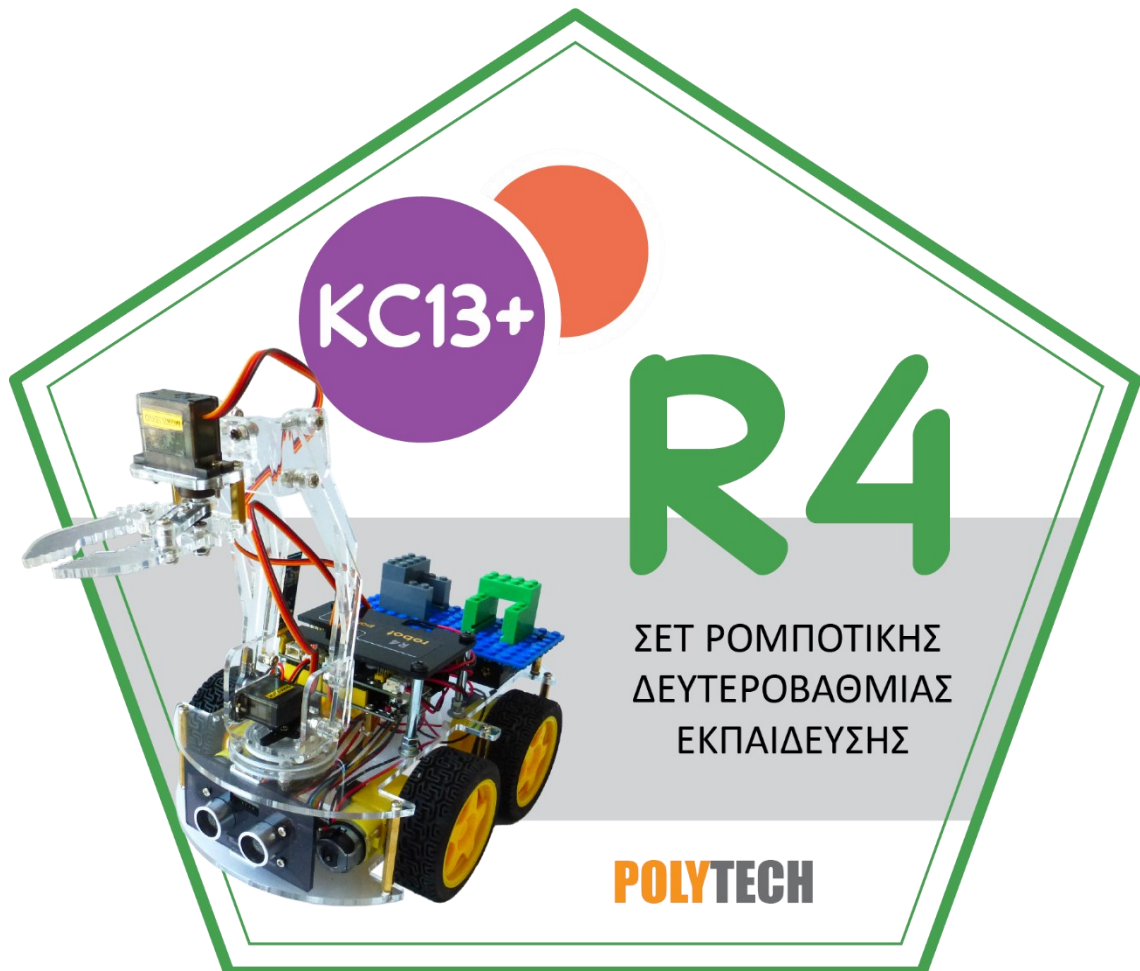


# ΕΝΟΤΗΤΑ 2 - ΔΡΑΣΤΗΡΙΟΤΗΤΕΣ ARDUINO IDE

---



## Περιεχόμενα

<b>1. Προγραμματισμός δραστηριοτήτων με Arduino IDE .....</b>	<b>3</b>
Δραστηριότητα 1.1: Προγραμματισμός Φως LED .....	3
Δραστηριότητα 1.2: Προγραμματισμός Φως LED που αναβοσβήνει .....	6
Δραστηριότητα 1.3: Προσαρμογή Φωτεινότητα LED.....	8
Δραστηριότητα 1.4: Προγραμματισμός Ελέγχου Σερβοκινητήρα βάσης (Servo 3).....	12
Δραστηριότητα 1.4.1: Προγραμματισμός Ελέγχου Σερβοκινητήρα βάσης (Servo 3) με βιβλιοθήκες .....	17
Δραστηριότητα 1.5: Προγραμματισμός Ελέγχου Σερβοκινητήρα βραχίονα (Σερβοκινητήρας 2).....	19
Δραστηριότητα 1.5.1: Προγραμματισμός Ελέγχου Σερβοκινητήρα βραχίονα (Servo 2) με βιβλιοθήκες .....	22
Δραστηριότητα 1.6: Προγραμματισμός Ελέγχου Σερβοκινητήρα δαγκάνας (Σερβοκινητήρας 1).....	24
Δραστηριότητα 1.6.1: Προγραμματισμός Ελέγχου Σερβοκινητήρα δαγκάνας (Σερβοκινητήρας 1) με βιβλιοθήκες .....	27
Δραστηριότητα 1.7: Προγραμματισμός αισθητήρα υπέρηχων.....	29
Δραστηριότητα 1.8: Προγραμματισμός αισθητήρα υπερήχων και ενεργοποίηση LED .....	33
Δραστηριότητα 1.9: Προγραμματισμός αισθητήρα υπερήχων και ενεργοποίηση LED και βομβητή.....	36
Δραστηριότητα 1.10: Προγραμματισμός αισθητήρα παρακολούθησης γραμμής.....	39
Δραστηριότητα 1.11: Προγραμματισμός αισθητήρα παρακολούθησης γραμμής και οπτική ειδοποίηση.....	42
Δραστηριότητα 1.12: Προγραμματισμός αισθητήρα παρακολούθησης γραμμής και οπτική ειδοποίηση (2 LED).....	45
Δραστηριότητα 1.13: Προγραμματισμός αισθητήρα παρακολούθησης γραμμής και ηχητική ειδοποίηση.....	48
Δραστηριότητα 1.14: Προγραμματισμός τηλεχειριστηρίου IR .....	52
Δραστηριότητα 1.15: Προγραμματισμός τηλεχειριστηρίου IR και οπτική ειδοποίηση .....	56
Δραστηριότητα 1.16: Προγραμματισμός τηλεχειριστηρίου IR και έλεγχος LED .....	58
Δραστηριότητα 1.17: Προγραμματισμός τηλεχειριστηρίου IR και συχνότητας ήχου βομβητή.....	61
Δραστηριότητα 1.18: Προγραμματισμός τηλεχειριστηρίου Bluetooth.....	64
Δραστηριότητα 1.19: Προγραμματισμός τηλεχειριστηρίου Bluetooth και ελέγχου LED (2 LED) .....	70
Δραστηριότητα 1.20: Προγραμματισμός τηλεχειριστηρίου Bluetooth και ελέγχου βομβητή.....	73
Δραστηριότητα 1.21: Πρόγραμμα οδήγησης κινητήρα.....	76
Δραστηριότητα 1.22: Πρόγραμμα οδήγησης κινητήρα και ελέγχου κίνησης.....	80
Δραστηριότητα 1.23: Πρόγραμμα οδήγησης κινητήρα και ηχητική ειδοποίηση.....	84
Δραστηριότητα 1.24: Σχεδιασμός κατά της πτώσης.....	88
Δραστηριότητα 1.25: Σχεδιασμός κατά της πτώσης και ηχητική ειδοποίηση.....	92
Δραστηριότητα 1.26: Προγραμματισμός παρακολούθησης γραμμής έξυπνου αυτοκινήτου .....	96

Δραστηριότητα 1.27: Προγραμματισμός παρακολούθησης γραμμής έξυπνου αυτοκινήτου και ηχητική ειδοποίηση .....	100
Δραστηριότητα 1.28: Προγραμματισμός παρακολούθησης εμποδίου .....	104
Δραστηριότητα 1.29: Προγραμματισμός παρακολούθησης εμποδίου και ηχητική ειδοποίηση .....	107
Δραστηριότητα 1.30: Έξυπνο αυτοκίνητο παρακολούθησης εμποδίου και ταυτόχρονη ηχητική και οπτική ειδοποίηση.....	110
Δραστηριότητα 1.31: Έξυπνο αυτοκίνητο αποφυγής εμποδίων και ηχητική ειδοποίηση .....	114
Δραστηριότητα 1.32: Έξυπνο αυτοκίνητο αποφυγής εμποδίων II .....	119
Δραστηριότητα 1.33: Έλεγχος έξυπνου αυτοκινήτου μέσω της εφαρμογής R4.....	122
Δραστηριότητα 1.34: Έλεγχος έξυπνου αυτοκινήτου μέσω της εφαρμογής R4 II .....	124
Δραστηριότητα 1.35: Έλεγχος έξυπνου αυτοκινήτου μέσω της εφαρμογής R4 III.....	127
Δραστηριότητα 1.36: Έλεγχος έξυπνου αυτοκινήτου μέσω της εφαρμογής R4 (4) .....	130
Δραστηριότητα 1.37: Έλεγχος έξυπνου αυτοκινήτου μέσω της εφαρμογής R4 (5) .....	133
Δραστηριότητα 1.38: Έλεγχος αυτοκινήτου με το τηλεχειριστήριο IR.....	136
Δραστηριότητα 1.39: Έλεγχος αυτοκινήτου με το τηλεχειριστήριο IR (2).....	139
Δραστηριότητα 1.40: Έλεγχος αυτοκινήτου με το τηλεχειριστήριο IR (3).....	142
Δραστηριότητα 1.41: Έλεγχος αυτοκινήτου με το τηλεχειριστήριο IR (4).....	145
Δραστηριότητα 1.42: Έλεγχος αυτοκινήτου με το τηλεχειριστήριο IR (5).....	148
Δραστηριότητα 1.43: Έλεγχος ταχύτητας έξυπνου αυτοκινήτου (1) .....	152
Δραστηριότητα 1.44: Έλεγχος ταχύτητας έξυπνου αυτοκινήτου (2) .....	156
Δραστηριότητα 1.45: Έλεγχος Σερβοκινητήρων μέσω Bluetooth(1).....	160
Δραστηριότητα 1.46: Έλεγχος Σερβοκινητήρων μέσω Bluetooth (2).....	163
Δραστηριότητα 1.47: Έλεγχος Σερβοκινητήρων μέσω Bluetooth (3) .....	166
Δραστηριότητα 1.48: Έλεγχος Σερβοκινητήρων βραχίονα μέσω Bluetooth.....	169
Δραστηριότητα 1.49: Λειτουργία Μνήμης (1).....	173
Δραστηριότητα 1.50: Λειτουργία Μνήμης (2).....	178
Δραστηριότητα 1.51: Λειτουργία Μνήμης (3).....	183
Δραστηριότητα 1.52: Λειτουργία Μνήμης (4).....	188
Δραστηριότητα 1.53: Έξυπνο αυτοκίνητο ελέγχου ρομποτικού βραχίονα .....	194

## 1. Προγραμματισμός δραστηριοτήτων με Arduino IDE

**Σημείωση:** Οι παρακάτω δραστηριότητες έχουν ως σκοπό να χτίσουν σιγά σιγά τον τελικό κώδικα της εφαρμογής R4. Για το λόγο αυτό προγραμματίζεται η κάθε λειτουργία ξεχωριστά και μετά συνδυαστικά όλες μαζί.

Σε κάθε δραστηριότητα προγραμματίζονται μόνο οι συσκευές που αναφέρονται στο διάγραμμα καλωδίωσης. Για παράδειγμα πρώτα θα μάθετε πως να προγραμματίζετε τους αισθητήρες ξεχωριστά ως μονάδες και μετά θα προσθέσετε και την κίνηση του ρομπότ.

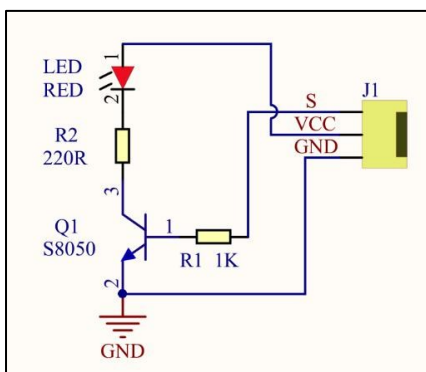
### Δραστηριότητα 1.1: Προγραμματισμός Φως LED

#### (1) Περιγραφή:



Για τους αρχάριους χρήστες του Arduino, αυτό είναι ένα βασικό πρόγραμμα --- LEDBlink/ αναβοσβήνω ένα LED. Το LED, συντομογραφία των διόδων εκπομπής φως, αποτελείται από χημικές ενώσεις των Ga, As, P, N κ.α. Η λυχνία LED μπορεί να αναβοσβήνει με διαφορετικό ρυθμό μεταβάλλοντας τον χρόνο καθυστέρησης στον κώδικα δοκιμής.

#### (2) Προδιαγραφή:





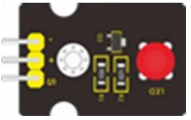


Διεπαφή ελέγχου: ψηφιακή θύρα

Τάση λειτουργίας: DC 3,3-5V

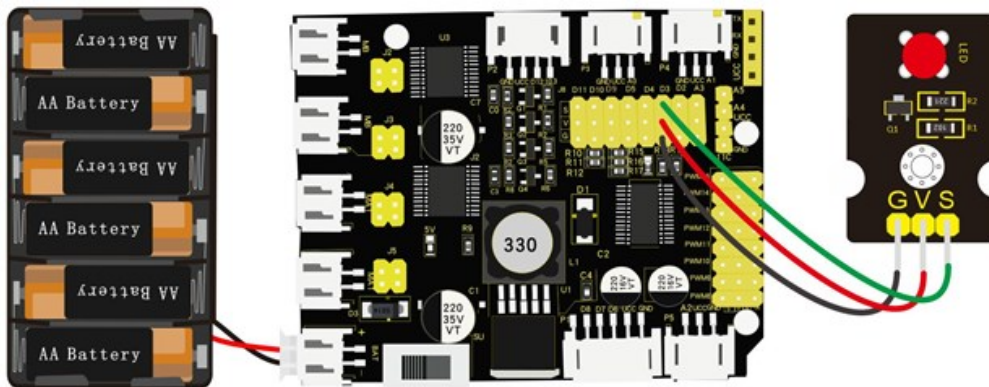
Διάσταση ακίδων: 2,54 mm

Χρώμα LED: κόκκινο

## (3) Τι χρειάζεστε:

Ελεγκτής	Πλακέτα οδήγησης κινητήρων	Κοκκινο Led	Καλώδιο USB	3pin F-F Dupont Line
				

## (4) Διάγραμμα καλωδίωσης:



Οι ακροδέκτες -, + και S της μονάδας LED συνδέονται στα G, V και D3 της πλακέτας οδήγησης. Συνδέστε την πλακέτα οδήγησης στον υπολογιστή σας με το καλώδιο USB.

## (5) Κώδικας δοκιμής:


```

/*
Δραστηριότητα 1.1
Φως LED
*/
void setup()
{
  pinMode(3, OUTPUT); // αρχικοποιήστε τον ακροδέκτη 3 ως έξοδο(OUTPUT).
}
void loop() // η συνάρτηση βρόχου εκτελείται ξανά και ξανά για πάντα
{
  digitalWrite(3, HIGH); // ανάβει η λυχνία LED (HIGH είναι το επίπεδο τάσης)
  delay(1000); // περιμένει ένα δευτερόλεπτο
  digitalWrite(3, LOW); // απενεργοποιήστε τη λυχνία LED κάνοντας την τάση χαμηλή(Low)
  delay(1000); // περιμένει ένα δευτερόλεπτο
}

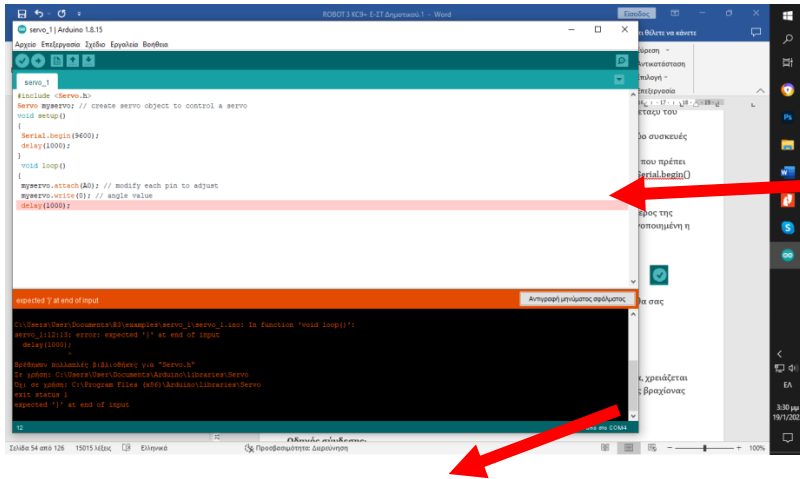
```

**Εναλλακτικά:** Μέσα στον φάκελο «ΚΩΔΙΚΕΣ ARDUINO» μπορείτε να ανοίξετε το αρχείο κάνοντας διπλό κλικ πάνω στο αρχείο  M\_1.1 .

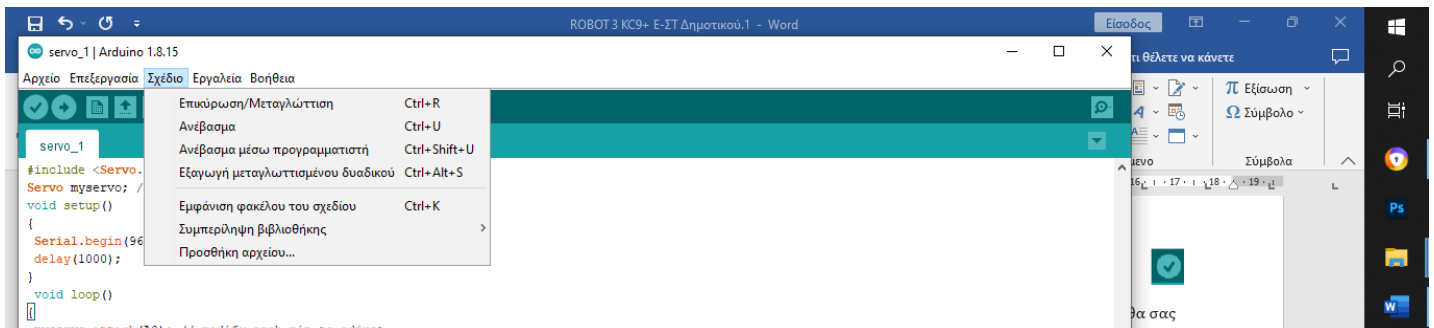
**(6) Αποτέλεσμα δοκιμής:**

Μόλις γράψετε τον κώδικα, πρώτα ελέγξτε για τυχόν σφάλματα επιλέγοντας το εικονίδιο  **Επικύρωση**.

Αν ο κώδικας που γράψατε έχει σφάλματα, τότε κάτω στη μαύρη οθόνη το πρόγραμμα θα σας ενημερώνει για το σφάλμα.



Αν ο κώδικας είναι σωστός, τότε επιλέξτε Σχέδιο → Ανέβασμα



Εναλλακτικά επιλέξτε το εικονίδιο  **Ανέβασμα**

Το LED αναβοσβήνει με διάστημα 1 δευτερολέπτου.

**(7) Επεξήγηση κώδικα:**


- pinMode(ledPin, OUTPUT)** - Αυτή η λειτουργία υποδηλώνει ότι η ακίδα είναι INPUT ή OUTPUT.
- digitalWrite(ledPin, HIGH)** - Όταν το pin είναι OUTPUT, μπορούμε να το θέσουμε σε HIGH (έξοδος 5V) ή LOW (έξοδος 0V).

## Δραστηριότητα 1.2: Προγραμματισμός Φως LED που αναβοσβήνει

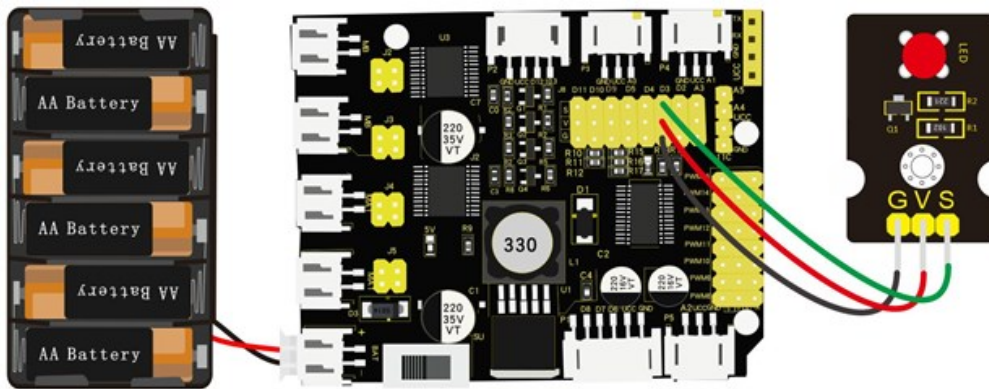
### (1) Περιγραφή:

Σε αυτή τη δραστηριότητα θα ελέγξετε τον ρυθμό που αναβοσβήνει το LED τροποποιώντας τον χρόνο καθυστέρησης.

### (2) Τι χρειάζεστε:

Ελεγκτής	Πλακέτα οδήγησης κινητήρων	Κοκκινο Led	Καλώδιο USB	3pin F-F Dupont Line
				

### (3) Διάγραμμα καλωδίωσης:




Οι ακροδέκτες -, + και S της μονάδας LED συνδέονται στα G, V και D3 της πλακέτας οδήγησης. Συνδέστε την πλακέτα οδήγησης στον υπολογιστή σας με το καλώδιο USB.

### (4) Κώδικας δοκιμής:

```
/*
 Δραστηριότητα 1.2
 Προγραμματισμός Φως LED που αναβοσβήνει
 */
void setup() { // αρχικοποιήστε τον ψηφιακό ακροδέκτη 11 ως έξοδο.
  pinMode(3, OUTPUT);
}

// η συνάρτηση βρόχου εκτελείται ξανά και ξανά για πάντα
void loop()
{ digitalWrite(3, HIGH); // ανάβει η λυχνία LED (HIGH είναι το επίπεδο τάσης)
  delay(100); // αναμονή για 0.1 δευτερόλεπτο
  digitalWrite(3, LOW); // απενεργοποιήστε τη λυχνία LED κάνοντας την τάση χαμηλή (LOW)
  delay(100); // αναμονή για 0.1 δευτερόλεπτο
} //*****
```

**Εναλλακτικά:** Μέσα στον φάκελο «ΚΩΔΙΚΕΣ ARDUINO» μπορείτε να ανοίξετε το αρχείο κάνοντας διπλό κλικ πάνω στο αρχείο  M\_1.2 .

**(5)Αποτέλεσμα δοκιμής:**

Ανεβάστε σωστά τον κώδικα χωρίς σφάλματα. Το LED αναβοσβήνει πιο γρήγορα σε σύγκριση με την προηγούμενη δραστηριότητα. Επομένως η καθυστέρηση του χρόνου (delay) μπορεί να επηρεάσει τη συχνότητα που αναβοσβήνει φλας.

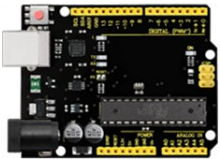

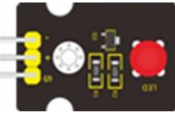





## Δραστηριότητα 1.3: Προσαρμογή Φωτεινότητα LED

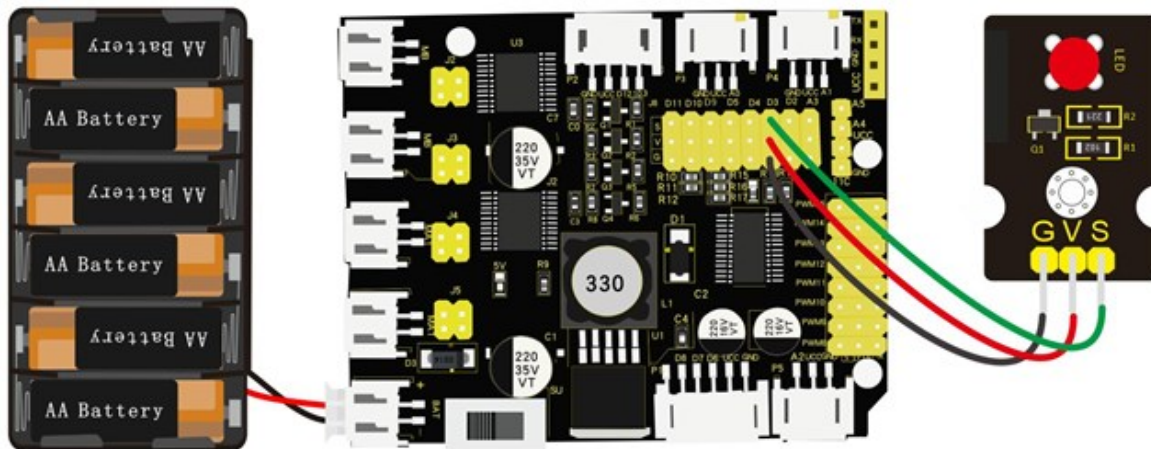
### (1) Περιγραφή:

Σε αυτή τη δραστηριότητα, θα ελέγξουμε τη φωτεινότητα των LED μέσω PWM για την προσομοίωση του εφέ αναπνοής. Ομοίως, μπορείτε να αλλάξετε το μήκος του βήματος και τον χρόνο καθυστέρησης στον κώδικα, ώστε να δείξετε διαφορετικό εφέ αναπνοής. Το PWM είναι ένα μέσο ελέγχου της αναλογικής εξόδου μέσω ψηφιακών μέσων.

### (2) Τι χρειάζεστε:

Ελεγκτής	Πλακέτα οδήγησης κινητήρων	Κοκκινο Led	Βάση μπαταρίας	Καλώδιο USB	3pin F-F Dupont Line
					

### (3) Διάγραμμα καλωδίωσης:



Οι ακροδέκτες -, + και S της μονάδας LED συνδέονται στα G, V και D3 της πλακέτας οδήγησης. Συνδέστε την πλακέτα οδήγησης στον υπολογιστή σας με το καλώδιο USB.

### (4) Κώδικας δοκιμής:

/\*

Δραστηριότητα 1.3

Προσαρμογή Φωτεινότητας LED

\*/

```
int ledPin = 3; // Ορίστε τον ακροδέκτη LED στο D3
```

```
int value;
```

```
void setup () {
```

```
  pinMode (ledPin, OUTPUT); // αρχικοποίηση της ακίδας led ως έξοδο.
```

```
}
```


```
void loop () {
```

```

for (value = 0; value <255; value = value + 1) {
  analogWrite (ledPin, value); // το φως LED ανάβουν σταδιακά
  delay (5); // καθυστέρηση 5MS
}
for (value = 255; value> 0; value = value-1) {
  analogWrite (ledPin, value); // το φως LED σβήνει σταδιακά
  delay (5); // καθυστέρηση 5MS
}
}

```

**Εναλλακτικά:** Μέσα στο φάκελο «ΚΩΔΙΚΕΣ ARDUINO» μπορείτε να ανοίξετε το αρχείο κάνοντας

διπλό κλικ πάνω στο αρχείο  M\_1.3

### (5) Αποτέλεσμα δοκιμής:

Ανεβάστε σωστά τον κώδικα χωρίς σφάλματα. Το LED σταδιακά γίνεται πιο φωτεινό και μετά πιο σκούρο, όπως η ανθρώπινη αναπνοή, αντί να ανάβει και να σβήνει αμέσως.

### (6) Επεξήγηση κώδικα:

Όταν χρειάζεται να επαναλάβουμε κάποιες δηλώσεις, θα μπορούσαμε να χρησιμοποιήσουμε τη δήλωση FOR.

Η μορφή δήλωσης FOR φαίνεται παρακάτω:

```

  ①      ② condition is true      ④
FOR (cycle initialization; cycle condition; cycle adjustment statement) {
  ③ loop body statement; ←
}

```

ΓΙΑ κυκλική ακολουθία:

Round1 : 1→2→3→4

Round2 : 2→3→4

...

Μέχρι να μην καθοριστεί ο αριθμός 2, ο βρόχος "for" έχει τελειώσει.

Αφού μάθετε αυτήν την διαταγή, επιστρέψτε στον κώδικα:

```
for (int value= 0; value< 255; value=value+1){
```

```
...}
```

```
for (int value= 255; value>0; value=value-1){
```

```
...}
```

Οι δύο δηλώσεις «FOR» κάνουν την τιμή να αυξάνεται από το 0 στο 255, μετά να μειώνεται από το 255 στο 0, μετά να αυξάνεται σε 255,...απεριόριστα.

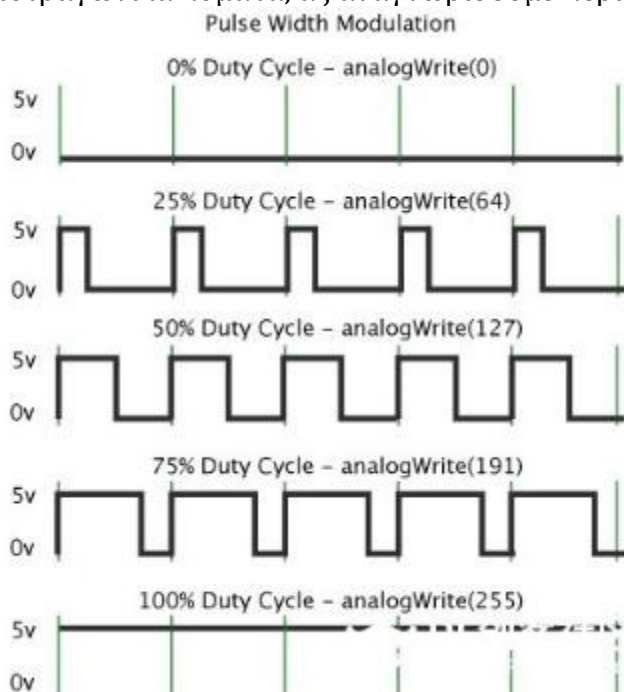
Υπάρχει μια νέα λειτουργία στα παρακάτω ----- analogWrite()

Γνωρίζουμε ότι η ψηφιακή θύρα έχει μόνο δύο καταστάσεις 0 και 1. Πώς να στείλετε μια αναλογική τιμή σε μια ψηφιακή τιμή; Εδώ, αυτή η λειτουργία είναι απαραίτητη. Ας παρατηρήσουμε την πλακέτα του Arduino και ας βρούμε σημειωμένες 6 ακίδες “ ~ ” που μπορεί να εξάγει σήματα PWM.

Μορφή λειτουργίας ως εξής:

`analogWrite(pin,value)`

`analogWrite()` χρησιμοποιείται για να γράψει μια αναλογική τιμή από 0~255 για θύρα PWM, επομένως η τιμή είναι στην περιοχή 0~255. Προσοχή να γράφετε μόνο τις ψηφιακές ακίδες με λειτουργία PWM, όπως οι ακίδες 3, 5, 6, 9, 10, 11. Το PWM είναι μια τεχνολογία λήψης αναλογικής ποσότητας μέσω ψηφιακής μεθόδου. Ο ψηφιακός έλεγχος σχηματίζει ένα τετραγωνικό κύμα και το σήμα τετραγωνικού κύματος έχει μόνο δύο καταστάσεις ενεργοποίησης και απενεργοποίησης (δηλαδή υψηλά ή χαμηλά επίπεδα). Με τον έλεγχο της αναλογίας της διάρκειας ενεργοποίησης και απενεργοποίησης, μπορεί να προσομοιωθεί μια τάση που κυμαίνεται από 0 έως 5 V. Ο χρόνος ενεργοποίησης (ακαδημαϊκά αναφέρεται ως υψηλό επίπεδο) ονομάζεται πλάτος παλμού, επομένως το PWM ονομάζεται επίσης διαμόρφωση πλάτους παλμού. Μέσα από τα ακόλουθα πέντε τετραγωνικά κύματα, ας αναγνωρίσουμε περισσότερα για το PWM.



Στο παραπάνω σχήμα, η πράσινη γραμμή αντιπροσωπεύει μια περίοδο και η τιμή της `analogWrite()` αντιστοιχεί σε ένα ποσοστό που ονομάζεται επίσης Duty Cycle. Ο κύκλος λειτουργίας υποδηλώνει ότι η διάρκεια υψηλού επιπέδου διαιρείται με τη διάρκεια χαμηλού επιπέδου σε έναν κύκλο. Από πάνω προς τα κάτω, ο κύκλος λειτουργίας του πρώτου τετραγωνικού κύματος είναι 0% και η αντίστοιχη τιμή του είναι 0. Η φωτεινότητα των LED είναι η χαμηλότερη, δηλαδή απενεργοποιείται. Όσο περισσότερο χρόνο διαρκεί το υψηλό επίπεδο, τόσο πιο φωτεινό είναι το LED. Επομένως, ο τελευταίος κύκλος λειτουργίας είναι 100%, που αντιστοιχεί σε 255, το LED είναι το πιο φωτεινό. 25% σημαίνει πιο σκούρο. Το PWM χρησιμοποιείται κυρίως για τη ρύθμιση της φωτεινότητας των LED ή της ταχύτητας περιστροφής του κινητήρα. Διαδραματίζει ζωτικό ρόλο στον έλεγχο του ρομπότ R4. Πιστεύω ότι ανυπομονείς να μπει στην επόμενη δραστηριότητα.

## (7) Πρακτική επέκτασης:


Ας τροποποιήσουμε την τιμή του χρόνου καθυστέρησης και παρατηρήστε πώς αλλάζει το LED.

```

/*
δραστηριότητα 1.3.β
pwm
*/
int ledPin = 3; // Define the LED pin at D3
void setup(){
  pinMode(ledPin, OUTPUT); // initialize ledpin as an output.
}
void loop(){
  for(int value = 0; value <255; value = value + 1){
    analogWrite(ledPin, value); // LED lights gradually light up
    delay(30); // delay 30MS
  }
  for(int value=255; value>0;value=value-1){
    analogWrite(ledPin, value); // LED gradually goes out
    delay(30); // delay 30MS
  }
}
} //*****

```

Ανεβάστε τον κώδικα και παρατηρήστε το LED να αναβοσβήνει πιο αργά.

