το τηλεχειριστήριό μας.

Μπορούμε να κατεβάσουμε τη βιβλιοθήκη από το σύνδεσμο <http://z3t0.github.io/Arduino-IRremote/> σε μορφή συμπιεσμένου αρχείου zip. Στη συνέχεια για να εγκαταστήσουμε τη βιβλιοθήκη από το αρχείο ZIP, ανοίγουμε το IDE του Arduino και από το μενού Sketch> Include Library> Add .ZIP Library επιλέγουμε το αρχείο IRremote.zip που κατεβάσαμε από τον παραπάνω σύνδεσμο.

Για να βρούμε τους κωδικούς από το τηλεχειριστήριό μας, μεταφορτώνουμε τον παρακάτω κώδικα στο Arduino και ανοίγουμε τη σειριακή οθόνη.

```
#include <IRremote.h> //ενσωμάτωση της βιβλιοθήκης IRremote.h στο IDE του Arduino

int RECV_PIN = 11; //ορισμός σύνδεσης του ακροδέκτη S (εικ. 8) του δέκτη IR στο αντίστοιχο pin του Arduino

IRrecv irrecv(RECV_PIN); // δημιουργία αντικειμένου κλάσης IRrecv με το όνομα irrecv και καθορισμός του ψηφιακού ακροδέκτη όπου συνδέεται ο δέκτης IR με το Arduino. Αυτό το αντικείμενο θα φροντίσει για το πρωτόκολλο και την επεξεργασία των πληροφοριών από τον δέκτη

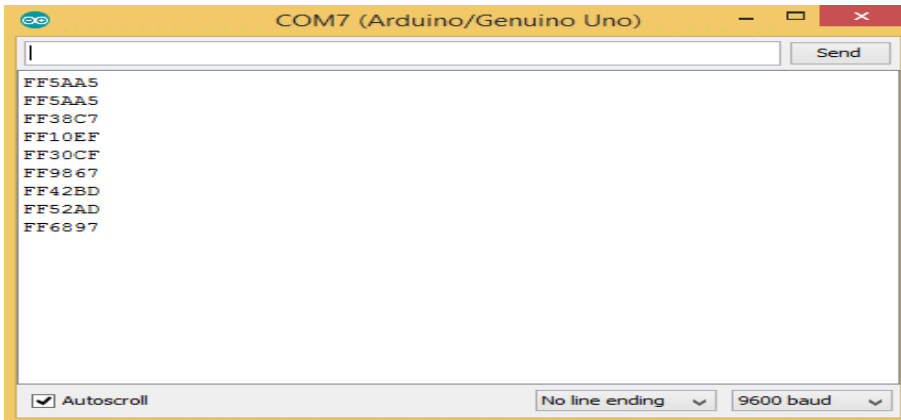
decode results results; // δημιουργία αντικειμένου κλάσης decode results με το όνομα results το οποίο θα χρησιμοποιηθεί από το αντικείμενο irrecv για να αποθηκεύσει (αποκωδικοποιήσει) τους κωδικούς που στέλνει ο πομπός IR στο δέκτη IR

void setup() {
  Serial.begin(9600); // ενεργοποίηση της σειριακής επικοινωνίας του Arduino με την οθόνη με ταχύτητα 9600bps

  irrecv.enableIRin(); // ενεργοποίηση του δέκτη IR ώστε να λαμβάνει σήματα IR
  irrecv.blink13(true); // κάθε φορά που ο δέκτης IR θα λαμβάνει σήμα IR θα αναβοσβήνει το ενσωματωμένο Led πάνω στη πλακέτα;
}

void loop() {
  if (irrecv.decode(&results)) { // έλεγχος αν έχει ληφθεί κάποιος κωδικός από το δέκτη IR
    Serial.println(results.value, HEX); // εμφάνιση της τιμής του πλήκτρου που έχει πατηθεί σε δεκαεξαδική μορφή
    irrecv.resume(); // προετοιμασία του δέκτη IR για λήψη του επόμενου κωδικού
  }
}
```