**Εναλλακτικά:** Μέσα στον φάκελο «ΚΩΔΙΚΕΣ ARDUINO» μπορείτε να ανοίξετε το αρχείο κάνοντας διπλό κλικ πάνω στο αρχείο  M\_1.3.b .

## Δραστηριότητα 1.4: Προγραμματισμός Ελέγχου Σερβοκινητήρα βάσης (Σερβοκινητήρας 3)

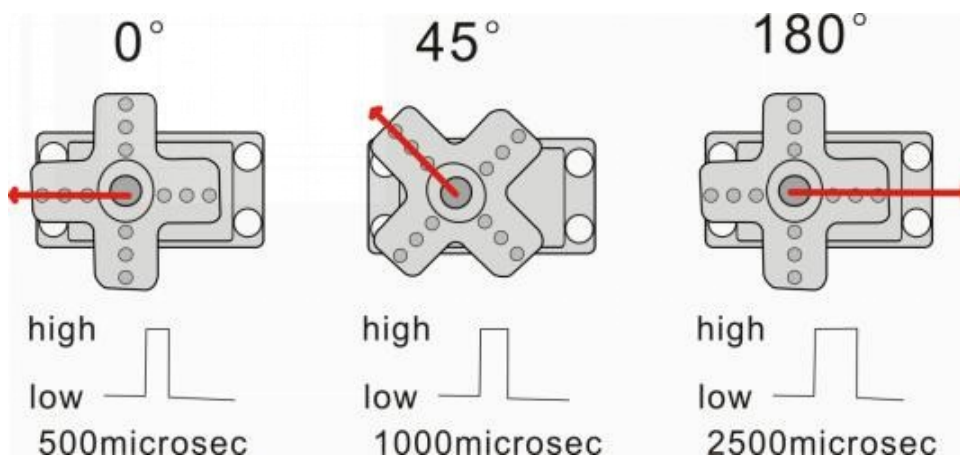
### (1) Περιγραφή:

Σε αυτή τη δραστηριότητα θα δείτε πως μπορείτε να ελέγξετε τη γωνία του σερβοκινητήρα (σερβοκινητήρας βάσης).

Ο σερβοκινητήρας είναι ένας περιστροφικός ενεργοποιητής ελέγχου θέσης. Η αρχή λειτουργίας του είναι ότι ο σερβοκινητήρας λαμβάνει το σήμα που αποστέλλεται από το MCU ή τον δέκτη και παράγει ένα σήμα αναφοράς με περίοδο 20 ms και πλάτος 1,5 ms, στη συνέχεια συγκρίνει την τάση πόλωσης DC που αποκτήθηκε με την τάση του ποτενσιόμετρου και λαμβάνει την έξοδο διαφοράς τάσης. Όταν η ταχύτητα του κινητήρα είναι σταθερή, το ποτενσιόμετρο κινείται για να περιστραφεί μέσω του μειωτήρα κατάλληλου μεγέθους, με αποτέλεσμα η διαφορά τάσης να είναι μηδενική και ο κινητήρας να σταματά να περιστρέφεται.

Γενικά, το εύρος γωνίας περιστροφής του σερβοκινητήρα είναι  $0^\circ$  - $180^\circ$ .

Η γωνία περιστροφής του σερβοκινητήρα ελέγχεται ρυθμίζοντας τον κύκλο λειτουργίας του σήματος PWM (Pulse-Width Modulation). Ο τυπικός κύκλος του σήματος PWM είναι 20ms (50Hz). Θεωρητικά, το πλάτος κατανέμεται μεταξύ 1ms-2ms, αλλά στην πραγματικότητα είναι μεταξύ 0,5ms-2,5ms. Το πλάτος αντιστοιχεί στη γωνία περιστροφής από  $0^\circ$  έως  $180^\circ$ . Αλλά σημειώστε ότι για κινητήρα διαφορετικής μάρκας, το ίδιο σήμα μπορεί να έχει διαφορετική γωνία περιστροφής.



Οι αντίστοιχες σερβογωνίες φαίνονται παρακάτω:

High level time	Servo angle
0.5ms	0 degree
1ms	45 degree
1.5ms	90 degree
2ms	135 degree
2.5ms	180 degree

## (2) Προδιαγραφή:

Τάση λειτουργίας: DC 4,8V ~ 6V

Εύρος γωνίας λειτουργίας: περίπου 180 ° (στα 500 → 2500 μsec)

Εύρος πλάτους παλμού: 500 → 2500 μsec

Ταχύτητα χωρίς φορτίο: 0,12 ± 0,01 sec / 60 (DC 4,8V) 0,1 ± 0,01 sec / 60 (DC 6V)


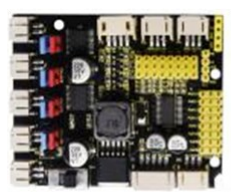


Ρεύμα χωρίς φορτίο: 200 ± 20mA (DC 4,8V) 220 ± 20mA (DC 6V)

Ροπή διακοπής: 1,3 ± 0,01kg • cm (DC 4,8V) 1,5 ± 0,1kg • cm (DC 6V)

Ρεύμα διακοπής: ≤ 850mA (DC 4,8V) ≤ 1000mA (DC 6V)

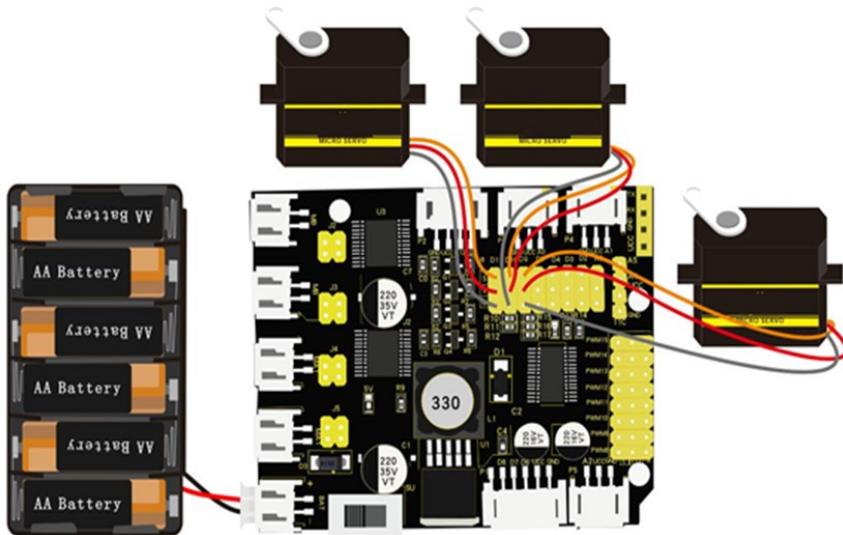
Ρεύμα αναμονής: 3 ± 1mA (DC 4,8V) 4 ± 1mA (DC 6V)

## (3) Τι χρειάζεστε:

Ελεγκτής	Πλακέτα οδήγησης κινητήρων	Σερβοκινητήρας	Βάση μπαταρίας	Καλώδιο USB
				

## (4) Διάγραμμα καλωδίωσης:

Σε αυτή τη δραστηριότητα θα ελέγξετε μόνο τον σερβομηχανισμό 3, στις επόμενες θα δείτε πως μπορείτε να επεκτείνετε αυτόν τον έλεγχο και στους υπόλοιπους σερβοκινητήρες.



**Σημείωση καλωδίωσης:** η καφέ γραμμή του σερβοκινητήρα συνδέεται με το Gnd(G), η κόκκινη γραμμή συνδέεται με το V και η πορτοκαλί με το S.  
 Συνδέστε την πλακέτα οδήγησης στον υπολογιστή σας με το καλώδιο USB.  
 Ο σερβοκινητήρας πρέπει να συνδεθεί με εξωτερική τροφοδοσία λόγω της μεγάλης απαίτησης του για ρεύμα. Το ρεύμα που παίρνει από τη σύνδεση του καλωδίου USB με τον υπολογιστή σας δεν είναι αρκετό για να τον κινήσει.

#### (5) Κώδικας δοκιμής:

```


/*
Δραστηριότητα 1.4
Έλεγχος Σερβοκινητήρα βάσης (Σερβοκινητήρας 3)
*/
#define servoPin 9 // ακροδέκτης 9 ελέγχει τον σερβοκινητήρα της βάσης του βραχίονα
int pos;           // μεταβλητή γωνίας του σερβοκινητήρα
int pulsewidth;   // μεταβλητή εύρους παλμού του σερβοκινητήρα

void setup() {
  pinMode(servoPin, OUTPUT); // ορίζει τους ακροδέκτες του σερβοκινητήρα ως έξοδο (OUTPUT)
  procedure(0); // ορίζει τη γωνία του σερβοκινητήρα σε 0°
}

void loop() {
  for (pos = 0; pos <= 180; pos += 1) { // από 0 μοίρες έως 180 μοίρες
    // σε βήματα του 1 βαθμού
    procedure(pos); // λέει στον σερβοκινητήρα να πάει στη θέση που αναφέρεται στη
    μεταβλητή 'pos'
    delay(15); // έλεγχος της ταχύτητας περιστροφής του σερβοκινητήρα
  }
  for (pos = 180; pos >= 0; pos -= 1) { // από 180 μοίρες έως 0 μοίρες
    procedure(pos); // λέει στον σερβοκινητήρα να πάει στη θέση που αναφέρεται στη
    μεταβλητή 'pos'
    delay(15);
  }
}
// Function για τον έλεγχο του σέρβο

```

```
void procedure(int myangle) {  
  pulsewidth = myangle * 11 + 500; // Υπολογισμός τιμής εύρους παλμού  
  digitalWrite(servoPin, HIGH);  
  delayMicroseconds(pulsewidth); // η διάρκεια του υψηλού επιπέδου είναι το εύρος παλμού  
  digitalWrite(servoPin, LOW);  
  delay((20 - pulsewidth / 1000)); // η περίοδος είναι 20ms, οπότε το χαμηλό επίπεδο διαρκεί για το  
  υπόλοιπο του χρόνου  
} //*****
```

**Εναλλακτικά:** Μέσα στον φάκελο «ΚΩΔΙΚΕΣ ARDUINO» μπορείτε να ανοίξετε το αρχείο κάνοντας διπλό κλικ πάνω στο αρχείο  M\_1.4



**(6) Αποτέλεσμα δοκιμής:**



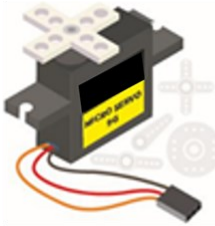
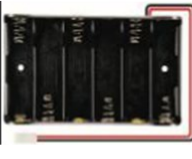

Ανεβάσατε τον κώδικα δοκιμής με επιτυχία. Ο σερβοκινητήρας κινείται εμπρός και πίσω στην περιοχή από  $0^\circ$  έως  $180^\circ$ , εάν έχετε συνδεδεμένο τον σερβοκινητήρα πάνω στο ρομπότ τότε θα δείτε και το βραχίονα του ρομπότ να μετακινείται αριστερά δεξιά από  $0^\circ$  έως  $180^\circ$ .

## Δραστηριότητα 1.4.1: Προγραμματισμός Ελέγχου Σερβοκινητήρα βάσης (Σερβοκινητήρας 3) με βιβλιοθήκες

### (1) Περιγραφή:

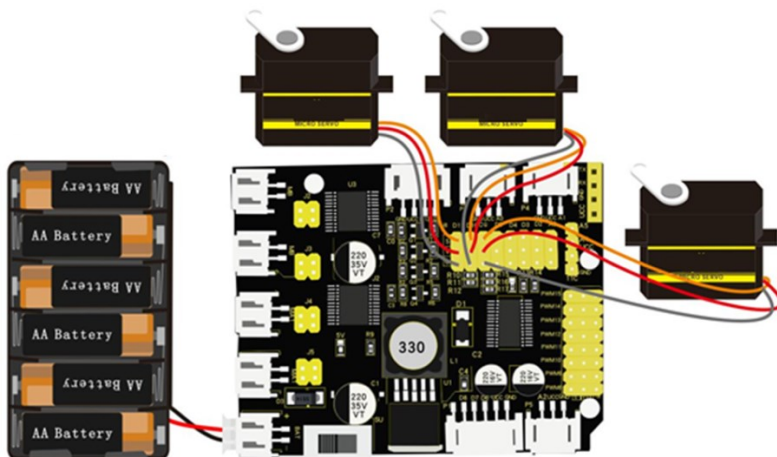
Σε αυτή τη δραστηριότητα, σε συνέχεια της προηγούμενης, θα προγραμματίσετε τον έλεγχο του Σερβοκινητήρα 3 εντάσσοντας τις βιβλιοθήκες στον κώδικα προγραμματισμού.

### (2) Τι χρειάζεστε:

Ελεγκτής	Πλακέτα οδήγησης κινητήρων	Σερβοκινητήρας	Βάση μπαταρίας	Καλώδιο USB
				

### (3) Διάγραμμα καλωδίωσης:

Σε αυτή τη δραστηριότητα θα ελέγξετε μόνο τον σερβοκινητήρα 3, στις επόμενες θα δείτε πως μπορείτε να επεκτείνετε αυτόν τον έλεγχο και στους υπόλοιπους σερβοκινητήρες.



**Σημείωση καλωδίωσης:** η καφέ γραμμή του σερβοκινητήρα συνδέεται με το Gnd(G), η κόκκινη γραμμή συνδέεται με το V και η πορτοκαλί με το S.

Συνδέστε την πλακέτα οδήγησης στον υπολογιστή σας με το καλώδιο USB.

Ο σερβοκινητήρας πρέπει να συνδεθεί με εξωτερική τροφοδοσία λόγω της μεγάλης απαίτησης του σε ρεύμα. Το ρεύμα που παίρνει από τη σύνδεση του καλωδίου USB με τον υπολογιστή σας δεν είναι αρκετό για να τον κινήσει.

**(4) Κώδικας δοκιμής:**

```

/*
Δραστηριότητα 1.4.1
Προγραμματισμός Ελέγχου Σερβοκινητήρα βάσης (Σερβοκινητήρας 3) με βιβλιοθήκες
*/
#include <Servo.h> // Βιβλιοθήκη για τη χρήση των σερβοκινητήρων
Servo myservo;    //δημιουργία αντικειμένου σερβοκινητήρα για τον έλεγχο ενός
σερβοκινητήρα
// Δώδεκα αντικείμενα σερβοκινητήρα μπορούν να δημιουργηθούν στις περισσότερες πλακέτες
int pos = 0;      // μεταβλητή για την αποθήκευση της θέσης του σερβοκινητήρα
void setup() {
  myservo.attach(9); // συνδέει τον σερβομηχανισμό στον ακροδέκτη 9 με το αντικείμενο του
σερβοκινητήρα
}
void loop() {
  for (pos = 0; pos <= 180; pos += 1) { // από 0 μοίρες έως 180 μοίρες
    // σε βήματα του 1 βαθμού
    myservo.write(pos); // λέει στον σερβοκινητήρα να πάει στη θέση που αναφέρεται στη
μεταβλητή 'pos'
    delay(15);          // περιμένει 15ms για να φτάσει ο σερβοκινητήρας στη θέση
  }
  for (pos = 180; pos >= 0; pos -= 1) { // από 180 μοίρες έως 0 μοίρες
    myservo.write(pos); // λέει στον σερβοκινητήρα να πάει στη θέση που
αναφέρεται στη μεταβλητή 'pos'
    delay(15);          // περιμένει 15ms για να φτάσει ο σερβοκινητήρας
στη θέση
  }
}
}

```

**Εναλλακτικά:** Μέσα στον φάκελο «ΚΩΔΙΚΕΣ ARDUINO» μπορείτε να ανοίξετε το αρχείο κάνοντας

διπλό κλικ πάνω στο αρχείο  M\_1.4.1 .

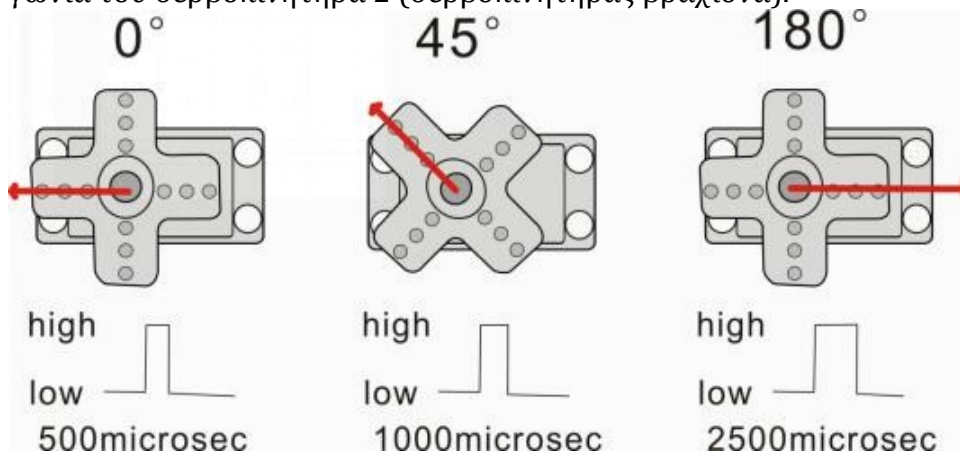
**(5) Αποτέλεσμα δοκιμής:**

Ανεβάστε σωστά τον κώδικα χωρίς σφάλματα. Ο σερβοκινητήρας κινείται εμπρός και πίσω στην περιοχή από 0° έως 180°, εάν έχετε συνδεδεμένο τον σερβοκινητήρα πάνω στο ρομπότ τότε θα δείτε και το βραχίονα του ρομπότ να μετακινείται αριστερά δεξιά από 0° έως 180°.

## Δραστηριότητα 1.5: Προγραμματισμός Ελέγχου Σερβοκινητήρα βραχίονα (Σερβοκινητήρας 2)

### (1) Περιγραφή:

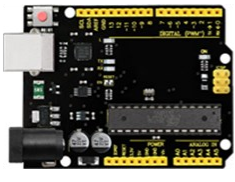
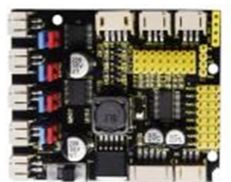



Σε αυτή τη δραστηριότητα, σε συνέχεια της προηγούμενης, θα δείτε πως μπορείτε να ελέγξετε τη γωνία του σερβοκινητήρα 2 (σερβοκινητήρας βραχίονα).



Οι αντίστοιχες σεργωνίες φαίνονται παρακάτω:

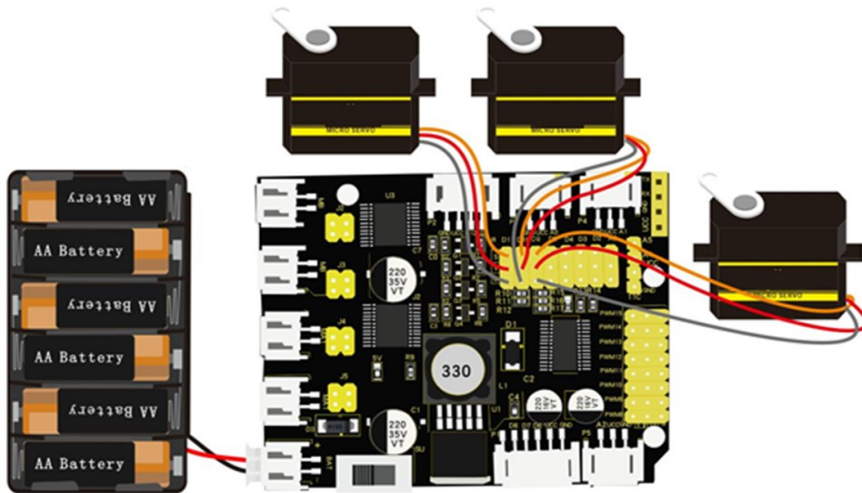
High level time	Servo angle
0.5ms	0 degree
1ms	45 degree
1.5ms	90 degree
2ms	135 degree
2.5ms	180 degree

### (2) Τι χρειάζεστε:

Ελεγκτής	Πλακέτα οδήγησης κινητήρων	Σερβοκινητήρας	Βάση μπαταρίας	Καλώδιο USB
				

**(3) Διάγραμμα καλωδίωσης:**

Σε αυτή τη δραστηριότητα θα ελέγξετε μόνο τον σερβοκινητήρα 2.



**Σημείωση καλωδίωσης:** η καφέ γραμμή του σερβοκινητήρα συνδέεται με το Gnd(G), η κόκκινη γραμμή συνδέεται με το V και η πορτοκαλί με το S.

Συνδέστε την πλακέτα οδήγησης στον υπολογιστή σας με το καλώδιο USB.

Ο σερβοκινητήρας πρέπει να συνδεθεί με εξωτερική τροφοδοσία λόγω της μεγάλης απαίτησης του σε ρεύμα. Το ρεύμα που παίρνει από τη σύνδεση του καλωδίου USB με τον υπολογιστή σας δεν είναι αρκετό για να τον κινήσει.


**(4) Κώδικας δοκιμής:**

```

/*
Δραστηριότητα 1.5
Έλεγχος Σερβοκινητήρα βραχίονα (Σερβοκινητήρας 2)
*/
#define servoPin 10 // ακροδέκτης σέρβο 10 είναι ελέγχει τον σερβοκινητήρα του βραχίονα
int pos; // μεταβλητή γωνίας του σερβοκινητήρα
int pulsewidth; // μεταβλητή εύρους παλμού του σερβοκινητήρα
void setup() {
  pinMode(servoPin, OUTPUT); // ορίζει τους ακροδέκτες του σερβοκινητήρα ως έξοδο (OUTPUT)
  procedure(0); // ορίζει τη γωνία του σερβοκινητήρα σε 0°
}
void loop() {
  for (pos = 0; pos <= 180; pos += 1) { // από 0 μοίρες έως 180 μοίρες
    // σε βήματα του 1 βαθμού
    procedure(pos); // λέει στον σερβοκινητήρα να πάει στη θέση που αναφέρεται στη
    μεταβλητή 'pos'
    delay(15); // έλεγχος της ταχύτητας περιστροφής του σερβοκινητήρα
  }
  for (pos = 180; pos >= 0; pos -= 1) { // από 180 μοίρες έως 0 μοίρες
    procedure(pos); // λέει στον σερβοκινητήρα να πάει στη θέση που αναφέρεται στη
    μεταβλητή 'pos'
    delay(15);
  }
}

```

```
// Function για τον έλεγχο του σερβοκινητήρα
void procedure(int myangle) {
  pulsewidth = myangle * 11 + 500; // Υπολογισμός τιμής εύρους παλμού
  digitalWrite(servoPin, HIGH);
  delayMicroseconds(pulsewidth); // η διάρκεια του υψηλού επιπέδου είναι το εύρος παλμού
  digitalWrite(servoPin, LOW);
  delay((20 - pulsewidth / 1000)); // η περίοδος είναι 20ms, οπότε το χαμηλό επίπεδο διαρκεί για το υπόλοιπο του χρόνου
} //*****
```

**Εναλλακτικά:** Μέσα στον φάκελο «ΚΩΔΙΚΕΣ ARDUINO» μπορείτε να ανοίξετε το αρχείο κάνοντας διπλό κλικ πάνω στο αρχείο  M\_1.5 .

### (5) Αποτέλεσμα δοκιμής:

Ανεβάστε σωστά τον κώδικα χωρίς σφάλματα. Ο σερβοκινητήρας κινείται εμπρός και πίσω στην περιοχή από 0° έως 180°, εάν έχετε συνδεδεμένο τον σερβοκινητήρα πάνω στο ρομπότ τότε θα δείτε και το βραχίονα του ρομπότ να μετακινείται πάνω κάτω.

### (1) Επεξήγηση κώδικα:

Το Arduino έρχεται με #include <Servo.h> (servo function and statement)

Ακολουθούν ορισμένες κοινές δηλώσεις της συνάρτησης servo:

1. **attach (interface)** — Διατίθεται σερβοδιασύνδεση, θύρες 9 και 10.

2. **write (angle)** — Η δήλωση για τη ρύθμιση της γωνίας περιστροφής του σερβοκινητήρα. Το εύρος γωνίας είναι από 0° έως 180°

3. **read ()** — χρησιμοποιείται για την ανάγνωση της γωνίας του σερβοκινητήρα. Διαβάστε την τιμή της εντολής «write()»

4. **attached ()** — Κρίνετε εάν η παράμετρος του servo αποστέλλεται στη διεπαφή του.

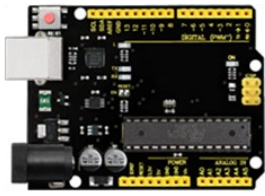
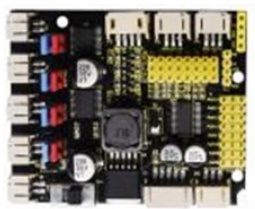



Σημείωση: Η παραπάνω γραπτή μορφή είναι «servovariablename, specificstatement ()», για παράδειγμα: myservo.attach(9)

## Δραστηριότητα 1.5.1: Προγραμματισμός Ελέγχου Σερβοκινητήρα βραχίονα (Σερβοκινητήρας 2) με βιβλιοθήκες

### (1) Περιγραφή:

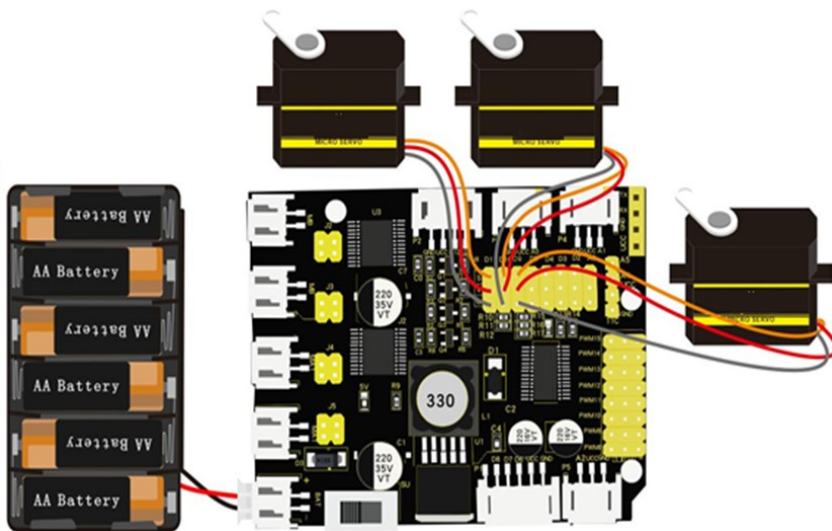
Σε αυτή τη δραστηριότητα, σε συνέχεια της προηγούμενης, θα προγραμματίσετε τον έλεγχο του σερβοκινητήρα 2 εντάσσοντας τις βιβλιοθήκες στον κώδικα προγραμματισμού.

### (2) Τι χρειάζεστε:

Ελεγκτής	Πλακέτα οδήγησης κινητήρων	Σερβοκινητήρας	Βάση μπαταρίας	Καλώδιο USB
				

### (3) Διάγραμμα καλωδίωσης:

Σε αυτή τη δραστηριότητα θα ελέγξετε μόνο τον σερβοκινητήρα 2.



**Σημείωση καλωδίωσης:** η καφέ γραμμή του σερβοκινητήρα συνδέεται με το Gnd(G), η κόκκινη γραμμή συνδέεται με το V και η πορτοκαλί με το S.

Συνδέστε την πλακέτα οδήγησης στον υπολογιστή σας με το καλώδιο USB.

Ο σερβοκινητήρας πρέπει να συνδεθεί με εξωτερική τροφοδοσία λόγω της μεγάλης απαίτησης του σε ρεύμα. Το ρεύμα που παίρνει από τη σύνδεση του καλωδίου USB με τον υπολογιστή σας δεν είναι αρκετό για να τον κινήσει.

**(4) Κώδικας δοκιμής:**


```

/*
Δραστηριότητα 1.5.1
Προγραμματισμός Ελέγχου Σερβοκινητήρα βραχίονα (Σερβοκινητήρας 2) με βιβλιοθήκες
*/
#include <Servo.h> // Βιβλιοθήκη για τη χρήση των σερβοκινητήρων
Servo myservo; // δημιουργία αντικειμένου σερβοκινητήρα για τον έλεγχο ενός σερβοκινητήρα
// Δώδεκα αντικείμενα σερβοκινητήρα μπορούν να δημιουργηθούν στις περισσότερες πλακέτες
int pos = 0; // // μεταβλητή για την αποθήκευση της θέσης του σερβοκινητήρα

void setup() {
  myservo.attach(9); // συνδέει τον σερβοκινητήρα στον ακροδέκτη 9 με το αντικείμενο του
  σερβοκινητήρα
}

void loop() {
  for (pos = 0; pos <= 180; pos += 1) { // από 0 μοίρες έως 180 μοίρες
    // σε βήματα του 1 βαθμού
    myservo.write(pos); // λέει στον σερβοκινητήρα να πάει στη θέση που αναφέρεται στη μεταβλητή
    'pos'
    delay(15); // περιμένει 15ms για να φτάσει ο σερβοκινητήρας στη θέση
  }
  for (pos = 180; pos >= 0; pos -= 1) { // από 180 μοίρες έως 0 μοίρες
    myservo.write(pos); // λέει στον σερβοκινητήρα να πάει στη θέση που αναφέρεται
    στη μεταβλητή 'pos'
    delay(15); // περιμένει 15ms για να φτάσει ο σερβοκινητήρας στη θέση
  }
}

```

**Εναλλακτικά:** Μέσα στον φάκελο «ΚΩΔΙΚΕΣ ARDUINO» μπορείτε να ανοίξετε το αρχείο κάνοντας διπλό κλικ πάνω στο αρχείο  M\_1.5.1 .

**(5) Αποτέλεσμα δοκιμής:**

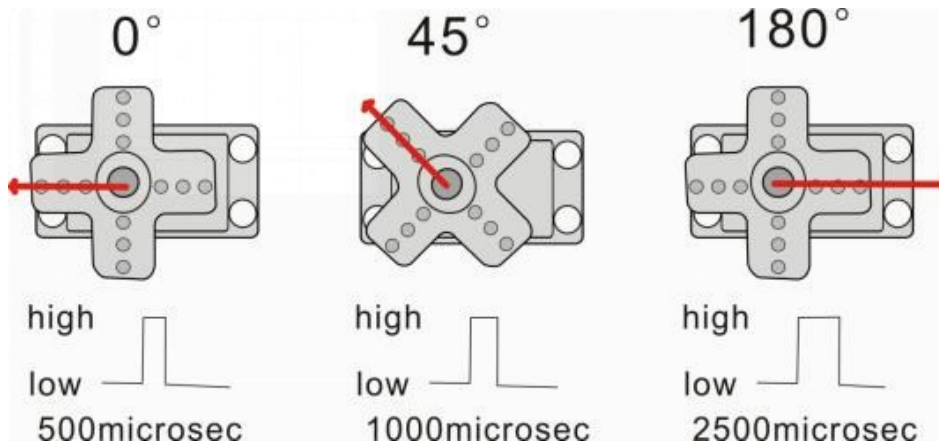
Ανεβάστε σωστά τον κώδικα χωρίς σφάλματα. Ο σερβοκινητήρας κινείται εμπρός και πίσω στην περιοχή από 0° έως 180°. Εάν έχετε συνδεδεμένο τον σερβοκινητήρα πάνω στο ρομπότ τότε θα δείτε και το βραχίονα του ρομπότ να μετακινείται πάνω κάτω.



## Δραστηριότητα 1.6: Προγραμματισμός Ελέγχου Σερβοκινητήρα δαγκάνας (Σερβοκινητήρας 1)

### (1) Περιγραφή:

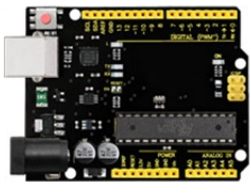
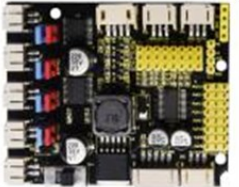
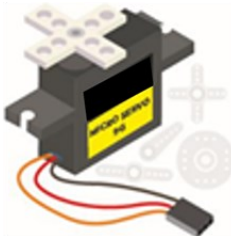


Σε αυτή τη δραστηριότητα, σε συνέχεια της προηγούμενης, θα δείτε πως μπορείτε να ελέγξετε τη γωνία του σερβοκινητήρα 1 (σερβοκινητήρας δαγκάνας).



Οι αντίστοιχες σεργογωνίες φαίνονται παρακάτω:

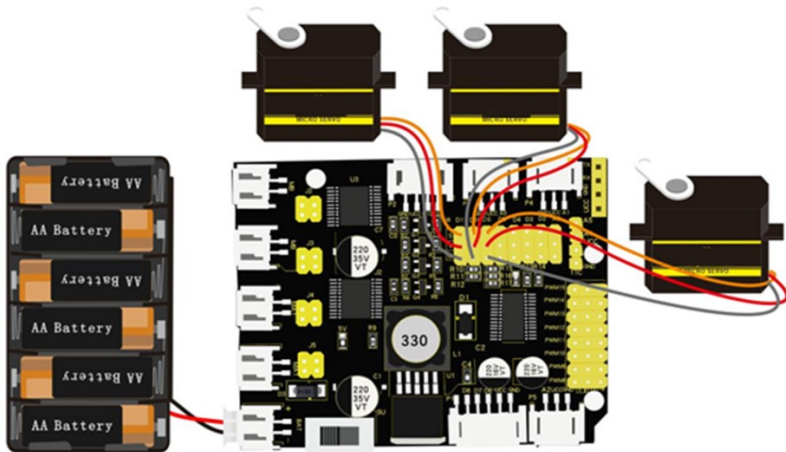
High level time	Servo angle
0.5ms	0 degree
1ms	45 degree
1.5ms	90 degree
2ms	135 degree
2.5ms	180 degree

### (2) Τι χρειάζεστε:

Ελεγκτής	Πλακέτα οδήγησης κινητήρων	Σερβοκινητήρας	Βάση μπαταρίας	Καλώδιο USB
				

**(3) Διάγραμμα καλωδίωσης:**

Σε αυτή τη δραστηριότητα θα ελέγξετε τον σερβοκινητήρα 1.



**Σημείωση καλωδίωσης:** η καφέ γραμμή του σερβοκινητήρα συνδέεται με το Gnd(G), η κόκκινη γραμμή συνδέεται με το V και η πορτοκαλί με το S.

Συνδέστε την πλακέτα οδήγησης στον υπολογιστή σας με το καλώδιο USB.

Ο σερβοκινητήρας πρέπει να συνδεθεί με εξωτερική τροφοδοσία λόγω της μεγάλης απαίτησής του σε ρεύμα. Το ρεύμα που παίρνει από τη σύνδεση του καλωδίου USB με τον υπολογιστή σας δεν είναι αρκετό για να τον κινήσει.

**(4) Κώδικας δοκιμής:**

```

/*
 Δραστηριότητα 1.6
 Έλεγχος Σερβοκινητήρα δαγκάνας (Σερβοκινητήρας 1)
 */
#define servoPin 11 //servo Pin
int pos; // μεταβλητή γωνίας του σερβοκινητήρα
int pulsewidth; // μεταβλητή εύρους παλμού του σερβοκινητήρα
void setup() {
  pinMode(servoPin, OUTPUT); // ορίζει τους ακροδέκτες του σερβοκινητήρα ως έξοδο (OUTPUT)
  procedure(0); // ορίζει τη γωνία του σερβοκινητήρα σε 0°
}
void loop() {
  for (pos = 0; pos <= 180; pos += 1) { // από 0 μοίρες έως 180 μοίρες
    // σε βήματα του 1 βαθμού
    procedure(pos); // λέει στον σερβοκινητήρα να πάει στη θέση που αναφέρεται στη
    μεταβλητή 'pos'
    delay(15); // έλεγχος της ταχύτητας περιστροφής του σερβοκινητήρα
  }
  for (pos = 180; pos >= 0; pos -= 1) { // από 180 μοίρες έως 0 μοίρες
    procedure(pos); // λέει στον σερβοκινητήρα να πάει στη θέση που αναφέρεται στη
    μεταβλητή 'pos'
    delay(15);
  }
}
// Function για τον έλεγχο του σερβοκινητήρα

```

```

void procedure(int myangle) {
  pulsewidth = myangle * 11 + 500; // Υπολογισμός τιμής εύρους παλμού
  digitalWrite(servoPin, HIGH);
  delayMicroseconds(pulsewidth); // η διάρκεια του υψηλού επιπέδου είναι το εύρος παλμού
  digitalWrite(servoPin, LOW);
  delay((20 - pulsewidth / 1000)); // η περίοδος είναι 20ms, οπότε το χαμηλό επίπεδο διαρκεί για το
  υπόλοιπο του χρόνου
} //*****

```

**Εναλλακτικά:** Μέσα στον φάκελο «ΚΩΔΙΚΕΣ ARDUINO» μπορείτε να ανοίξετε το αρχείο κάνοντας διπλό κλικ πάνω στο αρχείο  M\_1.6

#### (5) Αποτέλεσμα δοκιμής:

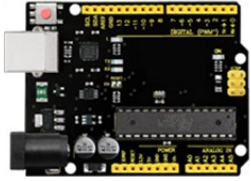
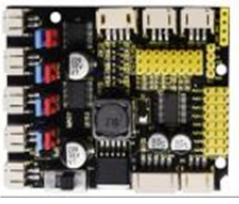
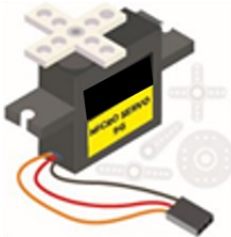


Ανεβάστε σωστά τον κώδικα χωρίς σφάλματα. Ο σερβοκινητήρας κινείται εμπρός και πίσω στην περιοχή από 0° έως 180°. Εάν έχετε συνδεδεμένο τον σερβοκινητήρα πάνω στο ρομπότ τότε θα δείτε και τη δαγκάνα του βραχίονα να ανοιγοκλείνει.

## Δραστηριότητα 1.6.1: Προγραμματισμός Ελέγχου Σερβοκινητήρα δαγκάνας (Σερβοκινητήρας 1) με βιβλιοθήκες

### (1) Περιγραφή:

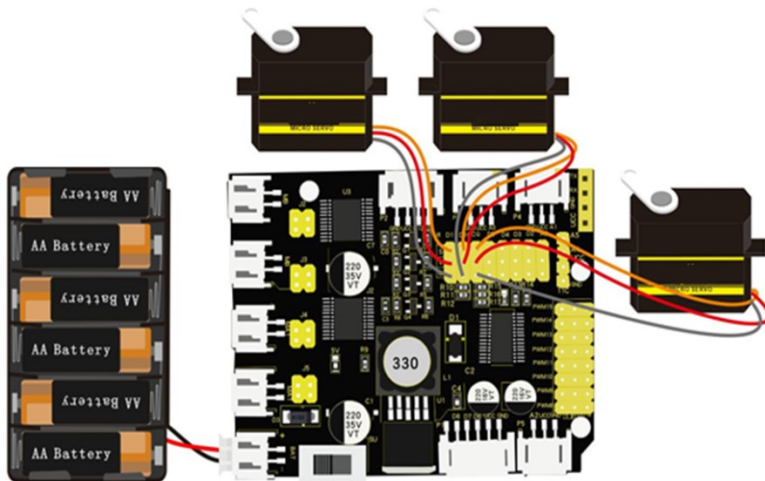
Σε αυτή τη δραστηριότητα, σε συνέχεια της προηγούμενης, θα προγραμματίσετε τον έλεγχο του σερβοκινητήρα 1 εντάσσοντας τις βιβλιοθήκες στον κώδικα προγραμματισμού.

### (2) Τυχριάζεστε:

Ελεγκτής	Πλακέτα οδήγησης κινητήρων	Σερβοκινητήρας	Βάση μπαταρίας	Καλώδιο USB
				

### (3) Διάγραμμα καλωδίωσης:

Σε αυτή τη δραστηριότητα θα ελέγξετε τον σερβοκινητήρα 1.



**Σημείωση καλωδίωσης:** η καφέ γραμμή του σερβοκινητήρα συνδέεται με το Gnd(G), η κόκκινη γραμμή συνδέεται με το V και η πορτοκαλί με το S.

Συνδέστε την πλακέτα οδήγησης στον υπολογιστή σας με το καλώδιο USB.

Ο σερβοκινητήρας πρέπει να συνδεθεί με εξωτερική τροφοδοσία λόγω της μεγάλης απαίτησής του σε ρεύμα. Το ρεύμα που παίρνει από τη σύνδεση του καλωδίου USB με τον υπολογιστή σας δεν είναι αρκετό για να τον κινήσει.

**(4) Κώδικας δοκιμής:**

```

/*
 Δραστηριότητα 1.6.1
 Προγραμματισμός Ελέγχου Σερβοκινητήρα δαγκάνας (Σερβοκινητήρας 1) με βιβλιοθήκες
*/
#include <Servo.h> // Βιβλιοθήκη για τη χρήση των σερβοκινητήρων
Servo myservo; // δημιουργία αντικείμενου σερβοκινητήρα για τον έλεγχο ενός σερβοκινητήρα
// Δώδεκα αντικείμενα σερβοκινητήρα μπορούν να δημιουργηθούν στις περισσότερες πλακέτες
int pos = 0; // // μεταβλητή για την αποθήκευση της θέσης του σερβοκινητήρα

void setup() {
  myservo.attach(11); // συνδέει τον σερβοκινητήρα στον ακροδέκτη 11 με το αντικείμενο του
  σερβοκινητήρα
}

void loop() {
  for (pos = 0; pos <= 180; pos += 1) { // από 0 μοίρες έως 180 μοίρες
    // σε βήματα του 1 βαθμού
    myservo.write(pos); // λέει στον σερβοκινητήρα να πάει στη θέση που αναφέρεται στη μεταβλητή
    'pos'
    delay(15); // περιμένει 15ms για να φτάσει ο σερβοκινητήρας στη θέση
  }
  for (pos = 180; pos >= 0; pos -= 1) { // από 180 μοίρες έως 0 μοίρες
    myservo.write(pos); // λέει στον σερβοκινητήρα να πάει στη θέση που αναφέρεται
    στη μεταβλητή 'pos'
    delay(15); // περιμένει 15ms για να φτάσει ο σερβοκινητήρας στη θέση
  }
}

```

**Εναλλακτικά:** Μέσα στον φάκελο «ΚΩΔΙΚΕΣ ARDUINO» μπορείτε να ανοίξετε το αρχείο κάνοντας διπλό κλικ πάνω στο αρχείο  M\_1.6.1 .

**(5) Αποτέλεσμα δοκιμής:**

Ανεβάστε σωστά τον κώδικα χωρίς σφάλματα. Ο σερβοκινητήρας κινείται εμπρός και πίσω στην περιοχή από 0° έως 180°. Εάν έχετε συνδεδεμένο τον σερβοκινητήρα πάνω στο ρομπότ τότε θα δείτε και τη δαγκάνα του βραχίονα να ανοιγοκλείνει.

## Δραστηριότητα 1.7: Προγραμματισμός αισθητήρα υπέρηχων

### (1) Περιγραφή



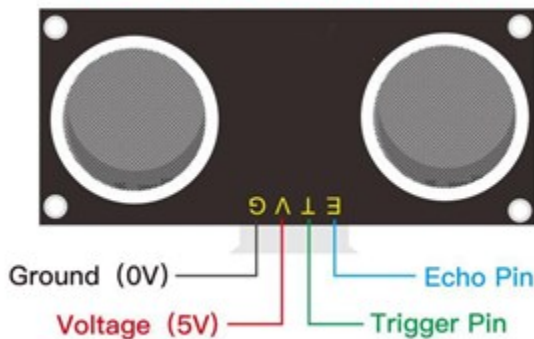
Ο αισθητήρας υπέρηχων χρησιμοποιεί σόναρ για να προσδιορίσει την απόσταση από ένα αντικείμενο, όπως λειτουργεί και η όραση μιας νυχτερίδας.

Προσφέρει εξαιρετική ανίχνευση εύρους χωρίς επαφή, με υψηλή ακρίβεια και σταθερές. Έρχεται πλήρης με μονάδες πομπού και δέκτη υπέρηχων. Ο αισθητήρας υπέρηχων χρησιμοποιείται σε ένα ευρύ φάσμα ηλεκτρονικών έργων για τη δημιουργία εφαρμογής ανίχνευσης εμποδίων και μέτρησης απόστασης καθώς και σε διάφορες άλλες εφαρμογές. Εδώ φέραμε την απλή μέθοδο μέτρησης της απόστασης με το Arduino και τον αισθητήρα υπέρηχων και τον τρόπο χρήσης του αισθητήρα υπέρηχων με το Arduino.

Σε αυτή τη δραστηριότητα θα προγραμματίσετε τον αισθητήρα υπέρηχων να σας ενημερώνει για την απόσταση από ένα αντικείμενο.

### (2) Προδιαγραφές:

#### Ultrasonic Sensor Pinout



Τροφοδοτικό: +5VDC

Ρεύμα ηρεμίας: <2 mA

Ρεύμα λειτουργίας: 15 mA

Γωνία επίδρασης: <math><15^\circ</math>








Απόσταση εμβέλειας: 2 cm – 400 cm

Ανάλυση: 0,3 cm Μέτρηση

Γωνία: Είσοδος σκανδάλης 30 μοιρών

Πλάτος παλμού: 10μS

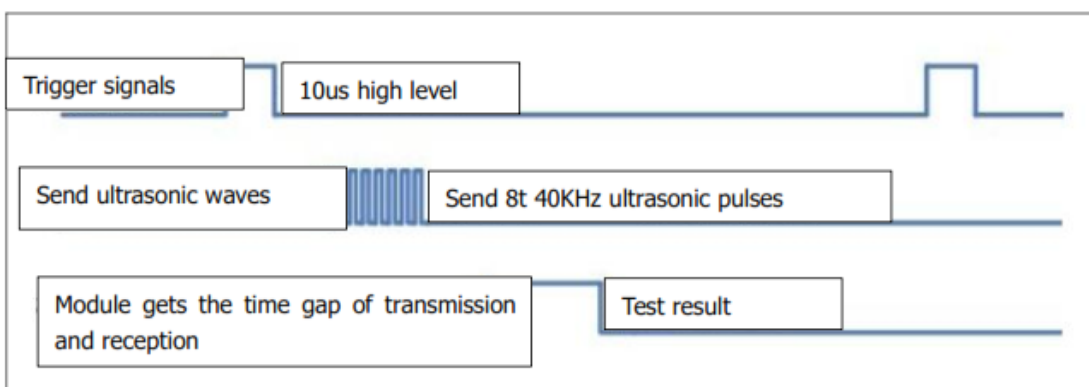
**(3) Τι χρειάζεστε:**

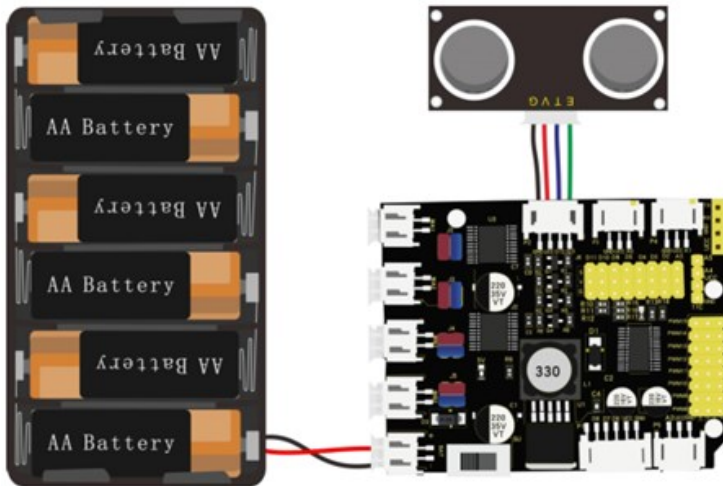
Ελεγκτής	Πλακέτα οδήγησης κινητήρων	Βάση μπαταρίας	Αισθητήρας Υπερήχων
			
Καλώδιο USB	(2x) 3pin F-F Dupont Line	4pin F-F Dupont Line	
			

**(4) Αρχή του αισθητήρα υπερήχων:**

Όπως μπορείτε να δείτε ο αισθητήρας υπερήχων είναι σαν δύο μάτια. Το ένα είναι το άκρο μετάδοσης, το άλλο είναι το άκρο λήψης. Η μονάδα υπερήχων εκπέμπει υπερηχητικά κύματα μετά το σήμα ενεργοποίησης. Όταν τα υπερηχητικά κύματα συναντούν το αντικείμενο και αντανακλώνται προς τα πίσω η μονάδα εξάγει ένα σήμα ήχου, ώστε να μπορεί να προσδιορίσει την απόσταση του αντικειμένου από τη χρονική διαφορά μεταξύ του σήματος ενεργοποίησης και του σήματος ήχου. Το  $t$  είναι ο χρόνος που το σήμα εκπέμπει όταν συναντά το εμπόδιο και επιστρέφει και η ταχύτητα διάδοσης του ήχου στον αέρα είναι περίπου 343 m/s, επομένως, απόσταση = ταχύτητα x χρόνος. Επειδή το υπερηχητικό κύμα εκπέμπει και επανέρχεται, δηλαδή 2 φορές απόσταση, οπότε πρέπει να διαιρεθεί με το 2, η απόσταση μετρημένη με υπερηχητικό κύμα =  $\frac{\text{ταχύτητα} \times \text{χρόνος}}{2}$

1. Χρησιμοποιήστε την παραπάνω μέθοδο και το διάγραμμα χρονισμού της μονάδας υπερήχων: Ρύθμιση του χρόνου καθυστέρησης του ακροδέκτη Trig του SR04 στα 10μs τουλάχιστον, που μπορεί να τον ενεργοποιήσει για ανίχνευση απόστασης.
2. Μετά την ενεργοποίηση, η μονάδα θα στείλει αυτόματα οκτώ παλμούς υπερήχων 40KHz και θα ανιχνεύσει εάν υπάρχει επιστροφή σήματος. Αυτό το βήμα θα ολοκληρωθεί αυτόματα από τη μονάδα.
3. Εάν το σήμα επιστρέψει, ο ακροδέκτης Echo θα εξάγει υψηλό επίπεδο και η διάρκεια του υψηλού επιπέδου είναι ο χρόνος από τη μετάδοση του υπερηχητικού κύματος έως την επιστροφή.



**(5) Διάγραμμα καλωδίωσης:**

Οδηγός καλωδίωσης:

Αισθητήρας υπερήχων

V →

T →

Echo →

G →

Ασπίδα αισθητήρα  
(V)

12(S)

13(S)

G(G)

Συνδέστε την πλακέτα οδήγησης στον υπολογιστή σας με το καλώδιο USB.

**(6) Κώδικας δοκιμής:**

/\*

Δραστηριότητα 1.7

Προγραμματισμός αισθητήρα υπερήχων

\*/

`int trigPin = 12;` // Trigger

`int echoPin = 13;` // Echo

`long duration, cm, inches;`

`void setup() {`

//Έναρξη σειριακής θύρας

`Serial.begin(9600);`

//Ορίστε εισόδους και εξόδους

`pinMode(trigPin, OUTPUT);`

`pinMode(echoPin, INPUT);`

`}`

`void loop() {`

// Ο αισθητήρας ενεργοποιείται από υψηλό παλμό (HIGH) 10 ή περισσότερων μικροδευτερόλεπτων (ms).

// Δώστε έναν σύντομο χαμηλό παλμό (LOW) εκ των προτέρων για να εξασφαλίσετε έναν καθαρό υψηλό παλμό (HIGH):

`digitalWrite(trigPin, LOW);`

`delayMicroseconds(2);`

`digitalWrite(trigPin, HIGH);`

`delayMicroseconds(10);`


`digitalWrite(trigPin, LOW);`

// Διαβάστε το σήμα από τον αισθητήρα: ένας υψηλός παλμός (HIGH) του οποίου



```
// η διάρκεια είναι ο χρόνος (σε μικροδευτερόλεπτα) από την αποστολή
// του ping στη λήψη της ηχώ του (Echo) από ένα αντικείμενο.
duration = pulseIn(echoPin, HIGH);
// Μετατρέψτε το χρόνο σε απόσταση.
cm = (duration/2) / 29.1; // Διαιρέστε με 29.1 ή πολλαπλασιάστε με 0.034
inches = (duration/2) / 74; // Διαιρέστε με 74 ή πολλαπλασιάστε με 0.0135
Serial.print(inches);
Serial.print("in, ");
Serial.print(cm);
Serial.print("cm");
Serial.println();
delay(50);
} //*****
```

**Εναλλακτικά:** Μέσα στον φάκελο «ΚΩΔΙΚΕΣ ARDUINO» μπορείτε να ανοίξετε το αρχείο κάνοντας

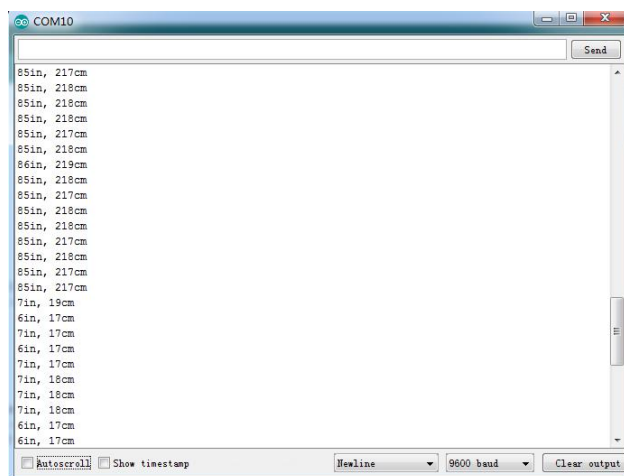
διπλό κλικ πάνω στο αρχείο  M\_1.7 .

### (7) Αποτέλεσμα δοκιμής:

Ανεβάστε σωστά τον κώδικα χωρίς σφάλματα. Ανοίξτε τη σειριακή οθόνη και ρυθμίστε τον ρυθμό baud στο 9600.

Θα εμφανιστεί η απόσταση που ανιχνεύτηκε, η μονάδα είναι cm και ίντσες.

Εμποδίστε τον αισθητήρα υπερήχων με το χέρι, η εμφανιζόμενη τιμή απόστασης γίνεται μικρότερη.



### (8) Επεξήγηση κώδικα

**inttrigPin**- Αυτή η ακίδα έχει οριστεί για να μεταδίδει υπερηχητικά κύματα, γενικά την έξοδο.

**intechoPin** - Αυτό ορίζεται ως η ακίδα λήψης, γενικά η είσοδος

$cm = (duration/2) / 29.1$ -unit is cm  $inches = (duration/2) / 74$ -unit is inch

Μπορούμε να υπολογίσουμε την απόσταση χρησιμοποιώντας τον ακόλουθο τύπο:

$distance=(traveltime/2) \times \text{ταχύτητα ήχου}$

Η ταχύτητα του ήχου είναι:  $343m/s = 0.0343 \text{ cm}/\mu s$

Ή σε ίντσες:  $13503.9in/s$

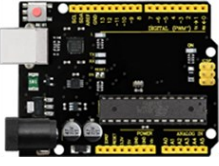

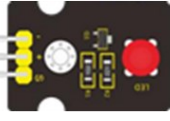





Πρέπει να διαιρέσουμε τον χρόνο ταξιδιού με το 2 καθώς πρέπει να λάβουμε υπόψη ότι το κύμα στάλθηκε, χτύπησε το αντικείμενο και μετά επέστρεψε στον αισθητήρα.

## Δραστηριότητα 1.8: Προγραμματισμός αισθητήρα υπερήχων και ενεργοποίηση LED

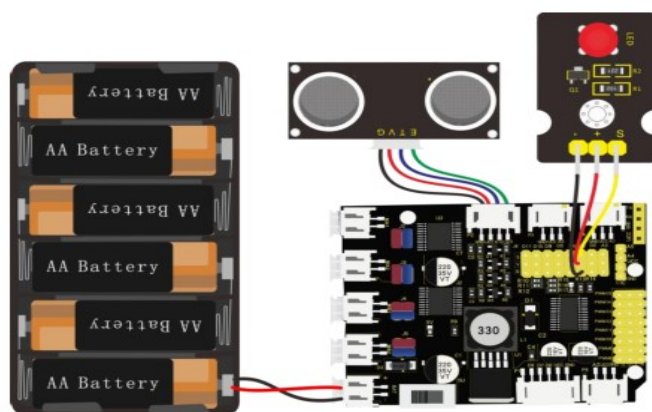
### (1) Περιγραφή

Σε αυτή τη δραστηριότητα, σε συνέχεια της προηγούμενης, θα προγραμματίσετε τον αισθητήρα υπερήχων να σας ενημερώνει για την απόσταση από ένα αντικείμενο ανάβοντας το κόκκινο LED όταν το αντικείμενο βρίσκεται σε απόσταση από 10 έως 2cm.

### (2) Τι χρειάζεστε:

Ελεγκτής	Πλακέτα οδήγησης κινητήρων	Κόκκινο Led	Βάση μπαταρίας
			
Καλώδιο USB	(2x) 3pin F-F Dupont Line	Αισθητήρας Υπερήχων	4pin F-F Dupont Line
			

### (3) Διάγραμμα καλωδίωσης:



Οδηγός καλωδίωσης:

Αισθητήρας υπερήχων

V →

T →

Echo →

Ασπίδα αισθητήρα

(V)

12(S)

13(S)

G → G(G)

Οι ακροδέκτες -, + και S της μονάδας LED συνδέονται στα G, V και D3 της θωράκισης. Συνδέστε την πλακέτα οδήγησης στον υπολογιστή σας με το καλώδιο USB.

#### (4) Κώδικας δοκιμής:

```

/*
/*
Δραστηριότητα 1.8
Προγραμματισμός αισθητήρα υπερήχων και ενεργοποίηση LED
*/
int trigPin = 12; // Trigger
int echoPin = 13; // Echo
long duration, cm, inches;
void setup() {
  Serial.begin(9600); // Έναρξη σειριακής θύρας
  pinMode(trigPin, OUTPUT); // Ορίστε εισόδους και εξόδους
  pinMode(echoPin, INPUT);
  pinMode(3, OUTPUT);
}
void loop() {
  // Ο αισθητήρας ενεργοποιείται από υψηλό παλμό (HIGH) των 10 ή περισσότερων
  μικροδευτερόλεπτων.
  // Δώστε έναν σύντομο χαμηλό παλμό (LOW) εκ των προτέρων για να εξασφαλίσετε έναν καθαρό
  υψηλό παλμό (HIGH):
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  // Διαβάστε το σήμα από τον αισθητήρα: ένας υψηλός παλμός (HIGH) του οποίου
  // η διάρκεια είναι ο χρόνος (σε μικροδευτερόλεπτα) από την αποστολή
  // του ping στη λήψη της ηχώ του από ένα αντικείμενο.
  duration = pulseIn(echoPin, HIGH);
  // Μετατρέψτε το χρόνο σε απόσταση
  cm = (duration/2) / 29.1; // Διαιρέστε με 29.1 ή πολλαπλασιάστε με 0.0343
  inches = (duration/2) / 74; // Διαιρέστε με 74 ή πολλαπλασιάστε με 0.0135
  Serial.print(inches);
  Serial.print("in, ");
  Serial.print(cm);
  Serial.print("cm");
  Serial.println();
  delay(50);
  if (cm>=2 && cm<=10)digitalWrite(3, HIGH);
  else digitalWrite(3, LOW);
}/******

```

**Εναλλακτικά:** Μέσα στον φάκελο «ΚΩΔΙΚΕΣ ARDUINO» μπορείτε να ανοίξετε το αρχείο κάνοντας διπλό κλικ πάνω στο αρχείο  M\_1.8 .

**(5) Αποτέλεσμα δοκιμής:**


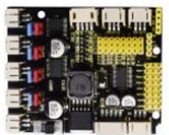
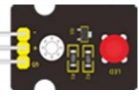







Ανεβάστε σωστά τον κώδικα χωρίς σφάλματα. Βάλτε το χέρι σας μακριά από τον αισθητήρα υπερήχων σε απόσταση από 2 cm έως 10 cm και δείτε το LED να ανάβει.

## Δραστηριότητα 1.9: Προγραμματισμός αισθητήρα υπερήχων και ενεργοποίηση LED και βομβητή

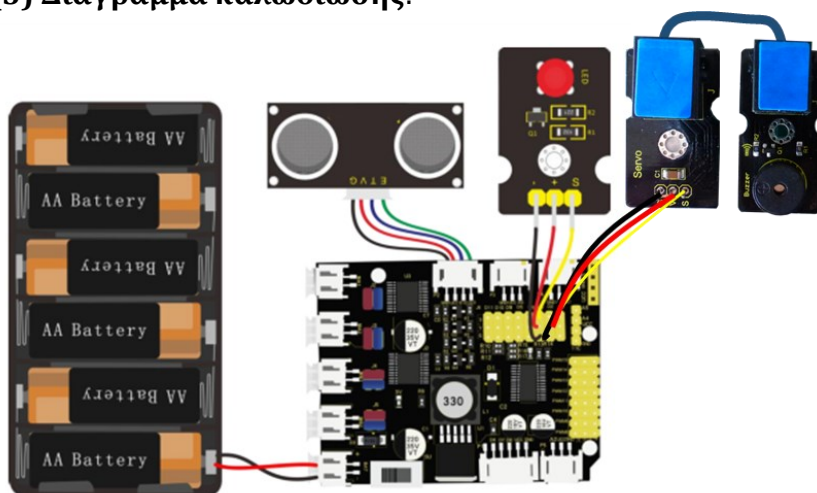
### (1) Περιγραφή

Σε αυτή τη δραστηριότητα όπως και στη προηγούμενη θα προγραμματίσετε τον αισθητήρα υπερήχων να σας ενημερώνει για την απόσταση από ένα αντικείμενο όχι μόνο ανάβοντας το κόκκινο LED όταν το αντικείμενο βρίσκεται σε απόσταση από 1 έως 2cm, αλλά επίσης και τον βομβητή ο οποίος θα σας ενημερώνει ηχητικά για την απόσταση αυξάνοντας τη συχνότητα που ακούγεται όσο το αντικείμενο πλησιάζει. Όσο πιο κοντά, τόσο πιο γρήγορα θα χτυπάει ο βομβητής.

### (2) Τι χρειάζεστε:

Ελεγκτής	Πλακέτα οδήγησης κινητήρων	Κόκκινο Led	Βάση μπαταρίας	Βομβητής με καλώδιο UTP
				
Καλώδιο USB	(2x) 3pin F-F Dupont Line	Αισθητήρας Υπερήχων	4pin F-F Dupont Line	Επέκταση RJ11
				

### (3) Διάγραμμα καλωδίωσης:



Οδηγός καλωδίωσης:

Αισθητήρας υπερήχων		Ασπίδα αισθητήρα
V	→	(V)
T	→	12(S)
Echo	→	13(S)
G	→	G(G)

Οι ακροδέκτες -, + και S της μονάδας LED συνδέονται στα G, V και D3 της πλακέτας οδήγησης.

Συνδέστε την επέκταση RJ 11 στα G, V και D2 της πλακέτας οδήγησης και συνδέστε την επέκταση με τον βομβητή χρησιμοποιώντας το καλώδιο UTP.

Συνδέστε την πλακέτα οδήγησης στον υπολογιστή σας με το καλώδιο USB.

#### (4) Κώδικας δοκιμής:

```

/*
 Δραστηριότητα 1.9
 Προγραμματισμός αισθητήρα υπερήχων και ενεργοποίηση LED και Βομβητή
 */
int trigPin = 12; // Trigger
int echoPin = 13; // Echo
long duration, cm, inches;
int led = 3; // ορίζουμε το LED στον ακροδέκτη 3
int buzzer=2; // ορίζουμε τον ψηφιακό ακροδέκτη (IO) του βομβητή

void setup() {
  Serial.begin(9600); // Έναρξη σειριακής θύρας
  pinMode(trigPin, OUTPUT); // Ορίστε εισόδους και εξόδους
  pinMode(echoPin, INPUT);
  pinMode(led, OUTPUT);
  pinMode(buzzer, OUTPUT); // ορίζουμε τον ψηφιακό ακροδέκτη IO, OUTPUT σε έξοδο.
}
void loop()
{
  // Ο αισθητήρας ενεργοποιείται από υψηλό παλμό (HIGH) των 10 ή περισσότερων
  μικροδευτερόλεπτων.
  // Δώστε έναν σύντομο χαμηλό παλμό (LOW) εκ των προτέρων για να εξασφαλίσετε έναν καθαρό
  υψηλό παλμό (HIGH):
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  // Διαβάστε το σήμα από τον αισθητήρα: ένας υψηλός παλμός (HIGH) του οποίου
  // η διάρκεια είναι ο χρόνος (σε μικροδευτερόλεπτα) από την αποστολή
  // του ping στη λήψη της ηχώ του από ένα αντικείμενο.

  duration = pulseIn(echoPin, HIGH);
  // Μετατρέψτε το χρόνο σε απόσταση
  cm = (duration/2) / 29.1; // Διαιρέστε με 29.1 ή πολλαπλασιάστε με 0.0343
  inches = (duration/2) / 74; // Διαιρέστε με 74 ή πολλαπλασιάστε με 0.0135

```



## Δραστηριότητα 1.10: Προγραμματισμός αισθητήρα παρακολούθησης γραμμής

### (1) Περιγραφή:

Ο αισθητήρας παρακολούθησης είναι στην πραγματικότητα ένας υπέρυθρος αισθητήρας. Η αρχή λειτουργίας του αισθητήρα είναι να χρησιμοποιεί τη διαφορετική ανακλαστικότητα του υπέρυθρου φωτός στο χρώμα και στη συνέχεια να μετατρέπει την ισχύ του ανακλώμενου σήματος σε σήμα ρεύματος.

Κατά τη διαδικασία ανίχνευσης, το μαύρο είναι ενεργό σε ΥΨΗΛΟ επίπεδο, αλλά το λευκό είναι ενεργό σε επίπεδο ΧΑΜΗΛΟ. Το ύψος ανίχνευσης είναι 0-3 cm. Η μονάδα παρακολούθησης γραμμής 3 καναλιών, έχει ενσωματώσει 3 σετ σωλήνων υπέρυθρων TCRT5000 σε μία μόνο πλακέτα

Περιστρέφοντας το ρυθμιζόμενο ποτενσιόμετρο στον αισθητήρα, μπορεί να ρυθμίσει την ευαισθησία ανίχνευσης του αισθητήρα.

Στη δραστηριότητα αυτή θα προγραμματίσετε τον αισθητήρα παρακολούθησης γραμμής να σας ενημερώνει στην σειριακή οθόνη για τις τιμές που λαμβάνει.




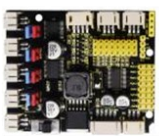
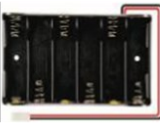
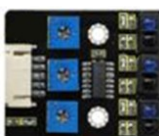



### (2) Προδιαγραφή:

- Τάση λειτουργίας: 3,3-5V (DC)
- Διεπαφή: 5 PIN
- Σήμα εξόδου: Ψηφιακό σήμα
- Ύψος ανίχνευσης: 0-3 cm

**Ειδική σημείωση:** πριν από τη δοκιμή, γυρίστε το ποτενσιόμετρο του αισθητήρα για να ρυθμίσετε την ευαισθησία ανίχνευσης.

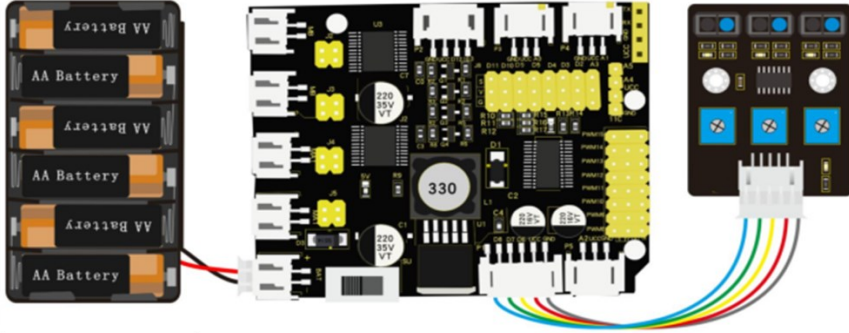
Όταν ρυθμίζετε το LED στο όριο μεταξύ ON και OFF, η ευαισθησία είναι η καλύτερη.