Τώρα πατήστε κάθε πλήκτρο στο τηλεχειριστήριο και καταγράψτε τον δεκαεξαδικό κώδικα που αντιστοιχεί σε κάθε πάτημα πλήκτρου (βλ. εικ. 10)



**Εικ. 10** Εμφάνιση δεκαεξαδικού κώδικα στη σειριακή οθόνη του Arduino

Από τη σειριακή οθόνη του Arduino σημειώνουμε τους δεκαεξαδικούς κωδικούς των πλήκτρων που θα χρησιμοποιήσουμε στην εφαρμογή μας (βλ. εικ. 11)



**Εικ. 11** Κωδικοί πλήκτρων του τηλεχειριστηρίου της εφαρμογής μας

### Κύκλωμα εφαρμογής

Η εφαρμογή που έχουμε αναπτύξει θα χρησιμοποιηθεί για τον έλεγχο του ρομποτικού οχήματος που έχουμε συναρμολογήσει στο εργαστήριο (βλ. εικόνες 13 και 14).

Για τη κίνηση του οχήματος θα χρησιμοποιήσουμε 2 κινητήρες συνεχούς ρεύματος μόνιμου μαγνήτη με γρανάζωμα υποπολλαπλασιασμού στροφών (βλ. εικ. 12) με τα εξής χαρακτηριστικά:

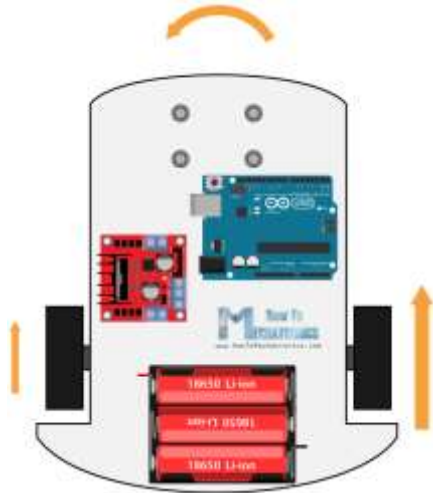


Τάση λειτουργίας: 3V - 12VDC  
(προτεινόμενη 6 - 8V)  
Μέγιστη ροπή: 0.8 KgrXcm  
Σχέση γραναζιών: 1:48  
Ταχύτητα περιστροφής: μέγιστη 170 rpm  
Ρεύμα: τυπικό 70mA (μέγιστο 250mA)  
Μέγεθος: 7x2.2x1.8cm

**Εικ. 12** DC κινητήρας



Οι κινητήρες θα οδηγούν τροχούς διαμέτρου 66mm και πάχους 17mm για καλύτερο οπτικό έλεγχο της περιστροφής τους. Για την οδήγηση και τον έλεγχο των κινητήρων θα χρησιμοποιηθεί ο οδηγός L298N που θα τοποθετηθεί επάνω στο σασί του οχήματος μαζί με τον μικροελεγκτή Arduino (βλ. εικ.13). Η τροφοδοσία των κινητήρων θα γίνει από μια συστοιχία 4 αλκαλικών μπαταριών τύπου AAA, συνολικής τάσης 6 V. Η κίνηση του οχήματος θα ελέγχεται από το τηλεχειριστήριο υπερύθρων με τα πλήκτρα που έχουμε σημειώσει παραπάνω (βλ. εικ. 11).