### (3) Τι χρειάζεστε:

Ελεγκτής	Πλακέτα οδήγησης κινητήρων	Βάση μπαταρίας	Αισθητήρας παρακολούθησης γραμμής	Καλώδιο USB	(2x) 3pin F-F Dupont Line	4pin F-F Dupont Line
						



## (4) Διάγραμμα καλωδίωσης:



Τα καλώδια G και V του αισθητήρα παρακολούθησης γραμμής συνδέονται με τα G και V της πλακέτας οδήγησης, οι υπόλοιπες ακίδες συνδέονται με D6, D7, D8 αντίστοιχα. Συνδέστε την πλακέτα οδήγησης στον υπολογιστή σας με το καλώδιο USB.

## (5) Κώδικας δοκιμής:

/\*

Δραστηριότητα 1.10

Αισθητήρας παρακολούθησης γραμμής


\*/

```
#define SensorLeft 6 // καρφίτσα εισαγωγής του αριστερού αισθητήρα
#define SensorMiddle 7 // καρφίτσα εισαγωγής του μεσαίου αισθητήρα
#define SensorRight 8 // καρφίτσα εισαγωγής του δεξιού αισθητήρα
```

```
unsigned char SL; // κατάσταση αριστερού αισθητήρα
unsigned char SM; // κατάσταση μεσαίου αισθητήρα
unsigned char SR; // κατάσταση δεξιού αισθητήρα
```

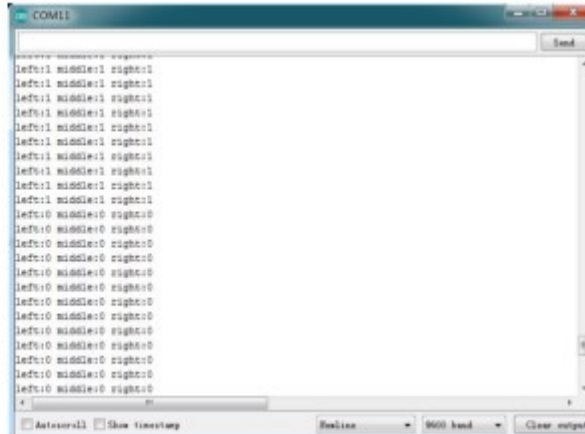
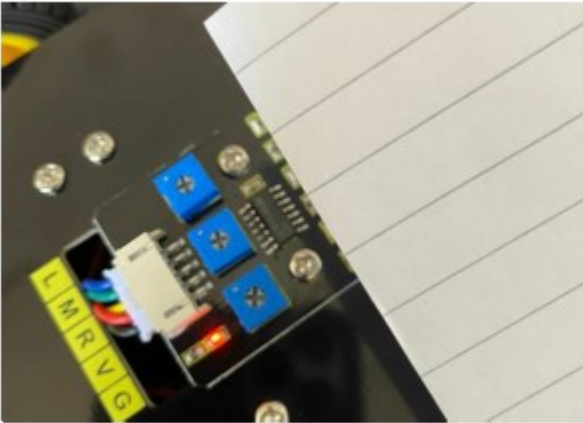
```
void setup(){
  // Ορίζουμε τον ρυθμό baud σε 9600
  Serial.begin(9600);
  // ρύθμιση σε λειτουργία εισαγωγής
  pinMode(SensorLeft, INPUT);
  pinMode(SensorMiddle, INPUT);
  pinMode(SensorRight, INPUT);
}
```

```
void loop(){
  // διαβάστε τις τιμές τριών αισθητήρων παρακολούθησης γραμμής
  SL = digitalRead(SensorLeft);
  SM = digitalRead(SensorMiddle);
  SR = digitalRead(SensorRight);
  // εκτυπώστε τις τιμές τριών αισθητήρων
  Serial.print("SL=");
  Serial.print(SL);
  Serial.print("SM=");
  Serial.print(SM);
  Serial.print("SR=");
  Serial.println(SR);
}
```

**Εναλλακτικά:** Μέσα στον φάκελο «ΚΩΔΙΚΕΣ ARDUINO» μπορείτε να ανοίξετε το αρχείο κάνοντας διπλό κλικ πάνω στο αρχείο  M\_8.10 .

### (6) Αποτέλεσμα δοκιμής:

Ανεβάστε σωστά τον κώδικα χωρίς σφάλματα. Ανοίξτε τη σειριακή οθόνη για να δείτε τα σήματα που λαμβάνει ο αισθητήρας παρακολούθησης γραμμής. Η εμφανιζόμενη τιμή είναι 1 (υψηλό επίπεδο) όταν δεν λαμβάνονται σήματα. Η τιμή γίνεται 0 όταν καλύπτετε τον αισθητήρα με χαρτί (λαμβάνει σήμα).



### (7) Επεξήγηση κώδικα:

**Serial.begin(9600)**- Αρχικοποίηση σειριακής θύρας, ορισμός ρυθμού baud σε 9600.

**pinMode**- Ορίστε την ακίδα ως λειτουργία εισόδου ή εξόδου.


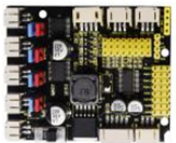




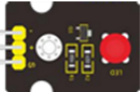

**digitalRead**- Διαβάστε την κατάσταση του pin, που είναι γενικά ΥΨΗΛΟ και ΧΑΜΗΛΟ επίπεδο.

## Δραστηριότητα 1.11: Προγραμματισμός αισθητήρα παρακολούθησης γραμμής και οπτική ειδοποίηση

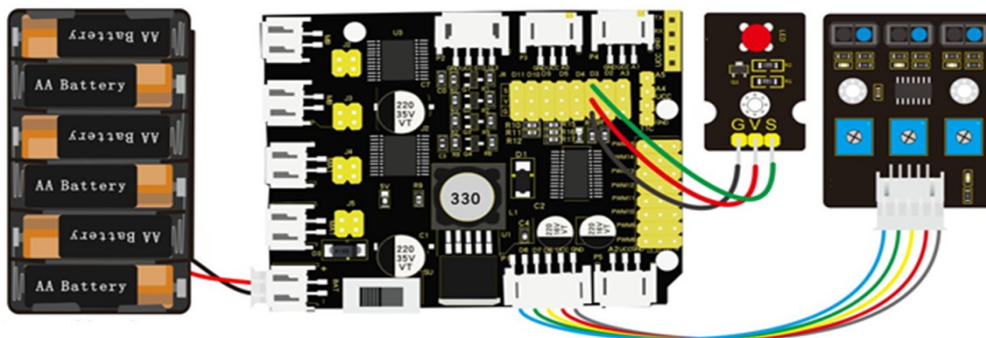
### (1) Περιγραφή:

Σε αυτή τη δραστηριότητα σε συνέχεια της προηγούμενης θα προγραμματίσετε τον αισθητήρα να σας ενημερώνει όταν χάνει τη γραμμή ανάβοντας το κόκκινο LED.

### (2) Τι χρειάζεστε:

Ελεγκτής	Πλακέτα οδήγησης κινητήρων	Βάση μπαταρίας	Αισθητήρας παρακολούθησης γραμμής
			
Καλώδιο USB	(2x) 3pin F-F Dupont Line	Κόκκινο Led	5pin F-F Dupont Line
			

### (3) Διάγραμμα καλωδίωσης:



Οι ακροδέκτες -, + και S της μονάδας LED συνδέονται στα G, V και D3 της πλακέτας οδήγησης. Συνδέστε την πλακέτα οδήγησης στον υπολογιστή σας με το καλώδιο USB.

## (4) Κώδικας δοκιμής:

```

/*
 Δραστηριότητα 1.11
 Αισθητήρας παρακολούθησης γραμμής και LED
 */
int L_pin = 6; // ακροδέκτες του αριστερού αισθητήρα παρακολούθησης γραμμής
int M_pin = 7; // ακροδέκτες του μεσαίου αισθητήρα παρακολούθησης γραμμής
int R_pin = 8; // ακροδέκτες του δεξιού αισθητήρα παρακολούθησης γραμμής
int val_L, val_R, val_M; // ορίστε τις μεταβλητές τριών αισθητήρων

void setup()
{
  Serial.begin(9600); // αρχικοποιήστε την σειριακή επικοινωνία σε 9600 bit ανά δευτερόλεπτο
  pinMode(L_pin, INPUT); // ορίστε τον ακροδέκτη L_pin ως είσοδο
  pinMode(M_pin, INPUT); // ορίστε τον ακροδέκτη M_pin ως είσοδο
  pinMode(R_pin, INPUT); // ορίστε τον ακροδέκτη R_pin ως είσοδο
  pinMode(3, OUTPUT);
}
void loop()
{
  val_L = digitalRead(L_pin); // ανάγνωση του L_pin
  val_R = digitalRead(R_pin); // ανάγνωση του M_pin
  val_M = digitalRead(M_pin); // ανάγνωση του R_pin

  Serial.print("left:");
  Serial.print(val_L);
  Serial.print(" middle:");
  Serial.print(val_M);
  Serial.print(" right:");
  Serial.println(val_R);


  if (val_L == HIGH) // εάν ο αριστερός αισθητήρας παρακολούθησης γραμμής ανιχνεύσει σήματα
  {
    digitalWrite(3, LOW); // LED σβήνει
  }
  else // εάν ο αριστερός αισθητήρας παρακολούθησης γραμμής δεν ανιχνεύει σήματα
  {
    digitalWrite(3, HIGH); // LED ανάβει
    delay(2000);
  }
  if (val_M == HIGH) // εάν ο μεσαίος αισθητήρας παρακολούθησης γραμμής ανιχνεύει σήματα
  {
    digitalWrite(3, LOW); // LED σβήνει
  }
  else // εάν ο μεσαίος αισθητήρας παρακολούθησης γραμμής δεν ανιχνεύει σήματα
  {
    digitalWrite(3, HIGH); // LED ανάβει
    delay(2000);
  }
}

```

```

if (val_R == HIGH) // εάν ο δεξιός αισθητήρας παρακολούθησης γραμμής ανιχνεύει σήματα
{
  digitalWrite(3, LOW); // LED σβήνει
}
else // εάν ο δεξιός αισθητήρας παρακολούθησης γραμμής δεν ανιχνεύει σήματα
{
  digitalWrite(3, HIGH); // LED ανάβει
  delay(2000);
}
}
//*****

```

**Εναλλακτικά:** Μέσα στον φάκελο «ΚΩΔΙΚΕΣ ARDUINO» μπορείτε να ανοίξετε το αρχείο κάνοντας διπλό κλικ πάνω στο αρχείο  M\_1.11 .

### (5) Αποτέλεσμα δοκιμής:

Ανεβάστε σωστά τον κώδικα χωρίς σφάλματα. Παρατηρείστε ότι το LED ανάβει όταν καλύπτετε τον αισθητήρα παρακολούθησης γραμμής με το χέρι.

Σχηματίστε μία μαύρη γραμμή με διάφορες στροφές αριστερά και δεξιά. Κινήσετε τον αισθητήρα με το χέρι σας κατά μήκος αυτής της γραμμής και παρατηρείστε το LED που ανάβει.

**Σημείωση 1:** Το πάχος της μαύρης γραμμής πρέπει να είναι τόσο ώστε να καλύπτει και τους 3 σωλήνες υπέρυθρων.



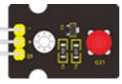







**Σημείωση 2:** Το ρομπότ δεν θα αρχίσει να κινείται πάνω στη γραμμή γιατί ακόμα δεν έχουμε προγραμματίσει την κίνηση των κινητήρων. Αυτό γίνεται από την δραστηριότητα 1.21 και κάτω. Οπότε σε αυτές τις δραστηριότητες θα πρέπει να κινείτε με το χέρι σας το ρομπότ πάνω στη γραμμή.

## Δραστηριότητα 1.12: Προγραμματισμός αισθητήρα παρακολούθησης γραμμής και οπτική ειδοποίηση (2 LED)

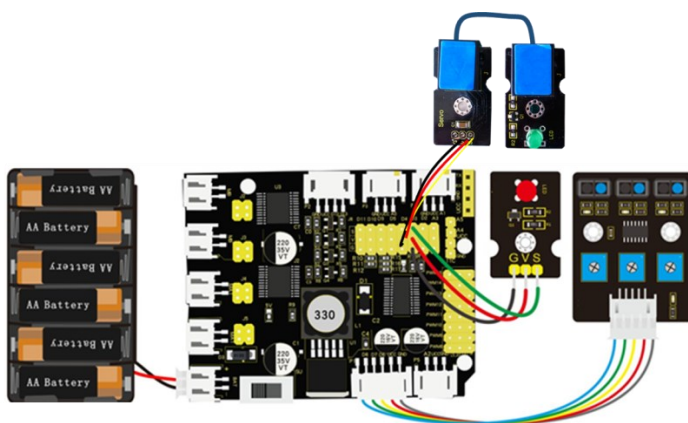
### (1) Περιγραφή:

Σε αυτή τη δραστηριότητα, σε συνέχεια της προηγούμενης, θα προσθέσετε ένα ακόμα LED, πράσινο LED. Θα προγραμματίσετε τον αισθητήρα παρακολούθησης γραμμής έτσι ώστε να σας ενημερώνει πότε το ρομπότ στρίβει δεξιά και πότε αριστερά. Πιο συγκεκριμένα όταν στρίβει δεξιά θα ανάβει το πράσινο LED, ενώ όταν στρίβει αριστερά το κόκκινο LED.

### (2) Τι χρειάζεστε:

Ελεγκτής	Πλακέτα οδήγησης κινητήρων	Κόκκινο Led	Βάση μπαταρίας	Πράσινο LED
				
Καλώδιο USB	(2x) 3pin F-F Dupont Line	Αισθητήρας Υπερήχων	4pin F-F Dupont Line	Επέκταση RJ11
				

### (3) Διάγραμμα καλωδίωσης:



Οι ακροδέκτες -, + και S της μονάδας LED συνδέονται στα G, V και D3 της πλακέτας οδήγησης. Συνδέστε την επέκταση RJ 11 στα G, V και D4 της πλακέτας οδήγησης και συνδέστε την επέκταση με το πράσινο LED χρησιμοποιώντας το καλώδιο UTP. Συνδέστε την πλακέτα οδήγησης στον υπολογιστή σας με το καλώδιο USB.

**(4) Κώδικας δοκιμής:**

```

/*
 Δραστηριότητα 1.12
 Αισθητήρας παρακολούθησης γραμμής και 2 LED
*/
int L_pin = 6; // ακροδέκτης του αριστερού αισθητήρα παρακολούθησης γραμμής
int M_pin = 7; // ακροδέκτης του μεσαίου αισθητήρα παρακολούθησης γραμμής
int R_pin = 8; // ακροδέκτης του δεξιού αισθητήρα παρακολούθησης γραμμής
int val_L, val_R, val_M; // μεταβλητές τριών αισθητήρων
int Red_Led = 3, Green_Led=4; // μεταβλητές των LED
void setup()
{
  Serial.begin(9600); // αρχικοποιήστε την σειριακή επικοινωνία σε 9600 bit ανά δευτερόλεπτο
  pinMode(L_pin, INPUT); // ορίστε τον ακροδέκτη L_pin ως είσοδο
  pinMode(M_pin, INPUT); // ορίστε τον ακροδέκτη M_pin ως είσοδο
  pinMode(R_pin, INPUT); // ορίστε τον ακροδέκτη R_pin ως είσοδο
  pinMode(Red_Led, OUTPUT); // Κόκκινος LED ακροδέκτης
  pinMode(Green_Led, OUTPUT); // Πράσινος LED ακροδέκτης
}
void loop()
{
  val_L = digitalRead(L_pin); // ανάγνωση του L_pin
  val_M = digitalRead(M_pin); // ανάγνωση του M_pin
  val_R = digitalRead(R_pin); // ανάγνωση του R_pin
  Serial.print("left:");
  Serial.print(val_L);
  Serial.print(" middle:");
  Serial.print(val_M);
  Serial.print(" right:");
  Serial.println(val_R);

  if (val_L == HIGH) // εάν ο αριστερός αισθητήρας παρακολούθησης γραμμής ανιχνεύσει σήμα
  {
    digitalWrite(Green_Led, LOW); // Πράσινο LED σβήνει
  }
  else // εάν ο αριστερός αισθητήρας παρακολούθησης γραμμής δεν ανιχνεύει σήμα
  {
    digitalWrite(Green_Led, HIGH); // Πράσινο LED ανάβει
    delay(500);
  }


  if (val_R == HIGH) // εάν ο δεξιός αισθητήρας παρακολούθησης γραμμής ανιχνεύσει σήμα
  {
    digitalWrite(Red_Led, LOW); // Κόκκινο LED σβήνει
  }
  else // εάν ο δεξιός αισθητήρας παρακολούθησης γραμμής δεν ανιχνεύσει σήμα
  {
    digitalWrite(Red_Led, HIGH); // Κόκκινο LED ανάβει
    delay(500);
  }
}

```

```

if (val_M == HIGH) // ο μεσαίος αισθητήρας δεν μας αφορά για την παρόν δραστηριότητα.
{
  return;
}
} //*****

```

**Εναλλακτικά:** Μέσα στον φάκελο «ΚΩΔΙΚΕΣ ARDUINO» μπορείτε να ανοίξετε το αρχείο κάνοντας διπλό κλικ πάνω στο αρχείο  M\_1.12 .

#### (5) Αποτέλεσμα δοκιμής:

Ανεβάστε σωστά τον κώδικα χωρίς σφάλματα. Σχηματίστε μία μαύρη γραμμή με διάφορες στροφές αριστερά και δεξιά. Κινήσετε τον αισθητήρα με το χέρι σας κατά μήκος αυτής της γραμμής και παρατηρείστε τα LED που ανάβουν.

**Σημείωση 1:** Το πάχος της μαύρης γραμμής πρέπει να είναι τόσο ώστε να καλύπτει και τους 3 σωλήνες υπέρυθρων.

**Σημείωση 2:** Το ρομπότ δεν θα αρχίσει να κινείται πάνω στη γραμμή γιατί ακόμα δεν έχουμε προγραμματίσει την κίνηση των κινητήρων. Αυτό γίνεται από την δραστηριότητα 1.21 και κάτω. Οπότε σε αυτές τις δραστηριότητες θα πρέπει να κινείτε με το χέρι σας το ρομπότ πάνω στη γραμμή.


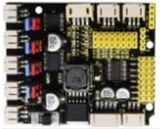









## Δραστηριότητα 1.13: Προγραμματισμός αισθητήρα παρακολούθησης γραμμής και ηχητική ειδοποίηση

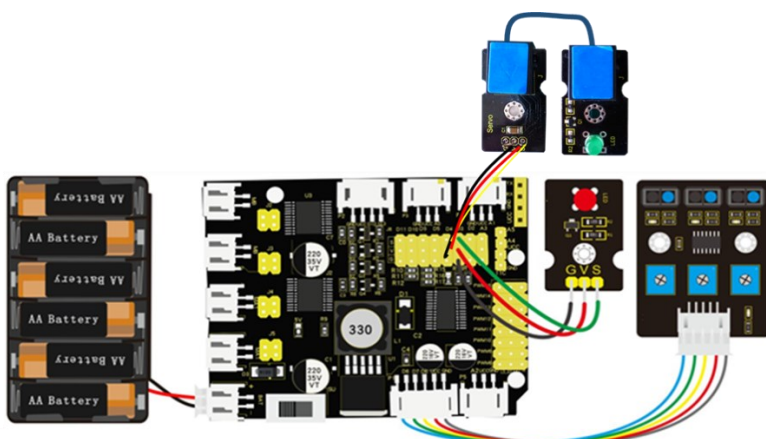
### (1) Περιγραφή:

Σε αυτή τη δραστηριότητα, σε συνέχεια των προηγούμενων, θα προγραμματίσετε τον αισθητήρα παρακολούθησης να σας ενημερώνει για τη πορεία του μέσω του βομβητή. Πιο συγκεκριμένα όταν κινείται ευθεία θα σας ενημερώνει με ήχο που θα αναβοσβήνει με χαμηλή συχνότητα, ενώ όταν στρίβει θα αυξάνεται για λίγο η συχνότητα του ήχου.

### (2) Τι χρειάζεστε:

Ελεγκτής	Πλακέτα οδήγησης κινητήρων	Πράσινο LED	Βάση μπαταρίας	Επέκταση RJ11
				
Καλώδιο USB	(2x) 3pin F-F Dupont Line	Αισθητήρας Υπερήχων	4pin F-F Dupont Line	
				

### (3) Διάγραμμα καλωδίωσης:



Οι ακροδέκτες -, + και S της μονάδας LED συνδέονται στα G, V και D3 της πλακέτας οδήγησης. Συνδέστε την επέκταση RJ 11 στα G, V και D4 της πλακέτας οδήγησης και συνδέστε την επέκταση με το πράσινο LED χρησιμοποιώντας το καλώδιο UTP. Συνδέστε τη πλακέτα οδήγησης στον υπολογιστή σας με το καλώδιο USB.

## (4) Κώδικας δοκιμής:

```

/*
 Δραστηριότητα 1.13
 Αισθητήρας παρακολούθησης γραμμής και Βομβητής
*/
int L_pin = 6; // ακροδέκτης του αριστερού αισθητήρα παρακολούθησης γραμμής
int M_pin = 7; // ακροδέκτης του μεσαίου αισθητήρα παρακολούθησης γραμμής
int R_pin = 8; // ακροδέκτης του δεξιού αισθητήρα παρακολούθησης γραμμής
int val_L, val_R, val_M; // μεταβλητές τριών αισθητήρων
int Red_Led = 3, Green_Led=4; // μεταβλητές των LED
int buzzer=2;
void setup()
{
  Serial.begin(9600); // αρχικοποιήστε την σειριακή επικοινωνία σε 9600 bit ανά δευτερόλεπτο
  pinMode(L_pin, INPUT); // ορίστε τον ακροδέκτη L_pin ως είσοδο
  pinMode(M_pin, INPUT); // ορίστε τον ακροδέκτη M_pin ως είσοδο
  pinMode(R_pin, INPUT); // ορίστε τον ακροδέκτη R_pin ως είσοδο
  pinMode(Red_Led, OUTPUT); // Κόκκινος LED ακροδέκτης
  pinMode(Green_Led, OUTPUT); // Πράσινος LED ακροδέκτης
}
void loop()
{
  val_L = digitalRead(L_pin); // ανάγνωση του L_pin
  val_R = digitalRead(R_pin); // ανάγνωση του M_pin
  val_M = digitalRead(M_pin); // ανάγνωση του R_pin
  Serial.print("left:");
  Serial.print(val_L);
  Serial.print("middle:");
  Serial.print(val_M);
  Serial.print("right:");
  Serial.println(val_R);

  if (val_L == HIGH) // εάν ο αριστερός αισθητήρας παρακολούθησης γραμμής ανιχνεύσει σήμα
  {
    digitalWrite(Green_Led, LOW); // Πράσινο LED σβήνει
  }
  Else // εάν ο αριστερός αισθητήρας παρακολούθησης γραμμής δεν ανιχνεύει σήμα
  {
    for(int i=0;i<80;i++) // παραγωγή ήχου συχνότητας
    { digitalWrite(buzzer, HIGH); // παράγει ήχο
      delay(1); //delay1ms
      digitalWrite(buzzer, LOW); // δεν παράγει ήχο
      delay(1); //ms delay
    }
    for(int i=0;i<100;i++) // παραγωγή ήχου συχνότητας
    {
      digitalWrite(buzzer, HIGH); // παράγει ήχο
      digitalWrite(buzzer, LOW); // δεν παράγει ήχο
      delay(1); //2ms delay
    }
  }
}

```


```

digitalWrite(Green_Led, HIGH); // Πράσινο LED ανάβει
// delay(500);
}

if (val_R == HIGH) // εάν ο δεξιός αισθητήρας παρακολούθησης γραμμής ανιχνεύσει σήμα
{
digitalWrite(Red_Led, LOW); // Κόκκινο LED σβήνει
}
Else // εάν ο δεξιός αισθητήρας παρακολούθησης γραμμής δεν ανιχνεύσει σήμα
{
for(int i=0;i<80;i++) // παραγωγή ήχου συχνότητας
{ digitalWrite(buzzer, HIGH); // παράγει ήχο
delay(1); // delay 1ms
digitalWrite(buzzer, LOW); // δεν παράγει ήχο
delay(1); // ms delay
}
for(int i=0;i<100;i++) // παραγωγή ήχου συχνότητας
{
digitalWrite(buzzer, HIGH); // παράγει ήχο
digitalWrite(buzzer, LOW); // δεν παράγει ήχο
delay(1); // 2ms delay
}
digitalWrite(Red_Led, HIGH); // Κόκκινο LED ανάβει
// delay(100);
}

if (val_M == HIGH) // αν ο αισθητήρας παρακολούθησης της μεσαίας γραμμής δεν ανιχνεύει σήματα
{
//digitalWrite(3, LOW); // LED is off
for(int i=0;i<80;i++)
{
digitalWrite(buzzer, HIGH); // παράγει ήχο
delay(1); // delay 1ms
digitalWrite(buzzer, LOW); // δεν παράγει ήχο
delay(1); // ms delay
}
for(int i=0;i<100;i++) // παραγωγή ήχου συχνότητας
{
digitalWrite(buzzer, HIGH); // παράγει ήχο
digitalWrite(buzzer, LOW); // δεν παράγει ήχο
delay(2); // 2ms delay
}
}
else // εάν ο αισθητήρας παρακολούθησης γραμμής δεν ανιχνεύει σήμα
{
//digitalWrite(3, HIGH); // LED is on
//delay(500);
}
}
//*****

```

**Εναλλακτικά:** Μέσα στον φάκελο «ΚΩΔΙΚΕΣ ARDUINO» μπορείτε να ανοίξετε το αρχείο κάνοντας διπλό κλικ πάνω στο αρχείο  M\_1.13 .

**(5) Αποτέλεσμα δοκιμής:**

Ανεβάστε σωστά τον κώδικα χωρίς σφάλματα. Σχηματίσετε μία μαύρη γραμμή με διάφορες στροφές αριστερά και δεξιά και κινήστε τον αισθητήρα με το χέρι σας κατά το μήκος αυτής της γραμμής. Στρίψτε αριστερά και δεξιά και παρατηρήστε τα LED που ανάβουν και τον βομβητή.

**Σημείωση 1:** Το πάχος της μαύρης γραμμής πρέπει να είναι τόσο ώστε να καλύπτει και τους 3 σωλήνες υπέρυθρων.

**Σημείωση 2:** Το ρομπότ δεν θα αρχίσει να κινείται πάνω στη γραμμή γιατί ακόμα δεν έχουμε προγραμματίσει την κίνηση των κινητήρων. Αυτό γίνεται από την δραστηριότητα 1.21 και κάτω. Οπότε σε αυτές τις δραστηριότητες θα πρέπει να κινείται το ρομπότ με το χέρι σας πάνω στη γραμμή.

## Δραστηριότητα 1.14: Προγραμματισμός τηλεχειριστηρίου IR

### (1) Περιγραφή:

Δεν υπάρχει αμφιβολία ότι το τηλεχειριστήριο υπέρυθρων είναι πανταχού παρόν στην καθημερινή ζωή.

Χρησιμοποιείται για τον έλεγχο διαφόρων οικιακών συσκευών, όπως τηλεοράσεις, στερεοφωνικά, συσκευές εγγραφής βίντεο και δέκτες δορυφορικού σήματος.

Το τηλεχειριστήριο IR αποτελείται από συστήματα εκπομπής υπέρυθρων και λήψης υπέρυθρων, διαθέτει ένα τηλεχειριστήριο IR, μια μονάδα λήψης υπέρυθρων και έναν μικροϋπολογιστή ενός τσιπ με δυνατότητα αποκωδικοποίησης. Το σήμα φορέα υπέρυθρων που εκπέμπεται από το τηλεχειριστήριο κωδικοποιείται από το τσιπ κωδικοποίησης στο τηλεχειριστήριο.



Αποτελείται από ένα τμήμα πιλοτικού κώδικα, κώδικα χρήστη, αντίστροφο κώδικα χρήστη, κώδικα δεδομένων και αντίστροφο κώδικα δεδομένων. Το χρονικό διάστημα του παλμού χρησιμοποιείται για να διακρίνει εάν είναι σήμα 0 ή 1 και η κωδικοποίηση αποτελείται από αυτά τα σήματα 0, 1. Ο κώδικας χρήστη του ίδιου τηλεχειριστηρίου παραμένει αμετάβλητος. Ο κώδικας δεδομένων μπορεί να διακρίνει το κλειδί. Όταν πατηθεί το κουμπί του τηλεχειριστηρίου, το τηλεχειριστήριο εκπέμπει ένα σήμα υπέρυθρου φορέα. Όταν ο δέκτης υπέρυθρων λάβει το σήμα, το πρόγραμμα θα αποκωδικοποιήσει το σήμα φορέα και θα καθορίσει ποιο πλήκτρο θα πατηθεί. Το MCU αποκωδικοποιεί το λαμβανόμενο σήμα 01, κρίνοντας έτσι ποιο πλήκτρο πατιέται από το τηλεχειριστήριο.

Ο δέκτης υπέρυθρων αποτελείται κυρίως από μια κεφαλή δέκτη υπέρυθρων, η οποία είναι συσκευή που ενσωματώνει τη λήψη, την ενίσχυση και την αποδιαμόρφωση του σήματος. Επιπλέον, είναι κατάλληλο για τηλεχειριστήριο IR και υπέρυθρη μετάδοση δεδομένων. Η μονάδα λήψης υπέρυθρων του δέκτη έχει μόνο τρεις γραμμές, σήμα, τάση VCC και γείωση GND.

Στη δραστηριότητα αυτή θα προγραμματίσετε ώστε το τηλεχειριστήριο να σας ενημερώνει για τα σήματα που στέλνει πατώντας κάθε κουμπί του μέσω εμφάνισης στη σειριακή οθόνη (serial monitor) του ARDUINO IDE.

### (2) Προδιαγραφή:

Τάση λειτουργίας: 3,3-5V (DC)

Διεπαφή: 3 PIN


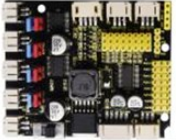





Σήμα εξόδου: Ψηφιακό σήμα

Γωνία λήψης: 90 μοίρες

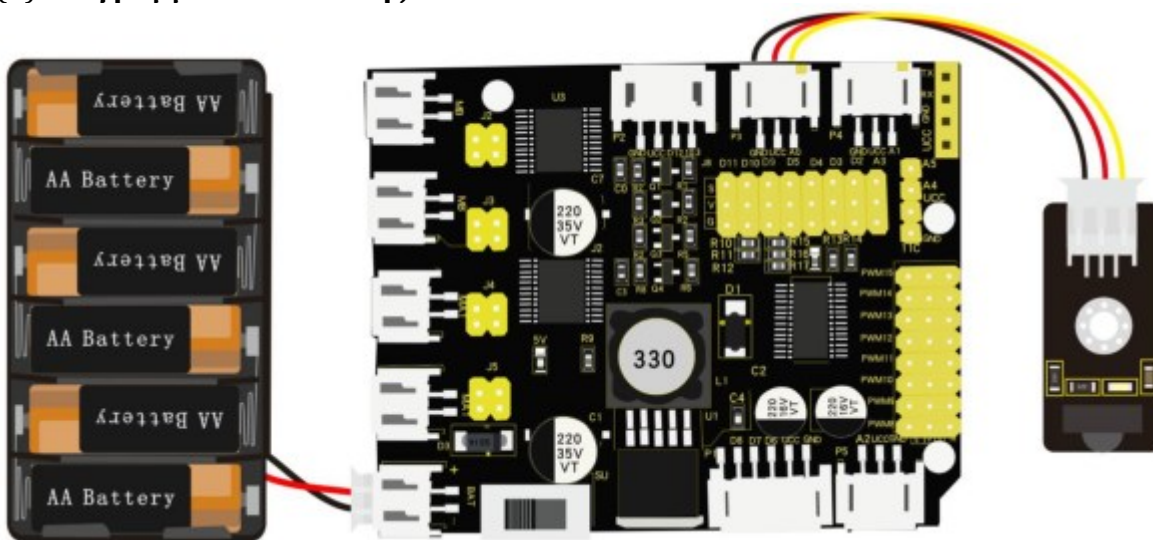
Συχνότητα: 38 kHz

Απόσταση παραλαβής: 10μ

(3) Τι χρειάζεστε:

Ελεγκτής	Πλακέτα οδήγησης κινητήρων	Βάση μπαταρίας	Δέκτης υπέρυθρων
			
Καλώδιο USB	(2x) 3pin F-F Dupont Line	3pin F-F Dupont Line	
			

(4) Διάγραμμα καλωδίωσης:



Αντίστοιχα συνδέστε τα «-», «+» και το S της μονάδας δέκτη υπέρυθρων με G (GND), V (VCC) και A0 της πλακέτας ανάπτυξης. Συνδέστε τη πλακέτα οδήγησης στον υπολογιστή σας με το καλώδιο USB.

(5) Κώδικας δοκιμής:

```

/*
 Δραστηριότητα 1.14
 Τηλεχειριστήριο IR
 */
#include <IRremote.h>
int RECV_PIN = A0; // Ο δέκτης IR είναι συνδεδεμένος στο A0


void setup(){
  Serial.begin(9600); // αρχικοποιήστε την σειριακή επικοινωνία σε 9600 bit ανά δευτερόλεπτο
  IrReceiver.begin(RECV_PIN); // Ενεργοποίηση δέκτη
  
```

```

}

void loop(){
  if (IrReceiver.decode()) {
    Serial.println(IrReceiver.decodedIRData.command); // Επιστρέφει έναν αριθμό που
    ανταποκρίνεται στο κουμπί του χειριστηρίου που πατήθηκε.
    IrReceiver.resume(); // Λαμβάνει την επόμενη τιμή
  }
}

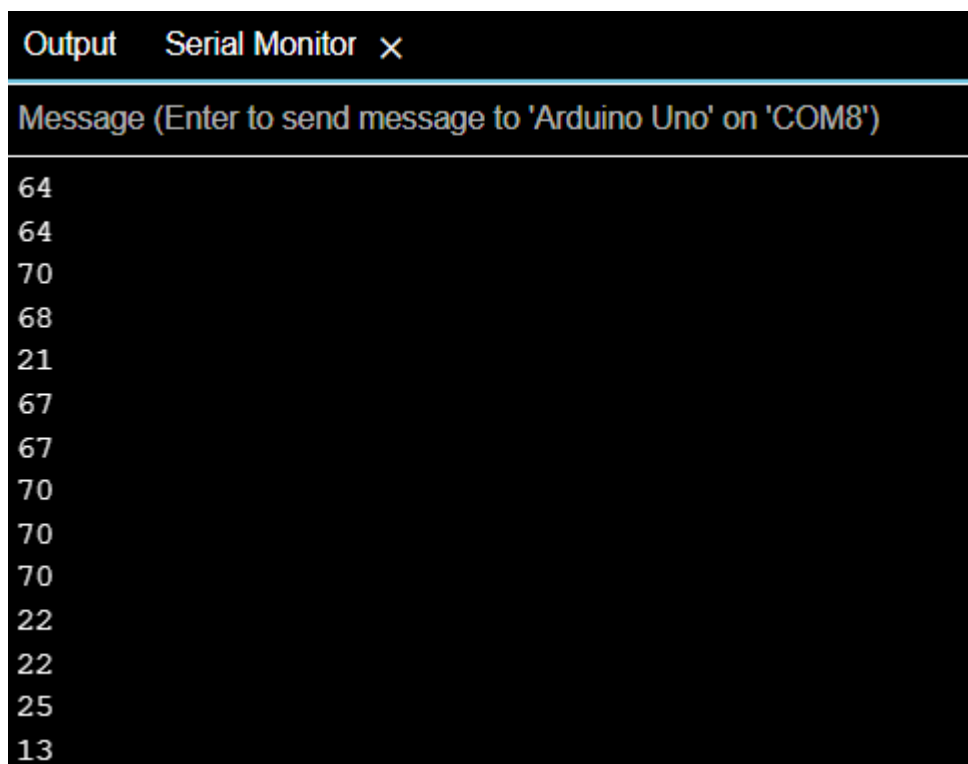
```

**Εναλλακτικά:** Μέσα στον φάκελο «ΚΩΔΙΚΕΣ ARDUINO» μπορείτε να ανοίξετε το αρχείο κάνοντας διπλό κλικ πάνω στο αρχείο  M\_1.14 .

**Προσοχή:** Με την προϋπόθεση ότι οι ψηφιακές θύρες δεν είναι διαθέσιμες, οι αναλογικές θύρες μπορούν να θεωρηθούν ως ψηφιακές θύρες. Το A0 ισούται με D14, το A1 είναι ισοδύναμο με το ψηφιακό 15.

#### (6) Αποτέλεσμα δοκιμής:

Ανεβάστε σωστά τον κώδικα χωρίς σφάλματα. Ανοίξτε τη σειριακή οθόνη και ρυθμίστε το ρυθμό baud στο 9600, σημαδέψτε το τηλεχειριστήριο στον δέκτη υπέρυθρων και θα εμφανιστεί η αντίστοιχη τιμή.

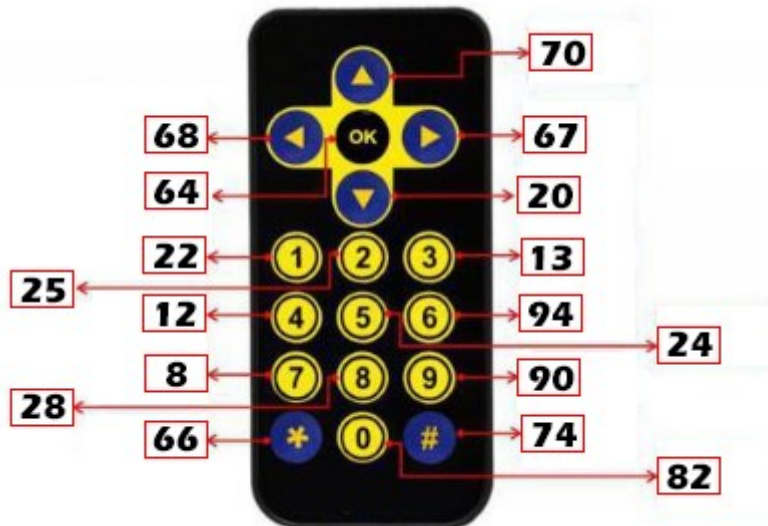


```

Output  Serial Monitor x
Message (Enter to send message to 'Arduino Uno' on 'COM8')
64
64
70
68
21
67
67
70
70
70
22
22
25
13

```

Παρακάτω παραθέτουμε τη τιμή κάθε κουμπιού του τηλεχειριστηρίου. Έτσι μπορείτε να το κρατήσετε ως αναφορά.



(7) Επεξήγηση κώδικα:

**IrReceiver.begin(RECV\_PIN):** Αρχικοποιεί τον δέκτη IR με τον καθορισμένο ακροδέκτη για να αρχίσει να λαμβάνει υπέρυθρα σήματα.

**IrReceiver.decode():** Ελέγχει εάν υπάρχει εισερχόμενο σήμα υπέρυθρων (IR) που έχει ληφθεί από τον δέκτη IR.


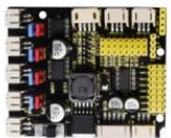




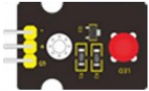



## Δραστηριότητα 1.15: Προγραμματισμός τηλεχειριστηρίου IR και οπτική ειδοποίηση

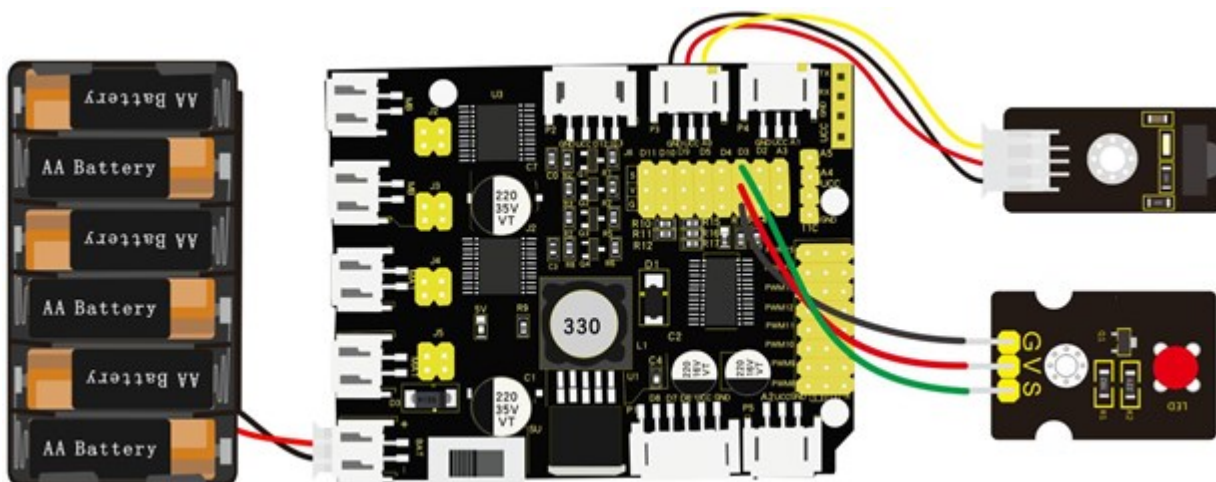
### (1) Περιγραφή:

Σε αυτή τη δραστηριότητα, σε συνέχεια της προηγούμενης, θα προγραμματίσετε το τηλεχειριστήριο να ανάβει το κόκκινο LED όταν πατάτε το κουμπί ok και να σβήνει το κόκκινο LED όταν ξαναπατάτε το κουμπί ok.

### (2) Τι χρειάζεστε:

Ελεγκτής	Πλακέτα οδήγησης κινητήρων	Βάση μπαταρίας	Δέκτης υπερύθρων
			
Καλώδιο USB	(2x) 3pin F-F Dupont Line	Κόκκινο LED	3pin F-F Dupont Line
			

### (3) Διάγραμμα καλωδίωσης:



Συνδέστε τα «-», «+» και το S της μονάδας δέκτη υπέρυθρων με G (GND), V (VCC) και A0 της πλακέτας ανάπτυξης. Οι ακροδέκτες -, + και S της μονάδας LED συνδέονται στα G, V και D3 της πλακέτας οδήγησης.

Συνδέστε την πλακέτα οδήγησης στον υπολογιστή σας με το καλώδιο USB.

**(4) Κώδικας δοκιμής:**

```

/*
 Δραστηριότητα 1.15
 Τηλεχειριστήριο IR και LED
 */
#include <IRremote.h>
int RECV_PIN = A0; // Ο δέκτης IR είναι συνδεδεμένος στο A0
int LED_PIN= 3; // Ορίζει τον ακροδέκτη του LED ως τον ακροδέκτη 3 (D3).
int a=0; // Αρχικοποίηση της κατάστασης του LED, 0 = σβηστό

void setup() {
  Serial.begin(9600);
  IrReceiver.begin(RECV_PIN); // Αρχικοποίηση του δέκτη IR
  pinMode(LED_PIN, OUTPUT); // ορίζει τον ακροδέκτη 3 του LED σε OUTPUT
}

void loop() {
  if (IrReceiver.decode()) {
    if (IrReceiver.decodedIRData.command == 64 & a == 0) // σύμφωνα με την παραπάνω τιμή του
    πλήκτρου, πατήστε "OK" στο τηλεχειριστήριο, το LED θα ελεγχθεί.
    {
      digitalWrite(LED_PIN, HIGH); // Το LED θα ανάψει
      a = 1;
    }
    else if (IrReceiver.decodedIRData.command == 64 & a == 1) // αν πατηθεί το κουμπί "OK" ξανά
    {
      digitalWrite(LED_PIN, LOW); // Το LED θα σβήσει
      a = 0;
    }
    IrReceiver.resume(); // λαμβάνει την επόμενη τιμή
    delay(100);
  }
}
//*****

```

**Εναλλακτικά:** Μέσα στον φάκελο «ΚΩΔΙΚΕΣ ARDUINO» μπορείτε να ανοίξετε το αρχείο κάνοντας διπλό κλικ πάνω στο αρχείο  M\_1.15 .

**(5) Αποτέλεσμα δοκιμής:**

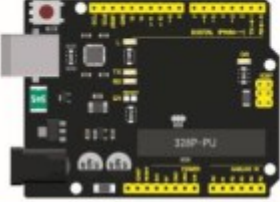
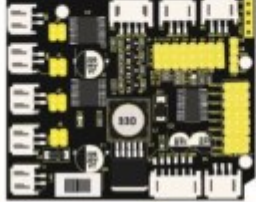






Ανεβάστε σωστά τον κώδικα χωρίς σφάλματα. Πατήστε το πλήκτρο "OK" στο τηλεχειριστήριο για να ενεργοποιήσετε και να απενεργοποιήσετε το LED.

## Δραστηριότητα 1.16: Προγραμματισμός τηλεχειριστηρίου IR και έλεγχος LED

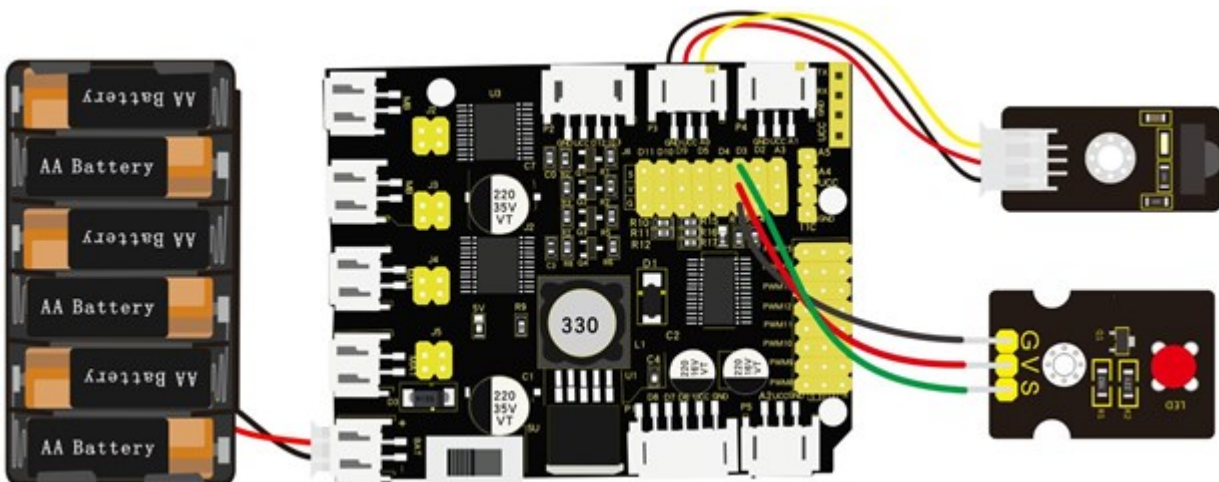
### (1) Περιγραφή:

Σε αυτή τη δραστηριότητα, σε συνέχεια της προηγούμενης, θα προγραμματίσετε το τηλεχειριστήριο να αναβοσβήνει το κόκκινο LED όταν πατάτε το κουμπί ok, να σβήνει το κόκκινο LED όταν ξαναπατάτε το κουμπί ok και επιπλέον να ρυθμίζει τη συχνότητα που αναβοσβήνει το LED, πιο γρήγορα με το πάνω βέλος και πιο αργά με το κάτω βέλος.

### (2) Τι χρειάζεστε:

Control Board *1	Motor Shield*1	Battery Holder *1	IR receiver module *1
			
USB Cable *1	3pin Dupont line *1	LED module *1	3pin Dupont line *1
			

### (3) Διάγραμμα καλωδίωσης:



Συνδέστε τα «-», «+» και το S της μονάδας δέκτη υπέρυθρων με G (GND), V (VCC) και A0 της πλακέτας ανάπτυξης. Οι ακροδέκτες -, + και S της μονάδας LED συνδέονται στα G, V και D3 της πλακέτας οδήγησης.

Συνδέστε την πλακέτα οδήγησης στον υπολογιστή σας με το καλώδιο USB.

**(4) Κώδικας δοκιμής:**

```

/*
 Δραστηριότητα 1.16
 Τηλεχειριστήριο IR και έλεγχος LED
 */
#include <IRremote.h>

int RECV_PIN = A0;
int LED_PIN = 3;
int ledState = LOW;
unsigned long previousMillis = 0;
const long interval = 100; // διάστημα για το αναβοσβήσιμο σε ms
bool speedMode = false; // μεταβλητή για να κρατάει την κατάσταση της λειτουργίας ταχύτητας
int brightness = 255; // αρχική φωτεινότητα του LED

void setup() {
  Serial.begin(9600);
  IrReceiver.begin(RECV_PIN);
  pinMode(LED_PIN, OUTPUT);
}

void loop() {
  if (IrReceiver.decode()) {
    if (IrReceiver.decodedIRData.command == 64) {
      if (ledState == LOW) {
        ledState = HIGH;
        brightness = 255;
      } else {
        ledState = LOW;
        brightness = 0;
        speedMode = false;
      }
      digitalWrite(LED_PIN, ledState);
      analogWrite(LED_PIN, brightness);
      delay(50);
    } else if (IrReceiver.decodedIRData.command == 70) { // Όταν πατηθεί το πάνω βελάκι στο
χειριστήριο ενεργοποιείται το speedMode.
      speedMode = true;
    } else if (IrReceiver.decodedIRData.command == 21) { // Όταν πατηθεί το κάτω βελάκι στο
χειριστήριο απενεργοποιείται το speedMode.
      speedMode = false;
    }
    IrReceiver.resume();
  }


  if (ledState == HIGH && !speedMode) {
    unsigned long currentMillis = millis();
    if (currentMillis - previousMillis >= interval) {
      previousMillis = currentMillis;
      if (brightness == 0) {

```

```

    brightness = 255;
  } else {
    brightness = 0;
  }
  analogWrite(LED_PIN, brightness);
}
} else if (ledState == HIGH && speedMode) {
  unsigned long currentMillis = millis();
  if (currentMillis - previousMillis >= interval / 2) {
    previousMillis = currentMillis;
    if (brightness == 0) {
      brightness = 255;
    } else {
      brightness = 0;
    }
    analogWrite(LED_PIN, brightness);
  }
}
}
}
}

```

**Εναλλακτικά:** Μέσα στον φάκελο «ΚΩΔΙΚΕΣ ARDUINO» μπορείτε να ανοίξετε το αρχείο κάνοντας διπλό κλικ πάνω στο αρχείο  M\_1.16 .

#### (5) Αποτέλεσμα δοκιμής:










Ανεβάστε σωστά τον κώδικα χωρίς σφάλματα. Πατήστε τα βέλη πάνω κάτω και παρατηρείστε τη φωτεινότητα του LED να αυξομειώνεται.

## Δραστηριότητα 1.17: Προγραμματισμός τηλεχειριστηρίου IR και συχνότητας ήχου βομβητή

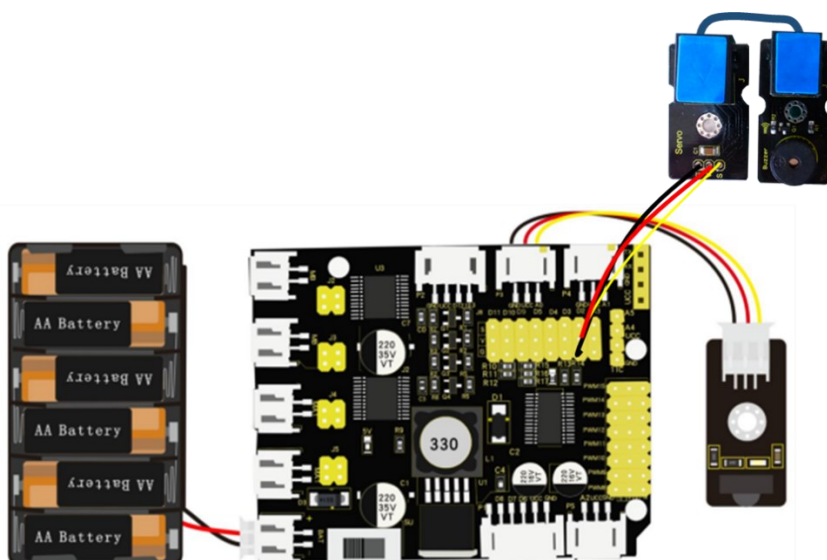
### (1) Περιγραφή:

Σε αυτή τη δραστηριότητα, σε συνέχεια της προηγούμενης, θα προγραμματίσετε το τηλεχειριστήριο να ρυθμίζει τη συχνότητα που ακούγεται ο ήχος του βομβητή χρησιμοποιώντας τα βέλη πάνω και κάτω. Επίσης με το κουμπί ok θα ανάβετε και θα σβήνετε το βομβητή.

### (2) Τυχριάζεστε:

Ελεγκτής	Πλακέτα οδήγησης κινητήρων	Βάση μπαταρίας	Δέκτης υπερώθρων	Βομβητής με καλώδιο UTP
				
Επέκταση RJ11	Καλώδιο USB	(2x) 3pin F-F Dupont Line	3pin F-F Dupont Line	
				

### (3) Διάγραμμα καλωδίωσης:



Συνδέστε τα «-», «+» και το S της μονάδας δέκτη υπέρυθρων με G (GND), V (VCC) και A0 της πλακέτας ανάπτυξης. Συνδέστε την επέκταση RJ 11 στα G, V και D2 της πλακέτας οδήγησης και συνδέστε την επέκταση με το βομβητή χρησιμοποιώντας το καλώδιο UTP. Συνδέστε την πλακέτα οδήγησης στον υπολογιστή σας με το καλώδιο USB.

**(4) Κώδικας δοκιμής:**

```

/*
 Δραστηριότητα 1.17
 Τηλεχειριστήριο IR και συχνότητα ήχου βομβητή
 */
#include <IRremote.h>

int RECV_PIN = A0;
int LED_PIN = 3;
int BUZZER_PIN = 2;
int ledState = LOW;
unsigned long previousMillis = 0;
const long interval = 100; // διάστημα για το αναβοσβήσιμο σε ms
bool speedMode = false; // μεταβλητή για να κρατάει την κατάσταση της λειτουργίας ταχύτητας
int brightness = 255; // αρχική φωτεινότητα του LED

void setup() {
  Serial.begin(9600);
  IrReceiver.begin(RECV_PIN);
  pinMode(LED_PIN, OUTPUT);
  pinMode(BUZZER_PIN, OUTPUT);
}

void loop() {
  if (IrReceiver.decode()) {
    if (IrReceiver.decodedIRData.command == 64) { // Όταν πατηθεί το "OK" στο χειριστήριο
      if (ledState == LOW) {
        ledState = HIGH;
        brightness = 255;
      } else {
        ledState = LOW;
        brightness = 0;
        speedMode = false;
      }
      digitalWrite(LED_PIN, ledState);
      analogWrite(LED_PIN, brightness);
      digitalWrite(BUZZER_PIN, LOW); // απενεργοποίηση βομβητή
      delay(50);
    } else if (IrReceiver.decodedIRData.command == 70) { // Όταν πατηθεί το πάνω βελάκι στο
χειριστήριο, ενεργοποιείται το speedMode
      speedMode = true;
    } else if (IrReceiver.decodedIRData.command == 21) { // Όταν πατηθεί το κάτω βελάκι στο
χειριστήριο, απενεργοποιείται το speedMode.
      speedMode = false;
    }
    IrReceiver.resume();
  }


  if (ledState == HIGH && !speedMode) {
    unsigned long currentMillis = millis();
  }
}

```

```

if (currentMillis - previousMillis >= interval) {
  previousMillis = currentMillis;
  if (brightness == 0) {
    brightness = 255;
    digitalWrite(BUZZER_PIN, HIGH); // ενεργοποίηση βομβητή
  } else {
    brightness = 0;
    digitalWrite(BUZZER_PIN, LOW); // απενεργοποίηση βομβητή
  }
  analogWrite(LED_PIN, brightness);
}
} else if (ledState == HIGH && speedMode) {
  unsigned long currentMillis = millis();
  if (currentMillis - previousMillis >= interval / 2) {
    previousMillis = currentMillis;
    if (brightness == 0) {
      brightness = 255;
      digitalWrite(BUZZER_PIN, HIGH); // ενεργοποίηση βομβητή
    } else {
      brightness = 0;
      digitalWrite(BUZZER_PIN, LOW); // απενεργοποίηση βομβητή
    }
    analogWrite(LED_PIN, brightness);
  }
} else {
  digitalWrite(BUZZER_PIN, LOW); // απενεργοποίηση βομβητή
}
}

```

**Εναλλακτικά:** Μέσα στον φάκελο «ΚΩΔΙΚΕΣ ARDUINO» μπορείτε να ανοίξετε το αρχείο κάνοντας διπλό κλικ πάνω στο αρχείο  M\_1.17 .

#### (5) Αποτέλεσμα δοκιμής:

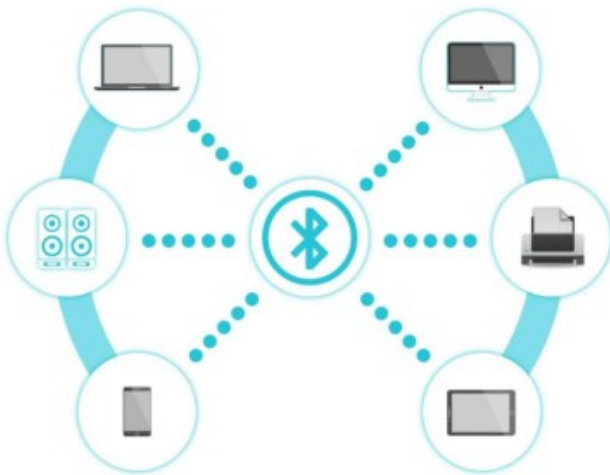
Ανεβάστε σωστά τον κώδικα χωρίς σφάλματα. Πατήστε τα βέλη πάνω κάτω και παρατηρείστε τη συχνότητά του ήχου του βομβητή να αυξομειώνεται.



## Δραστηριότητα 1.18: Προγραμματισμός τηλεχειριστηρίου Bluetooth

### (1) Περιγραφή:

Το Bluetooth, μια απλή μονάδα ασύρματης επικοινωνίας, η πιο δημοφιλής τις τελευταίες δεκαετίες και εύκολη στη χρήση, χρησιμοποιείται στις περισσότερες συσκευές που τροφοδοτούνται με μπαταρία.



Τα τελευταία χρόνια, πολλά πράγματα έχουν αλλάξει, όπως ο ρυθμός μετάδοσης δεδομένων, η κατανάλωση ενέργειας με φορητές συσκευές και συσκευές IoT και σύστημα ασφαλείας.

Εδώ θα μάθουμε για το HM-10 BLE 4.0 με το Arduino Board. Το HM-10 είναι μια εύκολα διαθέσιμη μονάδα Bluetooth 4.0.

Αυτή η μονάδα χρησιμοποιείται για τη δημιουργία ασύρματης επικοινωνίας δεδομένων.

Η μονάδα έχει σχεδιαστεί χρησιμοποιώντας το σύστημα Texas Instruments CC2540 ή CC2541 Bluetooth χαμηλής ενέργειας (BLE) σε Chip (SoC).

Στη δραστηριότητα αυτή θα προγραμματίσετε το τηλεχειριστήριο Bluetooth και σε επόμενες δραστηριότητες θα μάθετε να το προγραμματίζετε να εκτελεί διάφορες άλλες ενέργειες όπως την ενεργοποίηση ενός LED, βομβητή ακόμα και των σερβοκινητήρων αλλά και των κινητήρων οδήγησης.

### (2) Προδιαγραφή:

Πρωτόκολλο Bluetooth: Bluetooth



Χωρίς όριο byte στη σειριακή θύρα Transceiving

Συχνότητα εργασίας: Ζώνη ISM 2,4 GHz

Μέθοδος διαμόρφωσης: GFSK (Gaussian Frequency Shift Keying)

Ισχύς μετάδοσης: -23dBm, -6dBm, 0dBm, 6dBm, μπορεί να τροποποιηθεί με εντολή AT.

Ευαισθησία:  $\leq -84\text{dBm}$  σε 0,1% BER

Ρυθμός μετάδοσης: Ασύγχρονη: 6Kbyte ; Σύγχρονη: 6kByte

Χαρακτηριστικό ασφαλείας: Έλεγχος ταυτότητας και κρυπτογράφηση

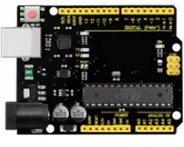





Υποστήριξη: Central & Peripheral UUID FFE0, FFE1

Κατανάλωση ενέργειας: Αυτόματη λειτουργία ύπνου, αναμονή ρεύματος 400uA~800uA, 8,5mA κατά τη μετάδοση.

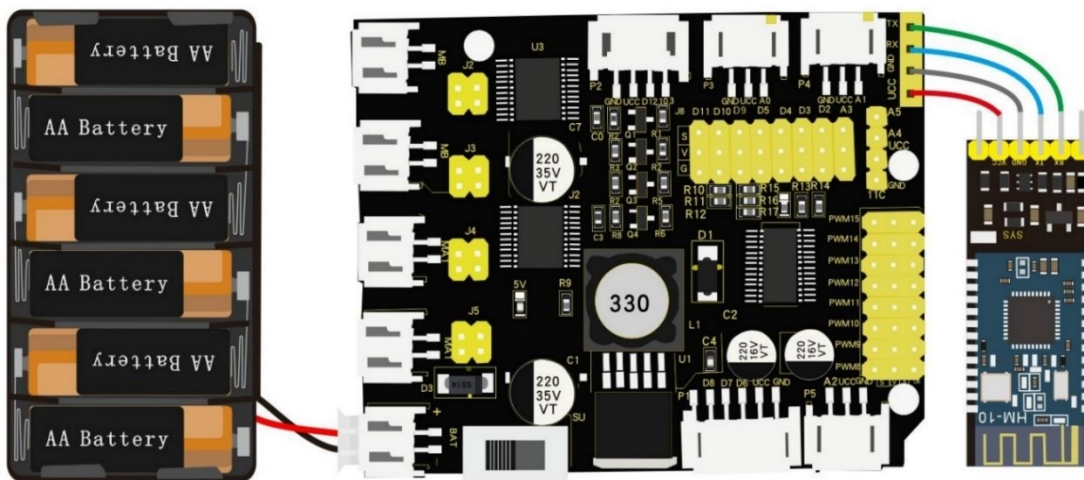
Τροφοδοσία: 5VDC

Θερμοκρασία λειτουργίας: -5 έως +65 Κελσίου

(3) Τι χρειάζεστε:

Ελεγκτής	Πλακέτα οδήγησης κινητήρων	Μονάδα Bluetooth	Βάση μπαταρίας	Καλώδιο USB	3pin F-F Dupont Line
					

(4) Διάγραμμα καλωδίωσης:



Τα VCC, GND, TXD και RXD της μονάδας Bluetooth συνδέονται με το VCC, το GND, TXD και το RXD της πλακέτας οδήγησης. Συνδέστε την πλακέτα οδήγησης στον υπολογιστή σας με το καλώδιο USB.

**ΣΗΜΕΙΩΣΗ:1.** Δώστε προσοχή στην κατεύθυνση των ακίδων κατά την εισαγωγή της μονάδας Bluetooth

**2. ΜΗΝ** εισάγετε τη μονάδα Bluetooth πριν ανεβάσετε τον κώδικα. (Για να αποφύγετε την αντίφαση μεταξύ σειριακής επικοινωνίας κώδικα και επικοινωνίας Bluetooth)

(5) Κώδικας δοκιμής:


```

/*
 Δραστηριότητα 1.18
 Bluetooth
*/
char blue_val; // χρησιμοποιείται για τη λήψη της τιμής του Bluetooth
void setup() {
  Serial.begin(9600); // ορίζει τον ρυθμό baud σε 9600
}

void loop() {
  if(Serial.available() > 0) // εάν τα σήματα Bluetooth λαμβάνονται
  {
    blue_val = Serial.read(); // λαμβάνει σήμα
    Serial.println(blue_val); // Η σειριακή θύρα εκτυπώνει την τιμή Bluetooth
  }
}

```

```
}
}
```

**Εναλλακτικά:** Μέσα στον φάκελο «ΚΩΔΙΚΕΣ ARDUINO» μπορείτε να ανοίξετε το αρχείο κάνοντας διπλό κλικ πάνω στο αρχείο  M\_1.18 .

Ανεβάστε σωστά τον κώδικα χωρίς σφάλματα και έπειτα τοποθετήστε τη μονάδα Bluetooth.

## (6) Λήψη APP

Ο Κώδικας είναι τα σήματα που λαμβάνονται από τη σειριακή θύρα, πρέπει όμως να στέλνουμε σήματα, εξου και η ανάγκη μιας εφαρμογής που στέλνει σήματα στη μονάδα Bluetooth για να τα εκτυπώσει στη σειριακή θύρα.

**Σημείωση:** Επιτρέψτε στην εφαρμογή να έχει πρόσβαση στην "τοποθεσία" στις ρυθμίσεις του κινητού σας τηλεφώνου όταν συνδέεστε σε μονάδα Bluetooth, διαφορετικά το Bluetooth ενδέχεται να μην είναι συνδεδεμένο.

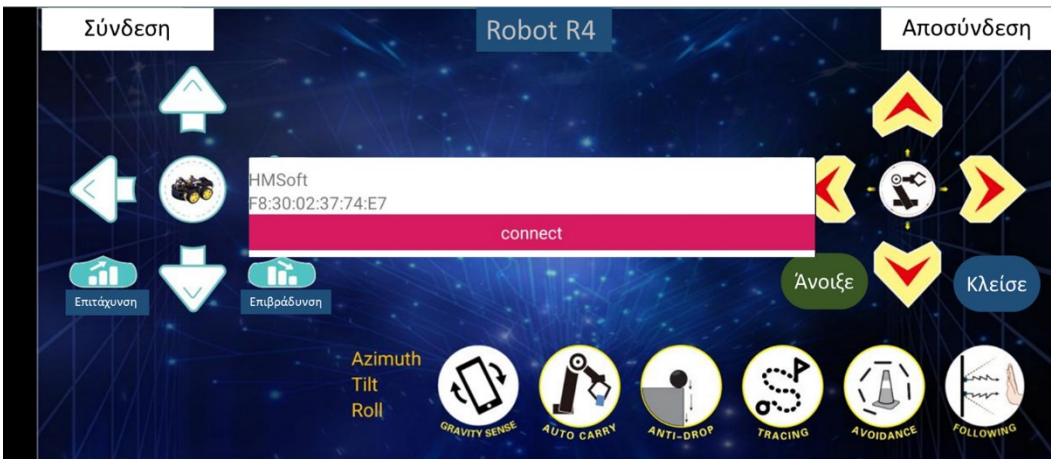
1. Τρέξτε την εφαρμογή "R4" επιλέγοντας πάνω στο εικονίδιο



Η σελίδα ελέγχου του ρομπότ φαίνεται παρακάτω:

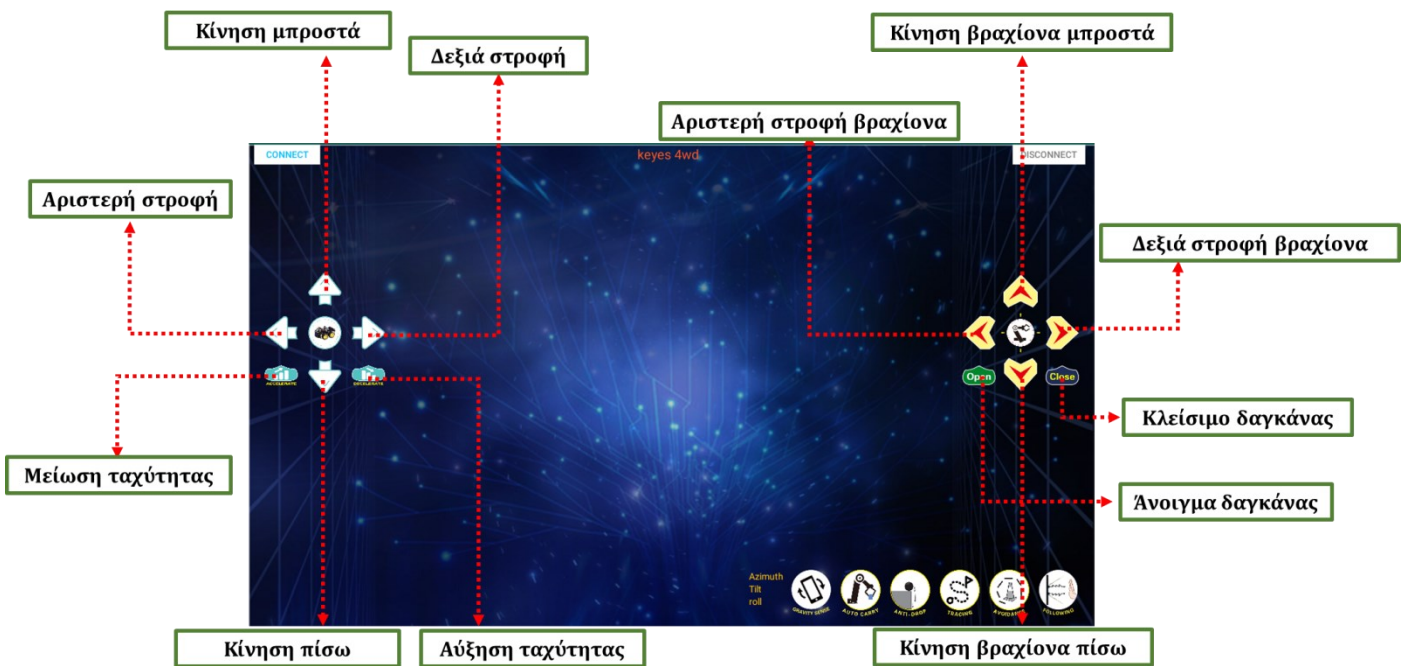


Ενεργοποιήστε το Bluetooth, κάντε κλικ στο εικονίδιο "Σύνδεση" και αναζητήστε το Bluetooth.