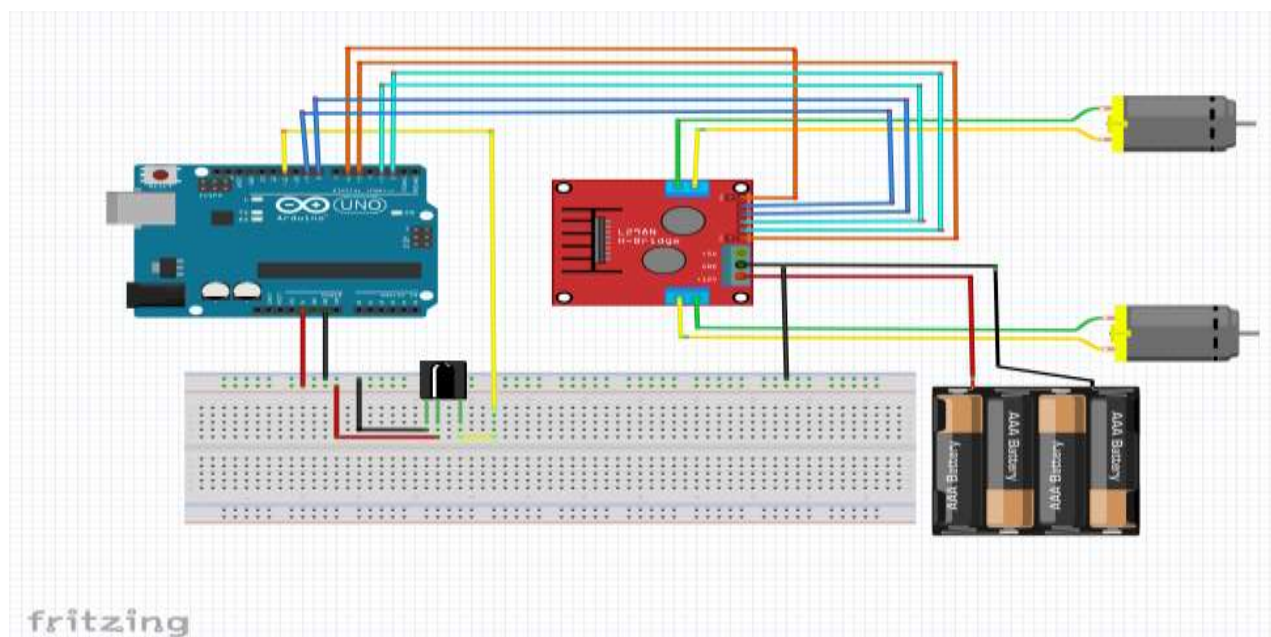
**Εικ. 13** Σχηματική απεικόνιση οχήματος



**Εικ. 14** Ρομπωτικό όχημα

Πιο αναλυτικά ο χειρισμός του οχήματος θα γίνεται σύμφωνα με τον παρακάτω πίνακα:

Πλήκτρο	Ενέργεια
▷▷  (NEXT)	Κίνηση μπροστά
◁◁ (PREV)	Κίνηση πίσω
▷   (PLAY/PAUSE)	Σταμάτημα (STOP)
CH-	Στροφή αριστερά
CH+	Στροφή δεξιά
- (VOL-)	Μείωση ταχύτητας
+ (VOL+)	Αύξηση ταχύτητας



Παρακάτω δίνεται ο κώδικας του προγράμματος στη Wiring C που θα φορτωθεί στο μικροελεγκτή Arduino και θα ελέγχει το ρομποτικό όχημα.