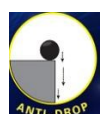

Κάντε κλικ στο «σύνδεση» εάν εμφανιστεί το HMSoFT, το LED Bluetooth θα ανάψει. Μετά τη λήψη, επιτρέπετε στην εφαρμογή να αποκτήσει πρόσβαση στην "τοποθεσία", και έτσι μπορείτε να ενεργοποιήσετε την "τοποθεσία" στις ρυθμίσεις του κινητού σας τηλεφώνου.

### 3. Εφαρμογή Ρομπότ 4



Η λειτουργία κάθε πλήκτρου της εφαρμογής φαίνεται παρακάτω:

	Χαρακτήρας Ελέγχου	<b>Λειτουργία</b>
	Άνοιγμα: L Κλείσιμο: S	Πατήστε το κουμπί, το αυτοκίνητο ρομπότ πηγαίνει μπροστά. Αφήστε το για να σταματήσει.
	Άνοιγμα: R Κλείσιμο: S	Πατήστε το κουμπί, το αυτοκίνητο ρομπότ στρίβει αριστερά. Αφήστε το για να σταματήσει.
	Άνοιγμα: B Κλείσιμο: S	Πατήστε το κουμπί, το αυτοκίνητο ρομπότ πηγαίνει προς τα πίσω. Αφήστε το για να σταματήσει.
	Άνοιγμα: a Κλείσιμο: S	Πατήστε για επιτάχυνση. Αφήστε το για να σταματήσει.
	Άνοιγμα: d Κλείσιμο: S	Πατήστε για να επιβραδύνετε. Αφήστε το για να σταματήσει.
	Άνοιγμα: Q Κλείσιμο: S	Πατήστε για να ανοίξει η δαγκάνα. Αφήστε για να σταματήσει.
	Άνοιγμα: E Κλείσιμο: S	Πατήστε για να κλείσει η δαγκάνα. Αφήστε το για να σταματήσει.
	Άνοιγμα: f Κλείσιμο: s	Πιέστε για να κινήσετε τον βραχίονα προς τα μπροστά. Αφήστε το για να σταματήσει.
	Άνοιγμα: b Κλείσιμο: s	Πατήστε για να κινήσετε τον βραχίονα προς τα πίσω. Αφήστε το για να σταματήσει.
	Άνοιγμα: l Κλείσιμο: s	Πατήστε για να στρίψει ο βραχίονας αριστερά. Αφήστε το για να σταματήσει.
	Άνοιγμα: r Κλείσιμο: s	Πατήστε για να στρίψει ο βραχίονας δεξιά. Αφήστε το για να σταματήσει.
	Άνοιγμα: t Κλείσιμο: s	Πατήστε το εικονίδιο για να αποθηκεύσετε την τρέχουσα τιμή γωνίας (εγγραφή κίνησης).

	Άνοιγμα: i Κλείσιμο: s	Πιέστε για να εκτελέσετε την αποθηκευμένη τιμή της γωνίας της μηχανής σε βρόγχο (εκτέλεση κίνησης).
	Κάντε κλικ για να ξεκινήσει η κινητή ανίχνευση βαρύτητας. Κάντε ξανά κλικ για έξοδο.	
	Κάντε κλικ για αποστολή "Y" και μετά κάντε κλικ στο "S"	Πατήστε το εικονίδιο για να εισέλθετε στη λειτουργία αποφυγής εμποδίων και πατήστε ξανά για έξοδο.
	Κάντε κλικ για να στείλετε το "U" και μετά κάντε κλικ στο "S"	Εκκίνηση της λειτουργίας παρακολούθησης υπερήχων. Κάντε κλικ στο STOP για έξοδο.
	Κάντε κλικ για αποστολή "G" και μετά κάντε κλικ στο "S"	Πατήστε το εικονίδιο για να ενεργοποιήσετε τη λειτουργία προστασίας από πτώση, πατήστε ξανά για έξοδο.
	Κάντε κλικ για να στείλετε το "X" και μετά κάντε κλικ στο "S"	Πατήστε το εικονίδιο για να ενεργοποιήσετε τη λειτουργία παρακολούθησης γραμμής, πατήστε ξανά για να τερματίσετε



### (7) Επεξήγηση κώδικα

**Serial.available():** Οι τρέχοντες χαρακτήρες ανάπαυσης, επιστρέφουν στην προσωρινή περιοχή. Γενικά, αυτή η συνάρτηση χρησιμοποιείται για να κρίνει εάν υπάρχουν δεδομένα στο buffer. Όταν `Serial.available()>0`, σημαίνει ότι η σειρά λαμβάνει τα δεδομένα και μπορεί να διαβαστεί


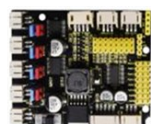







**Serial.read():** σημαίνει ότι η σειρά λαμβάνει τα δεδομένα και μπορεί να διαβαστεί "Serial.read()"

## Δραστηριότητα 1.19: Προγραμματισμός τηλεχειριστηρίου Bluetooth και ελέγχου LED (2 LED)

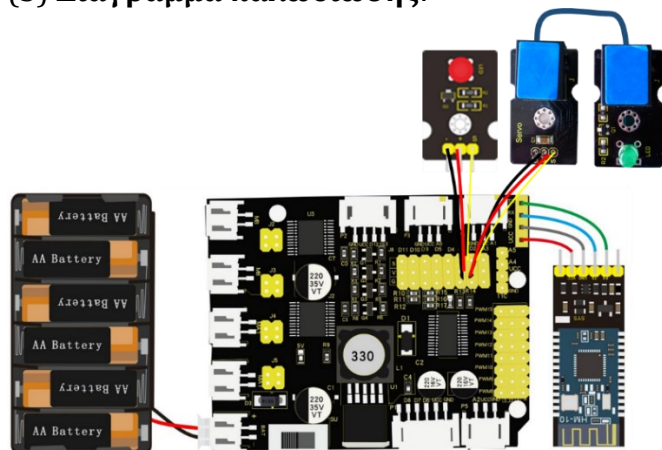
### (1) Περιγραφή:

Σε αυτή την δραστηριότητα σε συνέχεια της προηγούμενης θα προγραμματίσετε το χειριστήριο Bluetooth να ελέγχει ποιο LED ανάβει. Πιο συγκεκριμένα με το πάνω βέλος, , θα ενεργοποιείται το κόκκινο LED ενώ με το κάτω βέλος, , θα ενεργοποιείται το πράσινο LED.

### (2) Τι χρειάζεστε:

Ελεγκτής	Πλακέτα οδήγησης κινητήρων	Βάση μπαταρίας	Μονάδα Bluetooth	Πράσινο LED με καλώδιο UTP
				
Καλώδιο USB	(2x) 3pin F-F Dupont Line	Κόκκινο LED		Επέκταση RJ11
				

### (3) Διάγραμμα καλωδίωσης:



Τα VCC, GND, TXD και RXD της μονάδας Bluetooth συνδέονται με το VCC, το GND, RXD και το TXD της πλακέτας οδήγησης. Οι ακροδέκτες -, + και S της μονάδας κόκκινο LED συνδέονται στα G, V και D3 της πλακέτας οδήγησης. Συνδέστε την επέκταση RJ 11 στα G, V και D4 της πλακέτας οδήγησης και συνδέστε την επέκταση με το πράσινο LED χρησιμοποιώντας το καλώδιο UTP.

Συνδέστε την πλακέτα οδήγησης στον υπολογιστή σας με το καλώδιο USB.

**ΣΗΜΕΙΩΣΗ:1.** Δώστε προσοχή στην κατεύθυνση των ακίδων κατά την εισαγωγή της μονάδας Bluetooth

**2. ΜΗΝ** εισάγετε τη μονάδα Bluetooth πριν ανεβάσετε τον κώδικα. (Για να αποφύγετε την αντίφαση μεταξύ σειριακής επικοινωνίας κώδικα και επικοινωνίας Bluetooth)

**(4) Κώδικας δοκιμής:**

```

/*
Δραστηριότητα 1.19
Τηλεχειριστήριο Bluetooth και Έλεγχος των LED
*/
int ledpin_red=3;
int ledpin_green=4;
bool ledState_red = false; // μεταβλητή κατάστασης του κόκκινου LED
bool ledState_green = false; // μεταβλητή κατάστασης του πράσινου LED

void setup()
{
  Serial.begin(9600);
  pinMode(ledpin_red, OUTPUT);
  pinMode(ledpin_green, OUTPUT);
}

void loop()
{
  if (Serial.available())
  {
    char i = Serial.read();
    Serial.println("DATA RECEIVED: " + String(i));

    if(i == 'F')
    {
      ledState_red = !ledState_red; // αλλάζουμε την κατάσταση του κόκκινου LED
      digitalWrite(ledpin_red, ledState_red);
      Serial.println("led on/off");
    }
    if(i == 'B')
    {
      ledState_green = !ledState_green; // αλλάζουμε την κατάσταση του πράσινου LED
      digitalWrite(ledpin_green, ledState_green);
      Serial.println("led on/off");
    }
    if(i == 'S')
    {
    }

  }
}

```

**Εναλλακτικά:** Μέσα στον φάκελο «ΚΩΔΙΚΕΣ ARDUINO» μπορείτε να ανοίξετε το αρχείο κάνοντας διπλό κλικ πάνω στο αρχείο  M\_1.19 .






**(5) Αποτέλεσμα δοκιμής:**









Ανεβάστε σωστά τον κώδικα χωρίς σφάλματα, έπειτα τοποθετήστε τη μονάδα Bluetooth και τρέξτε την εφαρμογή R4. Πατήστε το άνω βέλος και δείτε το κόκκινο LED να ανάβει. Αντίστοιχα πατήστε το κάτω βέλος για να δείτε το πράσινο LED να ανάβει.

## Δραστηριότητα 1.20: Προγραμματισμός τηλεχειριστηρίου Bluetooth και ελέγχου βομβητή

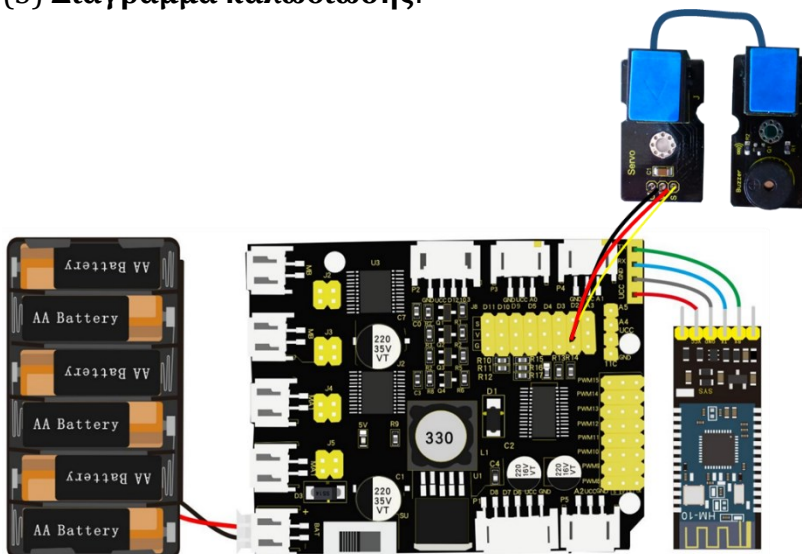
### (1) Περιγραφή:

Σε αυτή την δραστηριότητα, σε συνέχεια της προηγούμενης, θα προγραμματίσετε το χειριστήριο Bluetooth να ελέγχει τη συχνότητα του ήχου του βομβητή. Πιο συγκεκριμένα με το πάνω βέλος,  η συχνότητα του ήχου του βομβητή θα γίνεται πιο γρήγορη και με το κάτω βέλος,  θα γίνεται πιο αργή. Επίσης με το κεντρικό κουμπί , θα ενεργοποιείται ή θα απενεργοποιείται ο βομβητής.

### (2) Τι χρειάζεστε:

Ελεγκτής	Πλακέτα οδήγησης κινητήρων	Βάση μπαταρίας	Μονάδα Bluetooth
			
Καλώδιο USB	(2x) 3pin F-F Dupont Line	Βομβητής με καλώδιο UTP	Επέκταση RJ11
			

### (3) Διάγραμμα καλωδίωσης:



Τα VCC, GND, TXD και RXD της μονάδας Bluetooth συνδέονται με το VCC, το GNDRXD και το TXD της πλακέτας οδήγησης. Οι ακροδέκτες -, + και S της μονάδας κόκκινο LED συνδέονται στα G, V και D3 της πλακέτας οδήγησης.

Συνδέστε την επέκταση RJ 11 στα G, V και D2 της πλακέτας οδήγησης και συνδέστε την επέκταση με τον βομβητή χρησιμοποιώντας το καλώδιο UTP. Συνδέστε την πλακέτα οδήγησης στον υπολογιστή σας με το καλώδιο USB.

**ΣΗΜΕΙΩΣΗ:1.** Δώστε προσοχή στην κατεύθυνση των ακίδων κατά την εισαγωγή της μονάδας Bluetooth

**2. ΜΗΝ** εισάγετε τη μονάδα Bluetooth πριν ανεβάσετε τον κώδικα. (Για να αποφύγετε την αντίφαση μεταξύ σειριακής επικοινωνίας κώδικα και επικοινωνίας Bluetooth)

#### (4) Κώδικας δοκιμής:

```

/*
 Δραστηριότητα 1.20
 Τηλεχειριστήριο Bluetooth και Έλεγχος Βομβητή
 */
int BUZZER_PIN = 2;
bool buzzerState = false; // μεταβλητή κατάστασης του Buzzer
int buzzerSpeed = 1; // αρχική ταχύτητα του Buzzer
int freq=0;

void setup()
{
  Serial.begin(9600);
  pinMode(BUZZER_PIN, OUTPUT);
}

void loop()
{
  if (Serial.available())
  {
    char i = Serial.read();
    Serial.println("DATA RECEIVED: " + String(i));

    if (i == 'F')
    {
      buzzerSpeed -= 1;
      if (buzzerSpeed <1) buzzerSpeed = 1;

      Serial.println("Buzzer speed "+buzzerSpeed);
      Serial.println("Buzzer speed increased");
    }
    if (i == 'B')
    {
      buzzerSpeed += 1;
      if (buzzerSpeed >3) buzzerSpeed = 3;

      Serial.println("Buzzer speed "+buzzerSpeed);
      Serial.println("Buzzer speed decreased");
    }
  }
}

```

```


}
if (i == 'L')
{
  buzzerState = true;
  digitalWrite(BUZZER_PIN, HIGH);
  Serial.println("Buzzer turned on");
}
if (i == 'R')
{
  buzzerState = false;
  digitalWrite(BUZZER_PIN, LOW);
  Serial.println("Buzzer turned off");
}
}

if (buzzerState)
{
  for (int i=0;i<80;i++) // παραγωγή ήχου συχνότητας
  { digitalWrite(BUZZER_PIN, HIGH); // ακούγεται
    delay(1); // καθυστέρηση 1 ms
    digitalWrite(BUZZER_PIN, LOW); // δεν ακούγεται
    delay(2); // καθυστέρηση 2 ms
  }
  for (int i=0;i<100;i++) // παραγωγή ήχου συχνότητας
  {
    digitalWrite(BUZZER_PIN, HIGH); // ακούγεται
    digitalWrite(BUZZER_PIN, LOW); // δεν ακούγεται
    delay(buzzerSpeed); // καθυστέρηση
  }
}
}
}

```

**Εναλλακτικά:** Μέσα στον φάκελο «ΚΩΔΙΚΕΣ ARDUINO» μπορείτε να ανοίξετε το αρχείο κάνοντας διπλό κλικ πάνω στο αρχείο  M\_1.20 .

#### (5) Αποτέλεσμα δοκιμής:

Ανεβάστε σωστά τον κώδικα χωρίς σφάλματα, έπειτα τοποθετήστε τη μονάδα Bluetooth και τρέξτε την εφαρμογή R4. Πατήστε το κουμπί  για να ενεργοποιήσετε το βομβητή. Στη συνέχεια παίξτε με τα βέλη για να ακούσετε τη συχνότητα του ήχου να μεταβάλλεται.

## Δραστηριότητα 1.21: Πρόγραμμα οδήγησης κινητήρα

### (1) Περιγραφή:

Το πρόγραμμα οδήγησης κινητήρα στην πλακέτα επέκτασης υιοθετεί μια ειδική μέθοδο λογικής ελέγχου. Μόνο 4 ακίδες μπορούν να επιτύχουν έλεγχο διπλού κινητήρα. Σε σύγκριση με τα καθαρά τσιπ, δεν διαθέτει δύο ακίδες IO και μπορεί να εφαρμοστεί σε περισσότερα πεδία, εξοικονομώντας πολύτιμους πόρους IO για το Arduino και άλλους ελεγκτές. Είναι ένα τσιπ προγράμματος οδήγησης πλήρους γέφυρας διπλού καναλιού. Το μέγιστο συνεχές ρεύμα κίνησης ενός μεμονωμένου καναλιού μπορεί να φτάσει τα 1,2A και η τιμή αιχμής είναι 2A/3,2A (συνεχής παλμός/μονός παλμός), που μπορεί να οδηγήσει ορισμένους κινητήρες microDC.

Μέχρι τώρα είδατε πως να ελέγχετε και να προγραμματίζετε τους αισθητήρες και τις συσκευές (LED, βομβητής). Από αυτή τη δραστηριότητα και κάτω θα αρχίσετε να προγραμματίζετε τη κίνηση του ρομπότ και το συνδυασμό της κίνησης του με τους αισθητήρες που ήδη μάθατε να προγραμματίζετε. Σε αυτή τη δραστηριότητα θα προγραμματίσετε τους 4 κινητήρες έτσι ώστε το ρομπότ R4 να κινηθεί προς τα μπροστά.

### (2) Χαρακτηριστικά για πλακέτα οδήγησης κινητήρα 4WD

1. Η πλακέτα προγράμματος οδήγησης συνοδεύεται από ένα τσιπ PCA9685 (η διεύθυνση IIC είναι 0x47) που μπορεί να εξάγει εξόδους PWM 8 καναλιών.
2. Οι ακίδες D11/D10/D9/D5/D4/D3/D2/A3 στον πίνακα ελέγχου επεκτείνονται σε κεφαλίδα 3 ακίδων με βήμα 2,54 mm. Η κεφαλίδα 4 ακίδων οδηγεί στη διεπαφή επικοινωνίας I2C και η θηλυκή κεφαλίδα 4 ακίδων οδηγεί στη σειριακή διεπαφή επικοινωνίας, διευκολύνοντας τη σύνδεση της μονάδας Bluetooth.
3. Αναπτύξτε 3 PH2,0mm-3P, 1 PH2,0mm-4P και 1 PH2,0mm-5P διεπαφές κατά της αντίστροφης λειτουργίας, που είναι βολικό για εξωτερικούς αισθητήρες/μονάδες.
4. Το IC τροφοδοσίας της πλακέτας οδηγού είναι LM2596S-5.0, με ισχυρή χωρητικότητα φορτίου, το μέγιστο ρεύμα μπορεί να φτάσει τα 3A και μπορεί να οδηγήσει πολλαπλά εργαλεία διεύθυνσης.
5. Ο πίνακας κίνησης μπορεί να οδηγήσει έως και κινητήρες συνεχούς ρεύματος 12 V.

### (3) Προδιαγραφή:

Τάση λειτουργίας: DC 8-12V

Ρεύμα κίνησης: 3Amax

Μέγιστη ισχύς: 10W






Θερμοκρασία λειτουργίας: -20°C-60°C

Λευκός τύπος διεπαφής: PH2.0 (-2P -3P -4P -5P)

Διάσταση κεφαλίδας καρφίτσας/κεφαλίδας θηλυκών: 2,54 mm

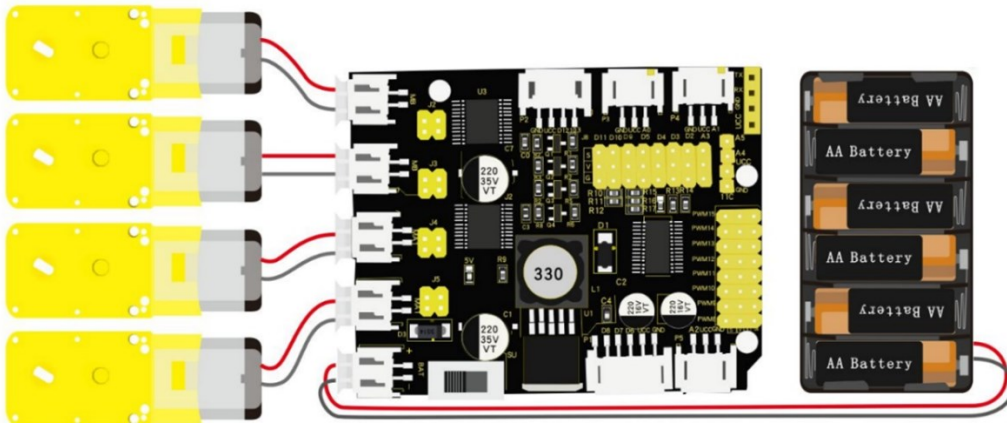
Μέγεθος: 68,7mm x 55mm

## (4) Τι χρειάζεστε:

Ελεγκτής	Πλακέτα οδήγησης κινητήρων	Κινητήρας 4,5V 200rpm	Βάση μπαταρίας	Καλώδιο USB
				

## Σχεδιασμός Κινητήρα

Προσαρμογή κατεύθυνσης	Ρυθμίστε την ταχύτητα περιστροφής	Κατάσταση
<code>pwm.setPWM(0,0,4095)</code>	<code>pwm.setPWM(1,0,1024)</code>	MB ο κινητήρας περιστρέφεται δεξιόστροφα
<code>pwm.setPWM(0,0,0)</code>	<code>pwm.setPWM(1,0,2048)</code>	MB ο κινητήρας περιστρέφεται αριστερόστροφα
<code>pwm.setPWM(2,0,4095)</code>	<code>pwm.setPWM(3,0,1024)</code>	MB1 ο κινητήρας περιστρέφεται δεξιόστροφα
<code>pwm.setPWM(2,0,0)</code>	<code>pwm.setPWM(3,0,2048)</code>	MB1 ο κινητήρας περιστρέφεται αριστερόστροφα
<code>pwm.setPWM(4,0,4095)</code>	<code>pwm.setPWM(5,0,1024)</code>	MA1 ο κινητήρας περιστρέφεται δεξιόστροφα
<code>pwm.setPWM(4,0,0)</code>	<code>pwm.setPWM(5,0,2048)</code>	MA1 ο κινητήρας περιστρέφεται αριστερόστροφα
<code>pwm.setPWM(6,0,4095)</code>	<code>pwm.setPWM(7,0,1024)</code>	MA ο κινητήρας περιστρέφεται δεξιόστροφα
<code>pwm.setPWM(6,0,0)</code>	<code>pwm.setPWM(7,0,2048)</code>	MA ο κινητήρας περιστρέφεται αριστερόστροφα

**(5) Διάγραμμα καλωδίωσης:**

Συνδέστε την πλακέτα οδήγησης στον υπολογιστή σας με το καλώδιο USB.

**(6) Προσαρμόστε την κατεύθυνση περιστροφής των σερβοκινητήρων**

Σε αυτή τη δραστηριότητα θα ρυθμίσετε τους τέσσερις κινητήρες του ρομπότ έτσι ώστε η κατεύθυνση των κινητήρων να είναι ίδια για τα επόμενα μαθήματα. 8 καλύμματα βραχυκυκλωτήρων στην πλακέτα οδηγού χρησιμοποιούνται για τον έλεγχο της κατεύθυνσης των κινητήρων. Για παράδειγμα, αλλάξτε τη φορά των 2 καλυμμάτων βραχυκυκλωτήρα από οριζόντια σύνδεση κινητήρα MA σε κάθετη σύνδεση και δείτε τη φορά περιστροφής του MA να αντριστρέφεται. Ο παρακάτω κώδικας θα κάνει το ρομπότ να προχωρήσει. Η διαφορετική καλωδίωση των τεσσάρων κινητήρων θα προκαλέσει τη διαφορετική φορά περιστροφής των κινητήρων, επομένως, πρέπει να προσαρμόσουμε την κατεύθυνση περιστροφής των κινητήρων μέσω της κατεύθυνσης των καλυμμάτων βραχυκυκλωτήρα.




**(7) Κώδικας δοκιμής:**

```

/*
 Δραστηριότητα 1.21
 Οδήγηση κινητήρα
 */
#include <Wire.h>
#include <Adafruit_PWMServoDriver.h>
Adafruit_PWMServoDriver pwm = Adafruit_PWMServoDriver(0x47);

void setup(){
  pwm.begin();
  pwm.setPWMPFreq(60);
}
void loop(){
  pwm.setPWM(0,0,1000);
  pwm.setPWM(1,0,0);
  pwm.setPWM(2,0, 1000);
  pwm.setPWM(3,0,0);
  pwm.setPWM(4,0,1000);
  pwm.setPWM(5,0,0);
  pwm.setPWM(6,0,1000);
  pwm.setPWM(7,0,0);
}

```

**Εναλλακτικά:** Μέσα στον φάκελο «ΚΩΔΙΚΕΣ ARDUINO» μπορείτε να ανοίξετε το αρχείο κάνοντας διπλό κλικ πάνω στο αρχείο  M\_1.21 .

**(8) Αποτέλεσμα δοκιμής:**

Ανεβάστε σωστά τον κώδικα χωρίς σφάλματα. Το ρομπότ R4 θα πάει προς τα εμπρός, εάν η κατεύθυνση της κίνησης δεν είναι σταθερή, προσαρμόστε τους βραχυκυκλωτήρες.








## Δραστηριότητα 1.22: Πρόγραμμα οδήγησης κινητήρα και ελέγχου κίνησης

### (1) Περιγραφή:

Σε αυτή τη δραστηριότητα, σε συνέχεια της προηγούμενης, θα προγραμματίσετε το ρομπότ R4 να κάνει μια συγκεκριμένη ρουτίνα κινήσεων. Το ρομπότ R4 θα πηγαίνει μπροστά για 1 δευτερόλεπτο, πίσω για 1 δευτερόλεπτο, θα στρίβει αριστερά για 1 δευτερόλεπτο και τέλος θα στρίβει δεξιά για 1 δευτερόλεπτο.

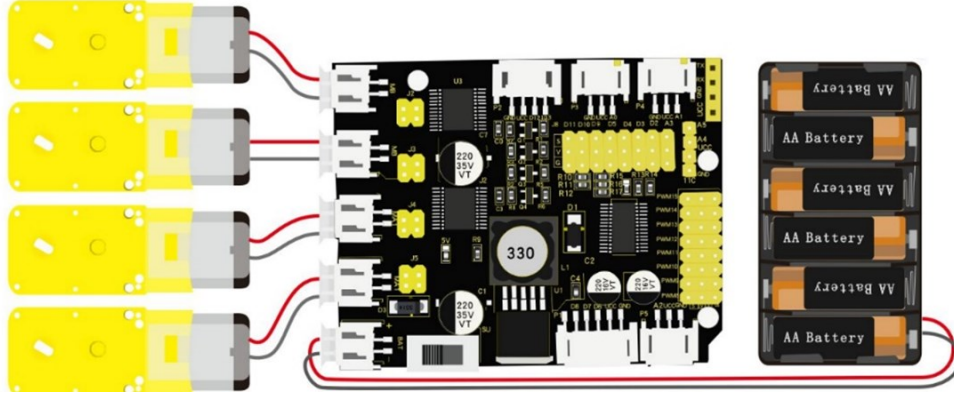
### (2) Τι χρειάζεστε:

Ελεγκτής	Πλακέτα οδήγησης κινητήρων	Κινητήρας 4,5V 200rpm	Βάση μπαταρίας	Καλώδιο USB
				

### Σχεδιασμός Κινητήρα

Προσαρμογή κατεύθυνσης	Ρυθμίστε την ταχύτητα περιστροφής	Κατάσταση
<code>pwm.setPWM(0,0,4095)</code>	<code>pwm.setPWM(1,0,1024)</code>	MB ο κινητήρας περιστρέφεται δεξιόστροφα
<code>pwm.setPWM(0,0,0)</code>	<code>pwm.setPWM(1,0,2048)</code>	MB ο κινητήρας περιστρέφεται αριστερόστροφα
<code>pwm.setPWM(2,0,4095)</code>	<code>pwm.setPWM(3,0,1024)</code>	MB1 ο κινητήρας περιστρέφεται δεξιόστροφα
<code>pwm.setPWM(2,0,0)</code>	<code>pwm.setPWM(3,0,2048)</code>	MB1 ο κινητήρας περιστρέφεται αριστερόστροφας
<code>pwm.setPWM(4,0,4095)</code>	<code>pwm.setPWM(5,0,1024)</code>	MA1 ο κινητήρας περιστρέφεται δεξιόστροφα
<code>pwm.setPWM(4,0,0)</code>	<code>pwm.setPWM(5,0,2048)</code>	MA1 ο κινητήρας περιστρέφεται αριστερόστροφα
<code>pwm.setPWM(6,0,4095)</code>	<code>pwm.setPWM(7,0,1024)</code>	MA ο κινητήρας περιστρέφεται δεξιόστροφα
<code>pwm.setPWM(6,0,0)</code>	<code>pwm.setPWM(7,0,2048)</code>	MA ο κινητήρας περιστρέφεται αριστερόστροφα

## (3) Διάγραμμα καλωδίωσης:



Συνδέστε την πλακέτα οδήγησης στον υπολογιστή σας με το καλώδιο USB.

## (4) Κώδικας δοκιμής:

```

/*
 Δραστηριότητα 1.22
 Οδήγηση κινητήρα και έλεγχος κίνησης
 */
#include <Wire.h>
#include <Adafruit_PWMServoDriver.h>
Adafruit_PWMServoDriver pwm = Adafruit_PWMServoDriver(0x47);
int speed = 1000; // αρχικοποίηση της τιμής ταχύτητας
int turnSpeed = 2000; // αρχικοποίηση της τιμής ταχύτητας για τις στροφές
// Χρησιμοποιούμε τις μεταβλητές speed και turnSpeed για λόγους αποτελεσματικότητας.

void setup() {
  Serial.begin(9600); // Ορίζει το ρυθμό baud σε 9600
  pwm.begin();

  pwm.setPWMPFreq(60);
  stop(); // το αυτοκίνητο σταματάει
}

void loop() {
  advance();
  delay(1000); // κινείται μπροστά για ένα δευτερόλεπτο
  back();
  delay(1000); // κινείται προς τα πίσω για ένα δευτερόλεπτο
  turnL();
  delay(1000); // στρίβει αριστερά για ένα δευτερόλεπτο
  turnR();
  delay(1000); // στρίβει δεξιά για ένα δευτερόλεπτο
  stop();
  delay(1000); // σταματάει για ένα δευτερόλεπτο, έπειτα επαναλαμβάνεται η διαδικασία.
}

```


```
void advance() // Κίνηση μπροστά
{
  pwm.setPWM(0, 0, speed);
  pwm.setPWM(1, 0, 0);
  pwm.setPWM(2, 0, speed);
  pwm.setPWM(3, 0, 0);
  pwm.setPWM(4, 0, speed);
  pwm.setPWM(5, 0, 0);
  pwm.setPWM(6, 0, speed);
  pwm.setPWM(7, 0, 0);
}

void turnR() // Στροφή δεξιά
{
  pwm.setPWM(0, 0, turnSpeed);
  pwm.setPWM(1, 0, 0);
  pwm.setPWM(2, 0, turnSpeed);
  pwm.setPWM(3, 0, 0);
  pwm.setPWM(4, 0, 0);
  pwm.setPWM(5, 0, turnSpeed);
  pwm.setPWM(6, 0, 0);
  pwm.setPWM(7, 0, turnSpeed);
}

void turnL() // Στροφή αριστερά
{
  pwm.setPWM(0, 0, 0);
  pwm.setPWM(1, 0, turnSpeed);
  pwm.setPWM(2, 0, 0);
  pwm.setPWM(3, 0, turnSpeed);
  pwm.setPWM(4, 0, turnSpeed);
  pwm.setPWM(5, 0, 0);
  pwm.setPWM(6, 0, turnSpeed);
  pwm.setPWM(7, 0, 0);
}

void stop() // Σταματάει
{
  pwm.setPWM(0, 0, 0);
  pwm.setPWM(1, 0, 0);
  pwm.setPWM(2, 0, 0);
  pwm.setPWM(3, 0, 0);
  pwm.setPWM(4, 0, 0);
  pwm.setPWM(5, 0, 0);
  pwm.setPWM(6, 0, 0);
  pwm.setPWM(7, 0, 0);
}
```

```
void back() // Κίνηση προς τα πίσω
{
  pwm.setPWM(0, 0, 0);
  pwm.setPWM(1, 0, speed);
  pwm.setPWM(2, 0, 0);
  pwm.setPWM(3, 0, speed);
  pwm.setPWM(4, 0, 0);
  pwm.setPWM(5, 0, speed);
  pwm.setPWM(6, 0, 0);
  pwm.setPWM(7, 0, speed);
}
```

**Εναλλακτικά:** Μέσα στον φάκελο «ΚΩΔΙΚΕΣ ARDUINO» μπορείτε να ανοίξετε το αρχείο κάνοντας διπλό κλικ πάνω στο αρχείο  M\_1.22 .

**(5)Αποτέλεσμα δοκιμής:**


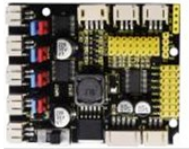






Ανεβάστε σωστά τον κώδικα χωρίς σφάλματα. Το ρομπότ R4 πηγαίνει μπροστά για 1 δευτερόλεπτο, πίσω για 1 δευτερόλεπτο, στρίβει αριστερά για 1 δευτερόλεπτο και στρίβει δεξιά για 1 δευτερόλεπτο.

## Δραστηριότητα 1.23: Πρόγραμμα οδήγησης κινητήρα και ηχητική ειδοποίηση

### (1) Περιγραφή:

Σε αυτή τη δραστηριότητα, σε συνέχεια της προηγούμενης, θα προσθέσουμε τον βομβητή ο οποίος θα σας ενημερώνει κάθε φορά που τελειώνει μια κίνηση και πριν ξεκινήσει την επόμενη.

### (2) Τι χρειάζεστε:

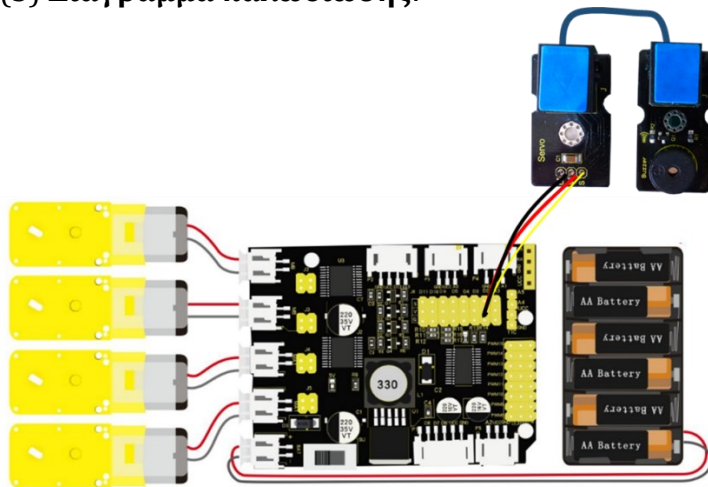
Ελεγκτής	Πλακέτα οδήγησης κινητήρων	Κινητήρας 4,5V 200rpm	Βομβητής με καλώδιο UTP
			
Επέκταση RJ11	Βάση μπαταρίας	Καλώδιο USB	3pin F-F Dupont Line
			

### Σχεδιασμός Κινητήρα

Προσαρμογή κατεύθυνσης	Ρυθμίστε την ταχύτητα περιστροφής	Κατάσταση
<code>pwm.setPWM(0,0,4095)</code>	<code>pwm.setPWM(1,0,1024)</code>	MB ο κινητήρας περιστρέφεται δεξιόστροφα
<code>pwm.setPWM(0,0,0)</code>	<code>pwm.setPWM(1,0,2048)</code>	MB ο κινητήρας περιστρέφεται αριστερόστροφα
<code>pwm.setPWM(2,0,4095)</code>	<code>pwm.setPWM(3,0,1024)</code>	MB1 ο κινητήρας περιστρέφεται δεξιόστροφα
<code>pwm.setPWM(2,0,0)</code>	<code>pwm.setPWM(3,0,2048)</code>	MB1 ο κινητήρας περιστρέφεται αριστερόστροφα
<code>pwm.setPWM(4,0,4095)</code>	<code>pwm.setPWM(5,0,1024)</code>	MA1 ο κινητήρας περιστρέφεται δεξιόστροφα
<code>pwm.setPWM(4,0,0)</code>	<code>pwm.setPWM(5,0,2048)</code>	MA1 ο κινητήρας περιστρέφεται αριστερόστροφα
<code>pwm.setPWM(6,0,4095)</code>	<code>pwm.setPWM(7,0,1024)</code>	MA ο κινητήρας περιστρέφεται δεξιόστροφα

<code>pwm.setPWM(6,0,0)</code>	<code>pwm.setPWM(7,0,2048)</code>	ΜΑ ο κινητήρας περιστρέφεται αριστερόστροφα
--------------------------------	-----------------------------------	---

### (3) Διάγραμμα καλωδίωσης:



Συνδέστε την επέκταση RJ 11 στα G, V και D2 της πλακέτας οδήγησης και συνδέστε την επέκταση με τον βομβητή χρησιμοποιώντας το καλώδιο UTP. Συνδέστε την πλακέτα οδήγησης στον υπολογιστή σας με το καλώδιο USB.

### (4) Κώδικας δοκιμής:

```

/*
 Δραστηριότητα 1.23
 Οδήγηση κινητήρα και έλεγχος κίνησης
 */
#include <Wire.h>
#include <Adafruit_PWMServoDriver.h>
Adafruit_PWMServoDriver pwm = Adafruit_PWMServoDriver(0x47);

int BUZZER_PIN = 2;
int speed = 1000;
int turnSpeed = 2000;

void setup() {
  pinMode(BUZZER_PIN, OUTPUT); // ρυθμίστε το μοτίβο ψηφιακών ακροδεκτών IO, OUTPUT για
  έξοδο
  Serial.begin(9600); // Ορίζει τον ρυθμό baud σε 9600bit.
  pwm.begin();

  pwm.setPWMPFreq(60);
  stop(); // Το αυτοκίνητο σταματάει
}

void loop() {
  advance(); // κινείται μπροστά για ένα δευτερόλεπτο
  buzzerCall();
  delay(1000);
  back(); // κινείται μπροστά για ένα δευτερόλεπτο
  buzzerCall();
}

```

```

delay(1000);
turnL(); // κινείται μπροστά για ένα δευτερόλεπτο
buzzerCall();
delay(1000);
turnR(); // κινείται μπροστά για ένα δευτερόλεπτο
buzzerCall();
delay(1000);
stop(); // κινείται μπροστά για ένα δευτερόλεπτο
delay(1000);
}

```

```

void advance() // Κίνηση μπροστά
{
  pwm.setPWM(0, 0, speed);
  pwm.setPWM(1, 0, 0);
  pwm.setPWM(2, 0, speed);
  pwm.setPWM(3, 0, 0);
  pwm.setPWM(4, 0, speed);
  pwm.setPWM(5, 0, 0);
  pwm.setPWM(6, 0, speed);
  pwm.setPWM(7, 0, 0);
}

```

```

void turnR() // Στροφή δεξιά
{
  pwm.setPWM(0, 0, turnSpeed);
  pwm.setPWM(1, 0, 0);
  pwm.setPWM(2, 0, turnSpeed);
  pwm.setPWM(3, 0, 0);
  pwm.setPWM(4, 0, 0);
  pwm.setPWM(5, 0, turnSpeed);
  pwm.setPWM(6, 0, 0);
  pwm.setPWM(7, 0, turnSpeed);
}

```

```

void turnL() // Στροφή αριστερά
{
  pwm.setPWM(0, 0, 0);
  pwm.setPWM(1, 0, turnSpeed);
  pwm.setPWM(2, 0, 0);
  pwm.setPWM(3, 0, turnSpeed);
  pwm.setPWM(4, 0, turnSpeed);
  pwm.setPWM(5, 0, 0);
  pwm.setPWM(6, 0, turnSpeed);
  pwm.setPWM(7, 0, 0);
}

```

```

void stop() // Σταματάει
{
  pwm.setPWM(0, 0, 0);
}

```


```

pwm.setPWM(1, 0, 0);
pwm.setPWM(2, 0, 0);
pwm.setPWM(3, 0, 0);
pwm.setPWM(4, 0, 0);
pwm.setPWM(5, 0, 0);
pwm.setPWM(6, 0, 0);
pwm.setPWM(7, 0, 0);
}

void back() // Κίνηση προς τα πίσω
{
  pwm.setPWM(0, 0, 0);
  pwm.setPWM(1, 0, speed);
  pwm.setPWM(2, 0, 0);
  pwm.setPWM(3, 0, speed);
  pwm.setPWM(4, 0, 0);
  pwm.setPWM(5, 0, speed);
  pwm.setPWM(6, 0, 0);
  pwm.setPWM(7, 0, speed);
}

void buzzerCall(){
  for(int i=0; i<180; i++) // παραγωγή ήχου συχνότητας
  { digitalWrite(BUZZER_PIN, HIGH); // παράγει ήχο
    delay(1); // καθυστέρηση 1 ms
    digitalWrite(BUZZER_PIN, LOW); // δεν παράγει ήχο
    delay(1); // καθυστέρηση 1 ms
  }
  for(int i=0; i<100; i++)// παραγωγή ήχου συχνότητας
  {
    digitalWrite(BUZZER_PIN, HIGH); // παράγει ήχο
    digitalWrite(BUZZER_PIN, LOW); // δεν παράγει ήχο
    delay(2); // καθυστέρηση 2 ms
  }
}

```

**Εναλλακτικά:** Μέσα στον φάκελο «ΚΩΔΙΚΕΣ ARDUINO» μπορείτε να ανοίξετε το αρχείο κάνοντας διπλό κλικ πάνω στο αρχείο  M\_1.23 .

#### (5)Αποτέλεσμα δοκιμής:

Ανεβάστε σωστά τον κώδικα χωρίς σφάλματα. Το ρομπότ R4 πηγαίνει μπροστά για 1 δευτερόλεπτο και έπειτα ακούγεται ο βομβητής, πηγαίνει πίσω για 1 δευτερόλεπτο και έπειτα ακούγεται ο βομβητής, στρίβει αριστερά για 1 δευτερόλεπτο και έπειτα ακούγεται ο βομβητής και τέλος στρίβει δεξιά για 1 δευτερόλεπτο και έπειτα ακούγεται ο βομβητής.


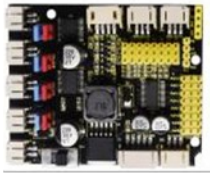








## Δραστηριότητα 1.24: Σχεδιασμός κατά της πτώσης

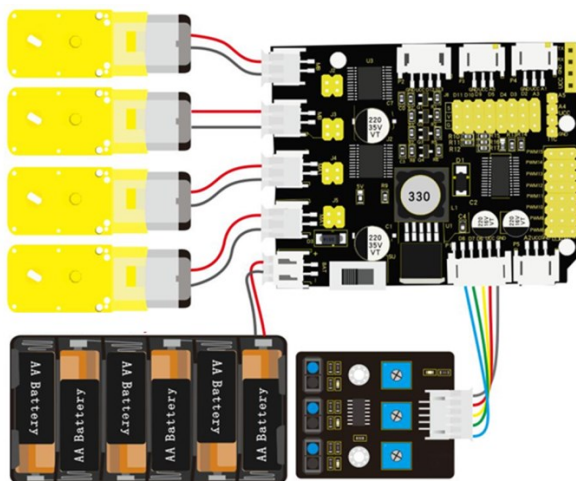
### (1) Περιγραφή:

Στη δραστηριότητα αυτή θα δείτε τη λειτουργία του ρομπότ R4 κατά της πτώσης. Πιο συγκεκριμένα θα προγραμματίσετε το ρομπότ R4 να κινείται σε ευθεία γραμμή και όταν πλησιάζει στην άκρη του γραφείου να σταματάει, να κάνει προς τα πίσω και έπειτα να στρίβει αριστερά.

### (2) Τι χρειάζεστε:

Ελεγκτής	Πλακέτα οδήγησης κινητήρων	Κινητήρας 4,5V 200rpm	Αισθητήρας παρακολούθησης γραμμής
			
5pin F-F Dupont Line	Βάση μπαταρίας	Καλώδιο USB	3pin F-F Dupont Line
			

### (3) Διάγραμμα καλωδίωσης:



Συνδέστε την πλακέτα οδήγησης στον υπολογιστή σας με το καλώδιο USB.

## (4) Κώδικας δοκιμής:

```

/*
 Δραστηριότητα 1.24
 Σχεδιασμός κατά της πτώσης
 */
#include <Wire.h>
#include <Adafruit_PWMServoDriver.h>
Adafruit_PWMServoDriver pwm = Adafruit_PWMServoDriver(0x47);
int speed = 1000; // Αρχικοποίηση της τιμής της ταχύτητας
int turnSpeed = 2000; // Αρχικοποίηση της ταχύτητας για τις μεθόδους στροφής

#define SensorLeft 6 // καρφίτσα εισαγωγής του αριστερού αισθητήρα
#define SensorMiddle 7 // καρφίτσα εισαγωγής του μεσαίου αισθητήρα
#define SensorRight 8 // καρφίτσα εισαγωγής του δεξιού αισθητήρα

unsigned char SL; // κατάσταση αριστερού αισθητήρα
unsigned char SM; // κατάσταση μεσαίου αισθητήρα
unsigned char SR; // κατάσταση δεξιού αισθητήρα

void setup() {
  Serial.begin(9600); // Έναρξη σειριακής θύρας στη συχνότητα 9600bit.
  pwm.begin();
  pwm.setPWMPFreq(60); // ρυθμίστε τον αισθητήρα παρακολούθησης γραμμής σε λειτουργία
  εισόδου
  pinMode(SensorLeft, INPUT);
  pinMode(SensorMiddle, INPUT);
  pinMode(SensorRight, INPUT);
  stop(); // Το αυτοκίνητο σταματάει
}

void loop() {
  // ανάγνωση των τιμών του αισθητήρα παρακολούθησης γραμμής
  SL = digitalRead(SensorLeft);
  SM = digitalRead(SensorMiddle);
  SR = digitalRead(SensorRight);
  // αν και τα δύο είναι χαμηλά, προχωρήστε προς τα εμπρός
  if (SM == LOW && SL == LOW && SR == LOW) {
    advance();
  }else{
  // διαφορετικά, γυρίστε πίσω και στρίψτε αριστερά
  back();
  delay(500);
  turnL();
  delay(500);
  }
}

```


```
void advance() // Κίνηση μπροστά
{
  pwm.setPWM(0, 0, speed);
  pwm.setPWM(1, 0, 0);
  pwm.setPWM(2, 0, speed);
  pwm.setPWM(3, 0, 0);
  pwm.setPWM(4, 0, speed);
  pwm.setPWM(5, 0, 0);
  pwm.setPWM(6, 0, speed);
  pwm.setPWM(7, 0, 0);
}

void turnR() // Στροφή δεξιά
{
  pwm.setPWM(0, 0, turnSpeed);
  pwm.setPWM(1, 0, 0);
  pwm.setPWM(2, 0, turnSpeed);
  pwm.setPWM(3, 0, 0);
  pwm.setPWM(4, 0, 0);
  pwm.setPWM(5, 0, turnSpeed);
  pwm.setPWM(6, 0, 0);
  pwm.setPWM(7, 0, turnSpeed);
}

void turnL() // Στροφή αριστερά
{
  pwm.setPWM(0, 0, 0);
  pwm.setPWM(1, 0, turnSpeed);
  pwm.setPWM(2, 0, 0);
  pwm.setPWM(3, 0, turnSpeed);
  pwm.setPWM(4, 0, turnSpeed);
  pwm.setPWM(5, 0, 0);
  pwm.setPWM(6, 0, turnSpeed);
  pwm.setPWM(7, 0, 0);
}

void stop() // Σταματάει
{
  pwm.setPWM(0, 0, 0);
  pwm.setPWM(1, 0, 0);
  pwm.setPWM(2, 0, 0);
  pwm.setPWM(3, 0, 0);
  pwm.setPWM(4, 0, 0);
  pwm.setPWM(5, 0, 0);
  pwm.setPWM(6, 0, 0);
  pwm.setPWM(7, 0, 0);
}
```

```
void back() // Κίνηση προς τα πίσω
{
  pwm.setPWM(0, 0, 0);
  pwm.setPWM(1, 0, speed);
  pwm.setPWM(2, 0, 0);
  pwm.setPWM(3, 0, speed);
  pwm.setPWM(4, 0, 0);
  pwm.setPWM(5, 0, speed);
  pwm.setPWM(6, 0, 0);
  pwm.setPWM(7, 0, speed);
}
```

**Εναλλακτικά:** Μέσα στον φάκελο «ΚΩΔΙΚΕΣ ARDUINO» μπορείτε να ανοίξετε το αρχείο κάνοντας διπλό κλικ πάνω στο αρχείο  M\_1.24 .

#### (4) Αποτέλεσμα δοκιμής:


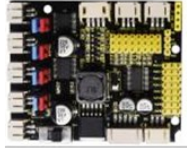

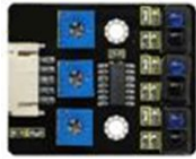






Ανεβάστε σωστά τον κώδικα χωρίς σφάλματα. Το ρομπότ R4 θα αρχίσει να κινείται στην ευθεία και όταν πλησιάσει στην άκρη του γραφείου θα κινηθεί προς τα πίσω, θα στρίψει αριστερά και θα συνεχίσει τη πορεία του.

## Δραστηριότητα 1.25: Σχεδιασμός κατά της πτώσης και ηχητική ειδοποίηση

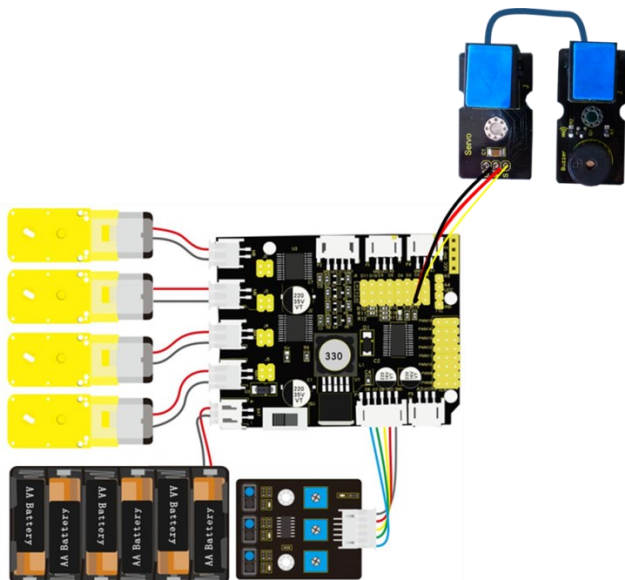
### (1) Περιγραφή:

Σε αυτή τη δραστηριότητα, σε συνέχεια της προηγούμενης, θα προσθέσετε τον βομβητή ο οποίος θα σας ενημερώνει πότε το ρομπότ R4 φτάνει στην άκρη του γραφείου.

### (2) Τι χρειάζεστε:

Ελεγκτής	Πλακέτα οδήγησης κινητήρων	Κινητήρας 4,5V 200rpm	Αισθητήρας παρακολούθησης γραμμής	Βομβητής με καλώδιο UTP
				
5pin F-F Dupont Line	Βάση μπαταρίας	Καλώδιο USB	3pin F-F Dupont Line	Επέκταση RJ11
				

### (3) Διάγραμμα καλωδίωσης:



Συνδέστε την επέκταση RJ 11 στα G, V και D2 της πλακέτας οδήγησης και συνδέστε την επέκταση με τον βομβητή χρησιμοποιώντας το καλώδιο UTP. Συνδέστε την πλακέτα οδήγησης στον υπολογιστή σας με το καλώδιο USB.

## (4) Κώδικας δοκιμής:

```

/*
 Δραστηριότητα 1.25
 Σχεδιασμός κατά της πτώσης και ηχητική ειδοποίηση
*/
#include <Wire.h>
#include <Adafruit_PWMServoDriver.h>
Adafruit_PWMServoDriver pwm = Adafruit_PWMServoDriver(0x47);
int speed = 1000; // Αρχικοποίηση της τιμής της ταχύτητας
int turnSpeed = 2000; // Αρχικοποίηση της ταχύτητας για τις μεθόδους στροφής
int BUZZER_PIN = 2; // Ορισμός pin του βομβητή

#define SensorLeft 6 // καρφίτσα εισαγωγής του αριστερού αισθητήρα
#define SensorMiddle 7 // καρφίτσα εισαγωγής του μεσαίου αισθητήρα
#define SensorRight 8 // καρφίτσα εισαγωγής του δεξιού αισθητήρα

unsigned char SL; // κατάσταση αριστερού αισθητήρα
unsigned char SM; // κατάσταση μεσαίου αισθητήρα
unsigned char SR; // κατάσταση δεξιού αισθητήρα

void setup() {
  Serial.begin(9600); // Έναρξη σειριακής θύρας στη συχνότητα 9600bit.
  pwm.begin();
  pwm.setPWMPFreq(60); // ρυθμίστε τον αισθητήρα παρακολούθησης γραμμής σε λειτουργία
  εισόδου
  pinMode(SensorLeft, INPUT);
  pinMode(SensorMiddle, INPUT);
  pinMode(SensorRight, INPUT);
  stop(); // Το αυτοκίνητο σταματάει
}

void loop() {
  // ανάγνωση των τιμών του αισθητήρα παρακολούθησης γραμμής
  SL = digitalRead(SensorLeft);
  SM = digitalRead(SensorMiddle);
  SR = digitalRead(SensorRight);
  // αν το σήμα και από τους 3 σένσορες είναι χαμηλό, το αυτοκίνητο κινείται προς τα εμπρός
  if (SM == LOW && SL == LOW && SR == LOW) {
    advance();
  }else{
    digitalWrite(BUZZER_PIN, HIGH);
    back();
    delay(500);
    turnR();
    delay(500);
    digitalWrite(BUZZER_PIN, LOW);
  }
}

```


```
void advance() // Κίνηση μπροστά
{
  pwm.setPWM(0, 0, speed);
  pwm.setPWM(1, 0, 0);
  pwm.setPWM(2, 0, speed);
  pwm.setPWM(3, 0, 0);
  pwm.setPWM(4, 0, speed);
  pwm.setPWM(5, 0, 0);
  pwm.setPWM(6, 0, speed);
  pwm.setPWM(7, 0, 0);
}

void turnR() // Στροφή δεξιά
{
  pwm.setPWM(0, 0, turnSpeed);
  pwm.setPWM(1, 0, 0);
  pwm.setPWM(2, 0, turnSpeed);
  pwm.setPWM(3, 0, 0);
  pwm.setPWM(4, 0, 0);
  pwm.setPWM(5, 0, turnSpeed);
  pwm.setPWM(6, 0, 0);
  pwm.setPWM(7, 0, turnSpeed);
}

void turnL() // Στροφή αριστερά
{
  pwm.setPWM(0, 0, 0);
  pwm.setPWM(1, 0, turnSpeed);
  pwm.setPWM(2, 0, 0);
  pwm.setPWM(3, 0, turnSpeed);
  pwm.setPWM(4, 0, turnSpeed);
  pwm.setPWM(5, 0, 0);
  pwm.setPWM(6, 0, turnSpeed);
  pwm.setPWM(7, 0, 0);
}

void stop() // Σταματάει
{
  pwm.setPWM(0, 0, 0);
  pwm.setPWM(1, 0, 0);
  pwm.setPWM(2, 0, 0);
  pwm.setPWM(3, 0, 0);
  pwm.setPWM(4, 0, 0);
  pwm.setPWM(5, 0, 0);
  pwm.setPWM(6, 0, 0);
  pwm.setPWM(7, 0, 0);
}
```

```
void back() // Κίνηση προς τα πίσω
{
  pwm.setPWM(0, 0, 0);
  pwm.setPWM(1, 0, speed);
  pwm.setPWM(2, 0, 0);
  pwm.setPWM(3, 0, speed);
  pwm.setPWM(4, 0, 0);
  pwm.setPWM(5, 0, speed);
  pwm.setPWM(6, 0, 0);
  pwm.setPWM(7, 0, speed);
}
```

**Εναλλακτικά:** Μέσα στον φάκελο «ΚΩΔΙΚΕΣ ARDUINO» μπορείτε να ανοίξετε το αρχείο κάνοντας διπλό κλικ πάνω στο αρχείο  M\_1.25 .

#### (5) Αποτέλεσμα δοκιμής:

Ανεβάστε σωστά τον κώδικα χωρίς σφάλματα. Το ρομπότ R4 θα αρχίσει να κινείται στην ευθεία και όταν πλησιάσει στην άκρη του γραφείου θα ενεργοποιήσει τον βομβητή, στη συνέχεια θα κινηθεί προς τα πίσω, θα στρίψει αριστερά και θα συνεχίσει τη πορεία του.


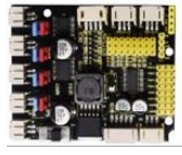








## Δραστηριότητα 1.26: Προγραμματισμός παρακολούθησης γραμμής έξυπνου αυτοκινήτου

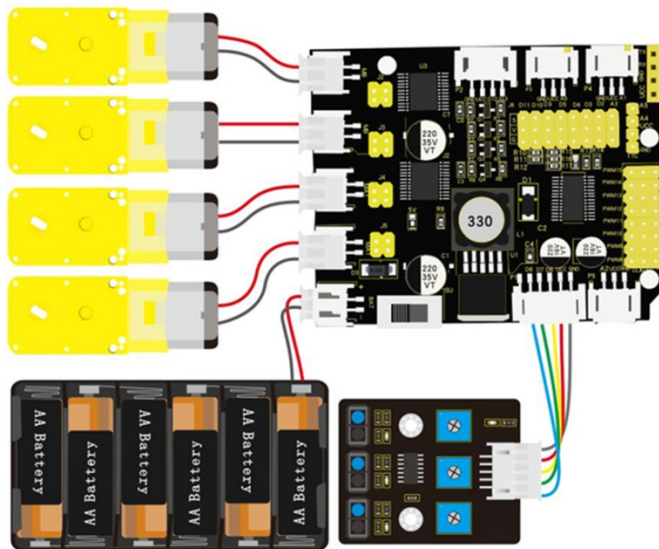
### (1) Περιγραφή:

Στη δραστηριότητα αυτή θα προγραμματίσετε το ρομπότ R4 να κινείται πάνω σε οποιαδήποτε μαύρη γραμμή που έχετε σχεδιάσει είτε είναι ευθεία είτε κύκλος.

### (2) Τι χρειάζεστε:

Ελεγκτής	Πλακέτα οδήγησης κινητήρων	Κινητήρας 4,5V 200rpm	Αισθητήρας παρακολούθησης γραμμής
			
5pin F-F Dupont Line	Βάση μπαταρίας	Καλώδιο USB	3pin F-F Dupont Line
			

### (3) Διάγραμμα καλωδίωσης:



Συνδέστε την πλακέτα οδήγησης στον υπολογιστή σας με το καλώδιο USB.

## (4) Κώδικας δοκιμής:

```

/*
 Δραστηριότητα 1.26
 Παρακολούθηση γραμμής έξυπνου αυτοκινήτου
*/
#include <Wire.h>
#include <Adafruit_PWMServoDriver.h>
Adafruit_PWMServoDriver pwm = Adafruit_PWMServoDriver(0x47);
int speed = 1000; // Αρχικοποίηση της τιμής της ταχύτητας
int turnSpeed = 2000; // Αρχικοποίηση της ταχύτητας για τις μεθόδους στροφής

#define SensorLeft 6 // καρφίτσα εισαγωγής του αριστερού αισθητήρα
#define SensorMiddle 7 // καρφίτσα εισαγωγής του μεσαίου αισθητήρα
#define SensorRight 8 // καρφίτσα εισαγωγής του δεξιού αισθητήρα

unsigned char SL; // κατάσταση αριστερού αισθητήρα
unsigned char SM; // κατάσταση μεσαίου αισθητήρα
unsigned char SR; // κατάσταση δεξιού αισθητήρα

void setup() {
  Serial.begin(9600); // Έναρξη σειριακής θύρας στη συχνότητα 9600bit
  pwm.begin();
  pwm.setPWMPFreq(60);
  pinMode(SensorLeft, INPUT);
  pinMode(SensorMiddle, INPUT);
  pinMode(SensorRight, INPUT);
  stop();
}

void loop() {
  // ανάγνωση των τιμών του αισθητήρα παρακολούθησης γραμμής
  SL = digitalRead(SensorLeft);
  SM = digitalRead(SensorMiddle);
  SR = digitalRead(SensorRight);
  if (SM == HIGH) {
    if (SL == LOW && SR == HIGH) { // Αν ανιχνευτεί μαύρο στα δεξιά και λευκό στα αριστερά, στρίβει
δεξιά
      turnR();
    }else if (SR == LOW && SL == HIGH) { // αν ανιχνευτεί μαύρο στα αριστερά και λευκό στα δεξιά,
στρίβει αριστερά
      turnL();
    }else { // Αν ανιχνευτεί λευκό και στις δύο άκρες, κινείται ευθεία
      advance();
    }
  }else {
    if (SL == LOW && SR == HIGH) { // Αν ανιχνευτεί μαύρο στα δεξιά και λευκό στα αριστερά, στρίβει
δεξιά
      turnR();
    }else if (SR == LOW && SL == HIGH) { // Αν ανιχνευτεί μαύρο στα αριστερά και λευκό στα δεξιά,
στρίβει αριστερά.

```

```

    turnL();
} else { // Αν όλοι οι σένσορες ανιχνεύουν λευκό, το αυτοκίνητο σταματάει.
    stop();
}
}
}

```

```

void advance() // Κίνηση μπροστά
{
    pwm.setPWM(0, 0, speed);
    pwm.setPWM(1, 0, 0);
    pwm.setPWM(2, 0, speed);
    pwm.setPWM(3, 0, 0);
    pwm.setPWM(4, 0, speed);
    pwm.setPWM(5, 0, 0);
    pwm.setPWM(6, 0, speed);
    pwm.setPWM(7, 0, 0);
}

```

```

void turnR() // Στροφή δεξιά
{
    pwm.setPWM(0, 0, turnSpeed);
    pwm.setPWM(1, 0, 0);
    pwm.setPWM(2, 0, turnSpeed);
    pwm.setPWM(3, 0, 0);
    pwm.setPWM(4, 0, 0);
    pwm.setPWM(5, 0, turnSpeed);
    pwm.setPWM(6, 0, 0);
    pwm.setPWM(7, 0, turnSpeed);
}

```

```

void turnL() // Στροφή αριστερά
{
    pwm.setPWM(0, 0, 0);
    pwm.setPWM(1, 0, turnSpeed);
    pwm.setPWM(2, 0, 0);
    pwm.setPWM(3, 0, turnSpeed);
    pwm.setPWM(4, 0, turnSpeed);
    pwm.setPWM(5, 0, 0);
    pwm.setPWM(6, 0, turnSpeed);
    pwm.setPWM(7, 0, 0);
}


```

```

void stop() // Σταματάει
{
    pwm.setPWM(0, 0, 0);
    pwm.setPWM(1, 0, 0);
    pwm.setPWM(2, 0, 0);
    pwm.setPWM(3, 0, 0);
    pwm.setPWM(4, 0, 0);
}

```

```
pwm.setPWM(5, 0, 0);  
pwm.setPWM(6, 0, 0);  
pwm.setPWM(7, 0, 0);  
}  
  
void back() // Κίνηση προς τα πίσω  
{  
  pwm.setPWM(0, 0, 0);  
  pwm.setPWM(1, 0, speed);  
  pwm.setPWM(2, 0, 0);  
  pwm.setPWM(3, 0, speed);  
  pwm.setPWM(4, 0, 0);  
  pwm.setPWM(5, 0, speed);  
  pwm.setPWM(6, 0, 0);  
  pwm.setPWM(7, 0, speed);  
}
```

**Εναλλακτικά:** Μέσα στον φάκελο «ΚΩΔΙΚΕΣ ARDUINO» μπορείτε να ανοίξετε το αρχείο κάνοντας διπλό κλικ πάνω στο αρχείο  M\_1.26 .

#### (5) Αποτέλεσμα δοκιμής:

Ανεβάστε σωστά τον κώδικα χωρίς σφάλματα. Σχεδιάστε μία μαύρη γραμμή και παρακολουθήστε το ρομπότ R4 να κινείται πάνω στη μαύρη γραμμή.











**Σημείωση:** Το πάχος της μαύρης γραμμής πρέπει να είναι τόσο ώστε να καλύπτει και τους 3 σωλήνες υπερύθρων.

## Δραστηριότητα 1.27: Προγραμματισμός παρακολούθησης γραμμής έξυπνου αυτοκινήτου και ηχητική ειδοποίηση

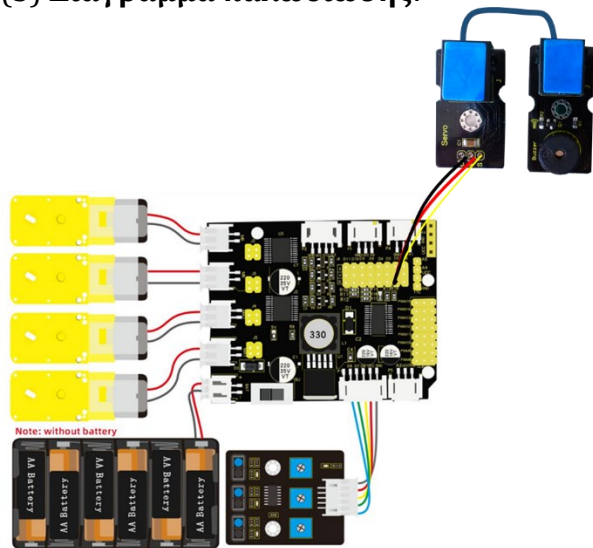
### (1) Περιγραφή:

Στη δραστηριότητα αυτή, σε συνέχεια της προηγούμενης, θα προσθέσετε τον βομβητή ο οποίος θα χτυπάει όσο το ρομπότ κινείται και θα σταματάει όταν σταματάει και το ρομπότ να κινείται.