### Κώδικας προγράμματος στη Wiring C

```
#include <IRremote.h> // ενσωμάτωση της βιβλιοθήκης IRremote.h στο IDE του Arduino
IRrecv rc(11); // δημιουργία αντικειμένου κλάσης IRrecv με το όνομα rc και
                // σύνδεση του ακροδέκτη S του δέκτη IR στο pin 11 του Arduino

decode_results results; // δημιουργία αντικειμένου κλάσης decode_results με το όνομα
                        // results

const int ENA = 5; // σύνδεση των ακροδεκτών ENA, IN1 και IN2 του L298N στους
const int IN1 = 2; // ψηφιακούς ακροδέκτες 5, 2, 3 του Arduino αντίστοιχα
const int IN2 = 3;
const int ENB = 6; // σύνδεση των ακροδεκτών ENB, IN3 και IN4 του L298N στους
const int IN3 = 8; // ψηφιακούς ακροδέκτες 6, 8, 9 του Arduino αντίστοιχα
const int IN4 = 9;

int speed = 100; // δήλωση μεταβλητής ακέραιου τύπου με αρχική τιμή 100 για
                 // την ταχύτητα περιστροφής των 2 κινητήρων

void setup(){
  Serial.begin(9600); // ενεργοποίηση της σειριακής επικοινωνίας του Arduino με την
                     // οθόνη με ταχύτητα 9600bps

  rc.enableIRIn(); // ενεργοποίηση του δέκτη IR ώστε να λαμβάνει σήματα IR

  pinMode(ENA,OUTPUT); // δήλωση των ψηφιακών ακροδεκτών του Arduino που έχουν
  pinMode(ENB,OUTPUT); // συνδεθεί με τους αντίστοιχους ακροδέκτες του L298N, ως
  pinMode(IN1,OUTPUT); // έξοδοι
  pinMode(IN2,OUTPUT);
  pinMode(IN3,OUTPUT);
  pinMode(IN4,OUTPUT);
}

void loop(){
  if (rc.decode(&results)){ // έλεγχος αν έχει ληφθεί κάποιος κωδικός από το δέκτη IR
    switch(results.value){ // έλεγχος της τιμής του κωδικού που έχει ληφθεί
      case 0xFFA25D: // αν έχει ληφθεί ο κωδικός που αντιστοιχεί στο πλήκτρο CH-
        turnLeft(); // ενεργοποίηση της εντολής «Στροφή αριστερά»
        break; // έξοδος από τη δομή ελέγχου switch
      case 0xFFE21D: // αν έχει ληφθεί ο κωδικός που αντιστοιχεί στο πλήκτρο CH+
        turnRight(); // ενεργοποίηση της εντολής «Στροφή δεξιά»
        break; // έξοδος από τη δομή ελέγχου switch
      case 0xFF22DD: // αν έχει ληφθεί ο κωδικός που αντιστοιχεί στο πλήκτρο PREV
        backward(); // ενεργοποίηση της εντολής «Κίνηση πίσω»
        break ; // έξοδος από τη δομή ελέγχου switch
      case 0xFF02FD: // αν έχει ληφθεί ο κωδικός που αντιστοιχεί στο πλήκτρο NEXT
        forward(); // ενεργοποίηση της εντολής «Κίνηση μπροστά»
        break; // έξοδος από τη δομή ελέγχου switch
      case 0xFFC23D: // αν έχει ληφθεί ο κωδικός που αντιστοιχεί στο πλήκτρο PAUSE
        stop(); // ενεργοποίηση της εντολής «Σταμάτημα»
        break; // έξοδος από τη δομή ελέγχου switch
      case 0xFFE01F: // αν έχει ληφθεί ο κωδικός που αντιστοιχεί στο πλήκτρο VOL-
        speedDown(); // ενεργοποίηση της εντολής «Μείωση ταχύτητας»
        break ; // έξοδος από τη δομή ελέγχου switch
      case 0xFFA857: // αν έχει ληφθεί ο κωδικός που αντιστοιχεί στο πλήκτρο VOL+
```

```

        speedUp();
        break ;
    }
    rc.resume();
}
}

void forward(){
    Serial.println("forward");
    analogWrite(ENA, speed);
    analogWrite(ENB, speed);
    digitalWrite(IN1, HIGH);
    digitalWrite(IN2, LOW);
    digitalWrite(IN3, HIGH);
    digitalWrite(IN4, LOW);
}

void backward(){
    Serial.println("backward");
    analogWrite(ENA, speed);
    analogWrite(ENB, speed);
    digitalWrite(IN1, LOW);
    digitalWrite(IN2, HIGH);
    digitalWrite(IN3, LOW);
    digitalWrite(IN4, HIGH);
}

void turnLeft(){
    Serial.println("turnLeft");
    analogWrite(ENA, 0);
    analogWrite(ENB, speed);
    digitalWrite(IN1, HIGH);
    digitalWrite(IN2, LOW);
    digitalWrite(IN3, HIGH);
    digitalWrite(IN4, LOW);
}

void turnRight(){
    Serial.println("turnRight");
    analogWrite(ENA, speed);
    analogWrite(ENB, 0);
    digitalWrite(IN1, HIGH);
    digitalWrite(IN2, LOW);
    digitalWrite(IN3, HIGH);
    digitalWrite(IN4, LOW);
}

void stop(){
    Serial.println("stop");
    digitalWrite(IN1, LOW);
    digitalWrite(IN2, LOW);
    digitalWrite(IN3, LOW);
    digitalWrite(IN4, LOW);
}

```

// ενεργοποίηση της εντολής «Αύξηση ταχύτητας»  
// έξοδος από τη δομή ελέγχου switch

// προετοιμασία του δέκτη IR για λήψη του επόμενου κωδικού

// εμφάνιση στη σειριακή οθόνη του Arduino της λέξης "forward"

// Εντολές για «Κίνηση μπροστά» (βλ. πίνακα σελ. 4)

// εμφάνιση στη σειριακή οθόνη του Arduino της λέξης "backward"

// Εντολές για «Κίνηση πίσω» (βλ. πίνακα σελ. 4)

// εμφάνιση στη σειριακή οθόνη του Arduino της λέξης "turnLeft"

// Εντολές για «Στροφή αριστερά» (βλ. πίνακα σελ. 4)

// εμφάνιση στη σειριακή οθόνη του Arduino της λέξης "turnRight"

// Εντολές για «Στροφή δεξιά» (βλ. πίνακα σελ. 4)

// εμφάνιση στη σειριακή οθόνη του Arduino της λέξης "stop"

// Εντολές για «Σταμάτημα» (βλ. πίνακα σελ. 4)

```

void speedUp(){
  Serial.println("speedUp");           // εμφάνιση στη σειριακή οθόνη του Arduino της λέξης "speedUp"
  speed+=10;                           // αύξηση της ταχύτητας κατά 10
  if(speed>255) speed =255;            // έλεγχος αν η ταχύτητα είναι μεγαλύτερη από 255
  analogWrite(ENA, speed);              // εντολές για Αύξηση ταχύτητας
  analogWrite(ENB, speed);              // (βλ. πίνακα σελ. 4)
}

```

```

void speedDown(){
  Serial.println("speedDown");         // εμφάνιση στη σειριακή οθόνη του Arduino της λέξης "speedDown"
  speed-=10;                           // μείωση της ταχύτητας κατά 10
  if(speed<0) speed =0;                 // έλεγχος αν η ταχύτητα είναι μικρότερη από 0
  analogWrite(ENA, speed);              // εντολές για Μείωση ταχύτητας
  analogWrite(ENB, speed);              // (βλ. πίνακα σελ. 4)
}

```

### Υλοποίηση και δοκιμή του υλικού της εφαρμογής και του κώδικα

Αφού συγκεντρώθηκαν όλα τα υλικά που αναφέρονται στην αρχή του φύλλου έργου, έγιναν οι συνδέσεις σύμφωνα με το σχεδιάγραμμα σύνδεσης των εξαρτημάτων (βλ. σελ. 9) που έκαναν οι μαθητές με το λογισμικό προσομοίωσης Fritzing, φορτώθηκε ο κώδικας του προγράμματος στο περιβάλλον IDE του Arduino και από τη σειριακή οθόνη διαπιστώθηκε η καλή λειτουργία και ανταπόκριση του υλικού στις εντολές του κώδικα.