### (2) Τι χρειάζεστε:

Ελεγκτής	Πλακέτα οδήγησης κινητήρων	Κινητήρας 4,5V 200rpm	Αισθητήρας παρακολούθησης γραμμής	Βομβητής με καλώδιο UTP
				
5pin F-F Dupont Line	Βάση μπαταρίας	Καλώδιο USB	3pin F-F Dupont Line	Επέκταση RJ11
				

### (3) Διάγραμμα καλωδίωσης:



Συνδέστε την επέκταση RJ 11 στα G, V και D2 της πλακέτας οδήγησης και συνδέστε την επέκταση με το βομβητή χρησιμοποιώντας το καλώδιο UTP.

Συνδέστε την πλακέτα οδήγησης στον υπολογιστή σας με το καλώδιο USB.

## (4) Κώδικας δοκιμής:

```

/*
 Δραστηριότητα 1.27
 Παρακολούθησης γραμμής έξυπνου αυτοκινήτου και ηχητική ειδοποίηση
*/
#include <Wire.h>
#include <Adafruit_PWMServoDriver.h>
Adafruit_PWMServoDriver pwm = Adafruit_PWMServoDriver(0x47);
int speed = 1000; // Αρχικοποίηση της τιμής της ταχύτητας
int turnSpeed = 2000; // Αρχικοποίηση της ταχύτητας για τις μεθόδους στροφής
int BUZZER_PIN = 2; // Ορισμός pin του βομβητή

#define SensorLeft 6 // καρφίτσα εισαγωγής του αριστερού αισθητήρα
#define SensorMiddle 7 // καρφίτσα εισαγωγής του μεσαίου αισθητήρα
#define SensorRight 8 // καρφίτσα εισαγωγής του δεξιού αισθητήρα

unsigned char SL; // κατάσταση αριστερού αισθητήρα
unsigned char SM; // κατάσταση μεσαίου αισθητήρα
unsigned char SR; // κατάσταση δεξιού αισθητήρα

void setup() {
  Serial.begin(9600); // Έναρξη σειριακής θύρας στη συχνότητα 9600bit
  pwm.begin();
  pwm.setPWMPFreq(60);
  pinMode(SensorLeft, INPUT);
  pinMode(SensorMiddle, INPUT);
  pinMode(SensorRight, INPUT);
  stop();
}

void loop() {
  // ανάγνωση των τιμών του αισθητήρα παρακολούθησης γραμμής
  SL = digitalRead(SensorLeft);
  SM = digitalRead(SensorMiddle);
  SR = digitalRead(SensorRight);
  if (SM == HIGH) {
    digitalWrite(BUZZER_PIN, HIGH); // παράγει ήχο
    if (SL == LOW && SR == HIGH) { // Αν ανιχνευτεί μαύρο στα δεξιά και λευκό στα αριστερά, στρίβει
δεξιά
      turnR();
    }else if (SR == LOW && SL == HIGH) { // αν ανιχνευτεί μαύρο στα αριστερά και λευκό στα δεξιά,
στρίβει αριστερά
      turnL();
    }else { // Αν ανιχνευτεί λευκό και στις δύο άκρες, κινείται ευθεία
      advance();
    }
  }else {
    if (SL == LOW && SR == HIGH) { // Αν ανιχνευτεί μαύρο στα δεξιά και λευκό στα αριστερά, στρίβει
δεξιά

```

```

    digitalWrite(BUZZER_PIN, HIGH); // παράγει ήχο
    turnR();
  }else if (SR == LOW && SL == HIGH) { // Αν ανιχνευτεί μαύρο στα αριστερά και λευκό στα δεξιά,
στρίβει αριστερά.
    digitalWrite(BUZZER_PIN, HIGH); // παράγει ήχο
    turnL();
  }else { // Αν όλοι οι σένσορες ανιχνεύουν λευκό, το αυτοκίνητο σταματάει.
    digitalWrite(BUZZER_PIN, LOW); // σταματάει να παράγει ήχο
    stop();
  }
}
}
}

```

```

void advance() // going forwards

```

```

{
  pwm.setPWM(0, 0, speeds);
  pwm.setPWM(1, 0, 0);
  pwm.setPWM(2, 0, speeds);
  pwm.setPWM(3, 0, 0);
  pwm.setPWM(4, 0, speeds);
  pwm.setPWM(5, 0, 0);
  pwm.setPWM(6, 0, speeds);
  pwm.setPWM(7, 0, 0);
}

```

```

void turnR() // turn right

```

```

{
  pwm.setPWM(0, 0, turnSpeeds);
  pwm.setPWM(1, 0, 0);
  pwm.setPWM(2, 0, turnSpeeds);
  pwm.setPWM(3, 0, 0);
  pwm.setPWM(4, 0, 0);
  pwm.setPWM(5, 0, turnSpeeds);
  pwm.setPWM(6, 0, 0);
  pwm.setPWM(7, 0, turnSpeeds);
}

```

```

void turnL() // turn left

```

```

{
  pwm.setPWM(0, 0, 0);
  pwm.setPWM(1, 0, turnSpeeds);
  pwm.setPWM(2, 0, 0);
  pwm.setPWM(3, 0, turnSpeeds);
  pwm.setPWM(4, 0, turnSpeeds);
  pwm.setPWM(5, 0, 0);
  pwm.setPWM(6, 0, turnSpeeds);
  pwm.setPWM(7, 0, 0);
}

```

```

void stopp() //stop


```

```

{
  pwm.setPWM(0, 0, 0);
}

```

```
pwm.setPWM(1, 0, 0);  
pwm.setPWM(2, 0, 0);  
pwm.setPWM(3, 0, 0);  
pwm.setPWM(4, 0, 0);  
pwm.setPWM(5, 0, 0);  
pwm.setPWM(6, 0, 0);  
pwm.setPWM(7, 0, 0);  
}  
void back() //back  
{  
  pwm.setPWM(0, 0, 0);  
  pwm.setPWM(1, 0, speeds);  
  pwm.setPWM(2, 0, 0);  
  pwm.setPWM(3, 0, speeds);  
  pwm.setPWM(4, 0, 0);  
  pwm.setPWM(5, 0, speeds);  
  pwm.setPWM(6, 0, 0);  
  pwm.setPWM(7, 0, speeds);  
}
```

**Εναλλακτικά:** Μέσα στον φάκελο «ΚΩΔΙΚΕΣ ARDUINO» μπορείτε να ανοίξετε το αρχείο κάνοντας διπλό κλικ πάνω στο αρχείο  M\_1.27 .

#### (5) Αποτέλεσμα δοκιμής:

Ανεβάστε σωστά τον κώδικα χωρίς σφάλματα. Σχεδιάστε μία μαύρη γραμμή (είτε ευθεία είτε κύκλο είτε με γωνίες) και παρακολουθήστε το ρομπότ R4 να κινείται πάνω στη μαύρη γραμμή και ακούστε τον βομβητή.

**Σημείωση:** Το πάχος της μαύρης γραμμής πρέπει να είναι τόσο ώστε να καλύπτει και τους 3 σωλήνες υπέρυθρων.



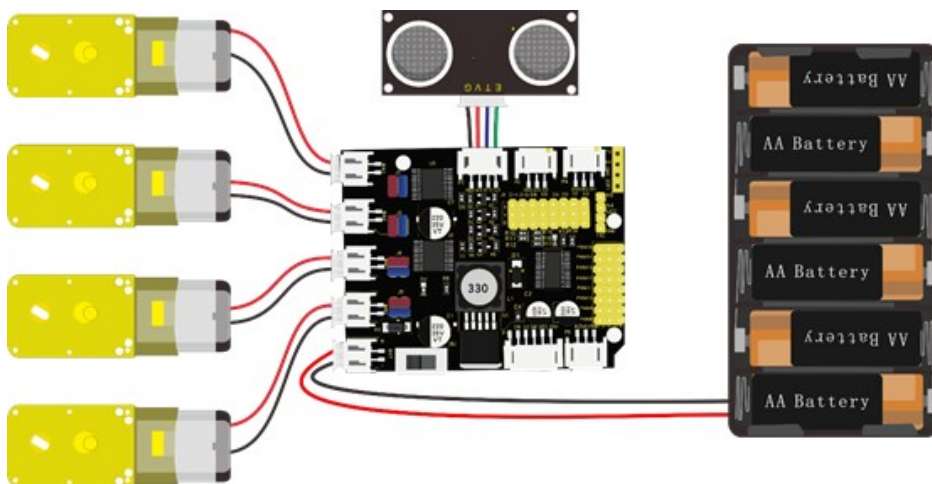
## Δραστηριότητα 1.28: Προγραμματισμός παρακολούθησης εμποδίου

### (1) Περιγραφή:

Σε αυτή τη δραστηριότητα θα προγραμματίσετε το ρομπότ R4 να παρακολουθεί ένα εμπόδιο.

Ο αισθητήρας υπερήχων ανιχνεύει την απόσταση του εμποδίου και στέλνει δεδομένα στον ελεγκτή ενός τσιπ, επομένως μπορείτε να προγραμματίσετε τους κινητήρες να κινούνται σύμφωνα με τα δεδομένα που λαμβάνει ο ελεγκτής.

### (2) Διάγραμμα καλωδίωσης:



Συνδέστε την πλακέτα οδήγησης στον υπολογιστή σας με το καλώδιο USB.

### (3) Κώδικας δοκιμής:

```

/*
 Δραστηριότητα 1.28
 Έξυπνο αυτοκίνητο παρακολούθησης εμποδίου
 */
#include <Wire.h>
#include <Adafruit_PWMServoDriver.h>
Adafruit_PWMServoDriver pwm = Adafruit_PWMServoDriver(0x47);
int speed = 2000; // Αρχικοποίηση της τιμής της ταχύτητας
int turnSpeed = 2000; // Αρχικοποίηση της ταχύτητας για τις μεθόδους στροφής
int echoPin = 13; // Ορισμός μονάδα υπερήχων ECHO στο D13
int trigPin = 12; // Ορισμός μονάδα υπερήχων TRIG στο D12

int Ultrasonic_Ranging() {
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  int distance = pulseIn(echoPin, HIGH); // Ανάγνωση της διάρκειας των υψηλών επιπέδων
  distance = distance / 58; // Μετατροπή του χρόνου παλμού σε απόσταση
  delay(50);
}

```

```

return distance;
}

void setup() {
  Serial.begin(9600); // Έναρξη σειριακής θύρας στη συχνότητα 9600
  pwm.begin();
  pwm.setPWMPfreq(60);
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  stop();
}

void loop() {
  int distance = Ultrasonic_Ranging();
  if (distance < 40) { // Αν η απόσταση είναι μικρότερη από 40 εκατοστά
    if (distance < 20) { // Αν η απόσταση είναι μικρότερη από 20 εκατοστά
      if (distance < 15) { // Αν η απόσταση είναι μικρότερη από 15 εκατοστά
        back();
      }
    }
    else {
      stop();
    }
  }
  else {
    advance();
  }
}
else {
  stop();
}
}

void advance() // Κίνηση μπροστά
{
  pwm.setPWM(0, 0, speed);
  pwm.setPWM(1, 0, 0);
  pwm.setPWM(2, 0, speed);
  pwm.setPWM(3, 0, 0);
  pwm.setPWM(4, 0, speed);
  pwm.setPWM(5, 0, 0);
  pwm.setPWM(6, 0, speed);
  pwm.setPWM(7, 0, 0);
}

void turnR() // Στροφή δεξιά
{
  pwm.setPWM(0, 0, turnSpeed);
  pwm.setPWM(1, 0, 0);
  pwm.setPWM(2, 0, turnSpeed);
  pwm.setPWM(3, 0, 0);
}

```

```


    pwm.setPWM(4, 0, 0);
    pwm.setPWM(5, 0, turnSpeed);
    pwm.setPWM(6, 0, 0);
    pwm.setPWM(7, 0, turnSpeed);
}

void turnL() // Στροφή αριστερά
{
    pwm.setPWM(0, 0, 0);
    pwm.setPWM(1, 0, turnSpeed);
    pwm.setPWM(2, 0, 0);
    pwm.setPWM(3, 0, turnSpeed);
    pwm.setPWM(4, 0, turnSpeed);
    pwm.setPWM(5, 0, 0);
    pwm.setPWM(6, 0, turnSpeed);
    pwm.setPWM(7, 0, 0);
}

void stop() // Σταματάει
{
    pwm.setPWM(0, 0, 0);
    pwm.setPWM(1, 0, 0);
    pwm.setPWM(2, 0, 0);
    pwm.setPWM(3, 0, 0);
    pwm.setPWM(4, 0, 0);
    pwm.setPWM(5, 0, 0);
    pwm.setPWM(6, 0, 0);
    pwm.setPWM(7, 0, 0);
}

void back() // Κίνηση προς τα πίσω
{
    pwm.setPWM(0, 0, 0);
    pwm.setPWM(1, 0, speed);
    pwm.setPWM(2, 0, 0);
    pwm.setPWM(3, 0, speed);
    pwm.setPWM(4, 0, 0);
    pwm.setPWM(5, 0, speed);
    pwm.setPWM(6, 0, 0);
    pwm.setPWM(7, 0, speed);
}

```

**Εναλλακτικά:** Μέσα στον φάκελο «ΚΩΔΙΚΕΣ ARDUINO» μπορείτε να ανοίξετε το αρχείο κάνοντας διπλό κλικ πάνω στο αρχείο  M\_1.28 .

#### (4) Αποτέλεσμα δοκιμής:

Ανεβάστε σωστά τον κώδικα χωρίς σφάλματα. Τοποθετήστε το χέρι σας μπροστά από τον αισθητήρα υπερήχων σε απόσταση μεγαλύτερη από 20 εκατοστά και μικρότερη από 40 εκατοστά. Το

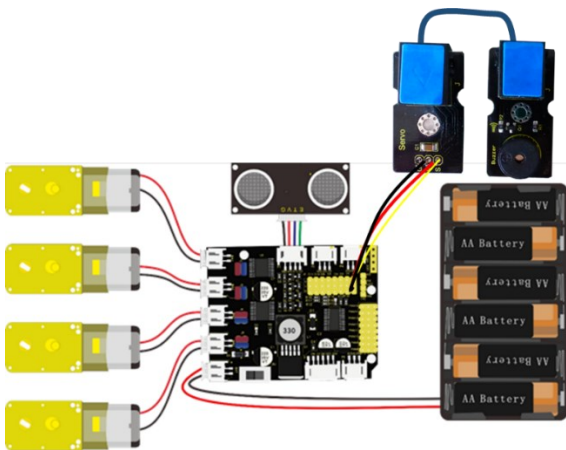
ρομπότ R4 θα κινηθεί προς το χέρι σας και όταν η απόσταση γίνει μικρότερη από 20 εκατοστά, τότε θα κινηθεί προς τα πίσω. Επαναλάβετε και δείτε το ρομπότ R4 να ακολουθεί το χέρι σας.

## Δραστηριότητα 1.29: Προγραμματισμός παρακολούθησης εμποδίου και ηχητική ειδοποίηση

### (1) Περιγραφή:

Σε αυτή τη δραστηριότητα, σε συνέχεια της προηγούμενης, θα προσθέσετε τον βομβητή ο οποίος θα σας ενημερώνει όταν το αυτοκίνητο κινείται προς τα πίσω.

### (2) Διάγραμμα καλωδίωσης:



Συνδέστε την επέκταση RJ 11 στα G, V και D2 της πλακέτας οδήγησης και συνδέστε την επέκταση με τον βομβητή χρησιμοποιώντας το καλώδιο UTP. Συνδέστε την πλακέτα οδήγησης στον υπολογιστή σας με το καλώδιο USB.

### (3) Κώδικας δοκιμής:

```

/*
 Δραστηριότητα 1.29
 Έξυπνο αυτοκίνητο παρακολούθησης εμποδίου και ηχητική ειδοποίηση
 */
#include <Wire.h>
#include <Adafruit_PWMServoDriver.h>
Adafruit_PWMServoDriver pwm = Adafruit_PWMServoDriver(0x47);
int speed = 1000; // Αρχικοποίηση της τιμής της ταχύτητας
int turnSpeed = 2000; // Αρχικοποίηση της ταχύτητας για τις μεθόδους στροφής
int BUZZER_PIN = 2; // Ορισμός pin του βομβητή

int Ultrasonic_Ranging() {
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  int distance = pulseIn(echoPin, HIGH); // Ανάγνωση της διάρκειας των υψηλών επιπέδων
  distance = distance / 58; // Μετατροπή του χρόνου παλμού σε απόσταση
}

```

```

    delay(50);
    return distance;
}

void setup() {
    Serial.begin(9600); // Έναρξη σειριακής θύρας στη συχνότητα 9600
    pwm.begin();
    pwm.setPWMFreq(60);
    pinMode(trigPin, OUTPUT);
    pinMode(echoPin, INPUT);
    stop();
}

void loop() {
    int distance = Ultrasonic_Ranging();
    if (distance < 40) { // Αν η απόσταση είναι μικρότερη από 40 εκατοστά
        if (distance < 20) { // Αν η απόσταση είναι μικρότερη από 20 εκατοστά
            if (distance < 15) { // Αν η απόσταση είναι μικρότερη από 15 εκατοστά
                digitalWrite(BUZZER_PIN, HIGH); // Συχνότητα ήχου υψηλή
                back();
                delay(1000);
                digitalWrite(BUZZER_PIN, LOW); // Συχνότητα ήχου χαμηλή
            }
            else {
                stop();
            }
        }
        else {
            advance();
        }
    }
    else {
        stop();
    }
}

void advance() // Κίνηση μπροστά
{
    pwm.setPWM(0, 0, speed);
    pwm.setPWM(1, 0, 0);
    pwm.setPWM(2, 0, speed);
    pwm.setPWM(3, 0, 0);
    pwm.setPWM(4, 0, speed);
    pwm.setPWM(5, 0, 0);
    pwm.setPWM(6, 0, speed);
    pwm.setPWM(7, 0, 0);
}

void turnR() // Στροφή δεξιά
{

```

```

pwm.setPWM(0, 0, turnSpeed);
pwm.setPWM(1, 0, 0);
pwm.setPWM(2, 0, turnSpeed);
pwm.setPWM(3, 0, 0);
pwm.setPWM(4, 0, 0);
pwm.setPWM(5, 0, turnSpeed);
pwm.setPWM(6, 0, 0);
pwm.setPWM(7, 0, turnSpeed);
}

```

```

void turnL() // Στροφή αριστερά
{
pwm.setPWM(0, 0, 0);
pwm.setPWM(1, 0, turnSpeed);
pwm.setPWM(2, 0, 0);
pwm.setPWM(3, 0, turnSpeed);
pwm.setPWM(4, 0, turnSpeed);
pwm.setPWM(5, 0, 0);
pwm.setPWM(6, 0, turnSpeed);
pwm.setPWM(7, 0, 0);
}

```

```


void stop() // Σταματάει
{
pwm.setPWM(0, 0, 0);
pwm.setPWM(1, 0, 0);
pwm.setPWM(2, 0, 0);
pwm.setPWM(3, 0, 0);
pwm.setPWM(4, 0, 0);
pwm.setPWM(5, 0, 0);
pwm.setPWM(6, 0, 0);
pwm.setPWM(7, 0, 0);
}

```

```

void back() // Κίνηση προς τα πίσω
{
pwm.setPWM(0, 0, 0);
pwm.setPWM(1, 0, speed);
pwm.setPWM(2, 0, 0);
pwm.setPWM(3, 0, speed);
pwm.setPWM(4, 0, 0);
pwm.setPWM(5, 0, speed);
pwm.setPWM(6, 0, 0);
pwm.setPWM(7, 0, speed);
}

```

**Εναλλακτικά:** Μέσα στον φάκελο «ΚΩΔΙΚΕΣ ARDUINO» μπορείτε να ανοίξετε το αρχείο κάνοντας διπλό κλικ πάνω στο αρχείο  M\_1.29 .

(4) Αποτέλεσμα δοκιμής:

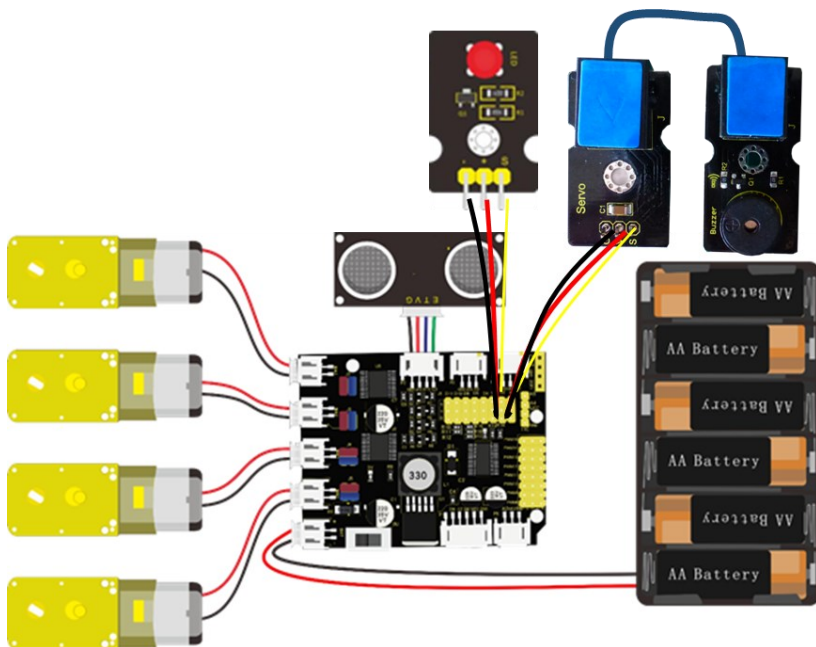
Ανεβάστε σωστά τον κώδικα χωρίς σφάλματα. Τοποθετήστε το χέρι σας μπροστά από τον αισθητήρα υπερήχων σε απόσταση μεγαλύτερη από 20 εκατοστά και μικρότερη από 40 εκατοστά. Το ρομπότ R4 θα κινηθεί προς το χέρι σας και όταν η απόσταση γίνει μικρότερη από 20 εκατοστά, τότε θα ακούσετε τον βομβητή και το αυτοκίνητο θα κινηθεί προς τα πίσω. Επαναλάβετε και δείτε το ρομπότ R4 να ακολουθεί το χέρι σας.

## Δραστηριότητα 1.30: Έξυπνο αυτοκίνητο παρακολούθησης εμποδίου και ταυτόχρονη ηχητική και οπτική ειδοποίηση

### (1) Περιγραφή:

Σε αυτή τη δραστηριότητα, σε συνέχεια της προηγούμενης, θα προσθέσετε το κόκκινο LED το οποίο θα ανάβει όσο το ρομπότ R4 κινείται προς το εμπόδιο και ο βομβητής θα σας ενημερώνει όταν το αυτοκίνητο κινείται προς τα πίσω.

### (2) Διάγραμμα καλωδίωσης:



Συνδέστε τους ακροδέκτες -, + και S της μονάδας κόκκινο LED στα G, V και D3 της πλακέτας οδήγησης. Συνδέστε την επέκταση RJ 11 στα G, V και D2 της πλακέτας οδήγησης και συνδέστε την επέκταση με τον βομβητή χρησιμοποιώντας το καλώδιο UTP. Συνδέστε την πλακέτα οδήγησης στον υπολογιστή σας με το καλώδιο USB.

### (3) Κώδικας δοκιμής:

```
/*
 Δραστηριότητα 1.30
 Έξυπνο αυτοκίνητο παρακολούθησης εμποδίου με ηχητική ειδοποίηση & LED
 */
#include <Wire.h>
#include <Adafruit_PWMServoDriver.h>
Adafruit_PWMServoDriver pwm = Adafruit_PWMServoDriver(0x47);
int BUZZER_PIN = 2; // Ορισμός βομβητή στο PIN 2
```

```

int Red_Led = 3; // Ορισμός κόκκινου LED στο PIN 3
int speed = 2000; // Αρχικοποίηση της τιμής ταχύτητας
int turnSpeed = 2000; // Αρχικοποίηση της τιμής ταχύτητας για τις στροφές
int echoPin = 13; // Ορισμός μονάδα υπερήχων ECHO στο D13
int trigPin = 12; // Ορισμός μονάδα υπερήχων TRIG στο D12

int Ultrasonic_Ranging() {
    digitalWrite(trigPin, LOW);
    delayMicroseconds(2);
    digitalWrite(trigPin, HIGH);
    delayMicroseconds(10);
    digitalWrite(trigPin, LOW);
    int distance = pulseIn(echoPin, HIGH); // Ανάγνωση της διάρκειας των υψηλών επιπέδων
    distance = distance / 58; // Μετατροπή του χρόνου παλμού σε απόσταση
    delay(50);
    return distance;
}

void setup() {
    Serial.begin(9600); // Έναρξη σειριακής θύρας στη συχνότητα 9600
    pinMode(Red_Led, OUTPUT); // Pin κόκκινου LED
    pinMode(BUZZER_PIN, OUTPUT); // Pin βομβητή
    pwm.begin();
    pwm.setPWMPFreq(60);
    pinMode(trigPin, OUTPUT);
    pinMode(echoPin, INPUT);
    stop();
}

void loop() {
    int distance = Ultrasonic_Ranging();
    if (distance >= 20 && distance <= 40) { // Ελέγχει αν η απόσταση είναι μεταξύ 20 και 40 εκατοστά
        digitalWrite(Red_Led, HIGH); // Το κόκκινο LED ενεργοποιείται
        advance();
        digitalWrite(BUZZER_PIN, LOW); // Ο βομβητής απενεργοποιείται
    } else if (distance < 20) { // Ελέγχει αν η απόσταση είναι μικρότερη από 20 εκατοστά
        digitalWrite(Red_Led, LOW); // Το κόκκινο LED απενεργοποιείται
        digitalWrite(BUZZER_PIN, HIGH); // Ο βομβητής ενεργοποιείται
        back();
        delay(1000);
        digitalWrite(BUZZER_PIN, LOW); // Ο βομβητής απενεργοποιείται
    } else {
        digitalWrite(Red_Led, LOW); // Το κόκκινο LED απενεργοποιείται
        stop();
        digitalWrite(BUZZER_PIN, LOW); // Ο βομβητής απενεργοποιείται
    }
}

void advance() // Κίνηση μπροστά
{

```



```

pwm.setPWM(0, 0, speed);
pwm.setPWM(1, 0, 0);
pwm.setPWM(2, 0, speed);
pwm.setPWM(3, 0, 0);
pwm.setPWM(4, 0, speed);
pwm.setPWM(5, 0, 0);
pwm.setPWM(6, 0, speed);
pwm.setPWM(7, 0, 0);
}

```

```

void turnR() // Στροφή δεξιά
{
  pwm.setPWM(0, 0, turnSpeed);
  pwm.setPWM(1, 0, 0);
  pwm.setPWM(2, 0, turnSpeed);
  pwm.setPWM(3, 0, 0);
  pwm.setPWM(4, 0, 0);
  pwm.setPWM(5, 0, turnSpeed);
  pwm.setPWM(6, 0, 0);
  pwm.setPWM(7, 0, turnSpeed);
}

```

```

void turnL() // Στροφή αριστερά
{
  pwm.setPWM(0, 0, 0);
  pwm.setPWM(1, 0, turnSpeed);
  pwm.setPWM(2, 0, 0);
  pwm.setPWM(3, 0, turnSpeed);
  pwm.setPWM(4, 0, turnSpeed);
  pwm.setPWM(5, 0, 0);
  pwm.setPWM(6, 0, turnSpeed);
  pwm.setPWM(7, 0, 0);
}

```

```

void stop() // Σταματάει
{
  pwm.setPWM(0, 0, 0);
  pwm.setPWM(1, 0, 0);
  pwm.setPWM(2, 0, 0);
  pwm.setPWM(3, 0, 0);
  pwm.setPWM(4, 0, 0);
  pwm.setPWM(5, 0, 0);
  pwm.setPWM(6, 0, 0);
  pwm.setPWM(7, 0, 0);
}

```

```

void back() // Κίνηση προς τα πίσω
{
  pwm.setPWM(0, 0, 0);
}

```

```
pwm.setPWM(1, 0, speed);  
pwm.setPWM(2, 0, 0);  
pwm.setPWM(3, 0, speed);  
pwm.setPWM(4, 0, 0);  
pwm.setPWM(5, 0, speed);  
pwm.setPWM(6, 0, 0);  
pwm.setPWM(7, 0, speed);  
}
```

**Εναλλακτικά:** Μέσα στον φάκελο «ΚΩΔΙΚΕΣ ARDUINO» μπορείτε να ανοίξετε το αρχείο κάνοντας διπλό κλικ πάνω στο αρχείο  M\_1.30 .

#### (4) Αποτέλεσμα δοκιμής:

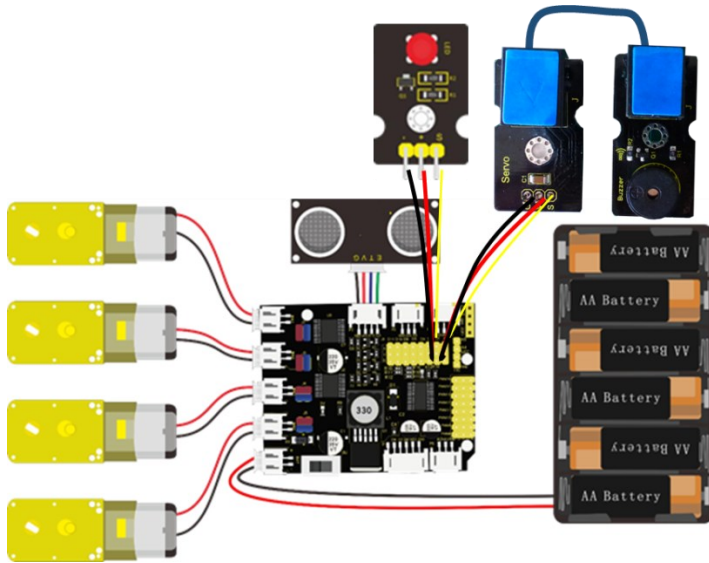
Ανεβάστε σωστά τον κώδικα χωρίς σφάλματα. Τοποθετήστε το χέρι σας μπροστά από τον αισθητήρα υπερήχων σε απόσταση μεγαλύτερη από 20 εκατοστά και μικρότερη από 40 εκατοστά. Το ρομπότ R4 θα κινηθεί προς το χέρι σας και το κόκκινο LED θα είναι αναμμένο, όταν η απόσταση γίνει μικρότερη από 20 εκατοστά τότε θα κινηθεί προς τα πίσω και θα ακουστεί ο βομβητής. Επαναλάβετε και δείτε το ρομπότ R4 να ακολουθεί το χέρι σας.

## Δραστηριότητα 1.31: Έξυπνο αυτοκίνητο αποφυγής εμποδίων και ηχητική ειδοποίηση

### (1) Περιγραφή:

Στη δραστηριότητα αυτή θα προγραμματίσετε το ρομπότ R4 να αναγνωρίζει το εμπόδιο και όταν βρεθεί σε απόσταση μικρότερη από 20 εκατοστά, τότε να σας ενημερώνει ηχητικά και να σταματάει να κινείται.

### (2) Διάγραμμα καλωδίωσης:



Οι ακροδέκτες -, + και S της μονάδας LED συνδέονται στα G, V και D3 της πλακέτας οδήγησης. Συνδέστε την επέκταση RJ 11 στα G, V και D4 της πλακέτας οδήγησης και συνδέστε την επέκταση με το πράσινο LED χρησιμοποιώντας το καλώδιο UTP. Συνδέστε την πλακέτα οδήγησης στον υπολογιστή σας με το καλώδιο USB.

### (3) Κώδικας δοκιμής:

```
/*
 Δραστηριότητα 1.31
 Έξυπνο αυτοκίνητο αποφυγής εμποδίων
*/
#include <Wire.h>
#include <Adafruit_PWMServoDriver.h>
Adafruit_PWMServoDriver pwm = Adafruit_PWMServoDriver(0x47);
int BUZZER_PIN = 2; // Ορισμός βομβητή στο PIN 2
int speed = 2000; // Αρχικοποίηση της τιμής ταχύτητας
int turnSpeed = 2000; // Αρχικοποίηση της τιμής ταχύτητας για τις στροφές
int echoPin = 13; // Ορισμός μονάδα υπερήχων ECHO στο D13
int trigPin = 12; // Ορισμός μονάδα υπερήχων TRIG στο D12

int Ultrasonic_Ranging() {
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
```

```

digitalWrite(trigPin, HIGH);
delayMicroseconds(10);
digitalWrite(trigPin, LOW);
int distance = pulseIn(echoPin, HIGH); // Ανάγνωση της διάρκειας των υψηλών επιπέδων
distance = distance / 58;           // Μετατροπή του χρόνου παλμού σε απόσταση
delay(50);
return distance;
}

```

```

void setup() {
  Serial.begin(9600);
  pwm.begin();
  pwm.setPWMFreq(60);
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  stop();
}

```

```

void loop() {
  int distance = Ultrasonic_Ranging();
  Serial.print("distance=");
  Serial.println(distance);
  if (distance < 30) {
    if (distance < 15) {
      digitalWrite(BUZZER_PIN, HIGH); // Ο βομβητής ενεργοποιείται
      stop();
      delay(100);
      digitalWrite(BUZZER_PIN, LOW); // Ο βομβητής απενεργοποιείται
    } else {
      stop();
      delay(100);
      turnL();
      delay(500);
    }
  }
  else {
    advance();
  }
}

```

```

void advance() // Κίνηση μπροστά
{
  pwm.setPWM(0, 0, speed);
  pwm.setPWM(1, 0, 0);
  pwm.setPWM(2, 0, speed);
  pwm.setPWM(3, 0, 0);
  pwm.setPWM(4, 0, speed);
  pwm.setPWM(5, 0, 0);
  pwm.setPWM(6, 0, speed);
  pwm.setPWM(7, 0, 0);
}

```

}

```
void turnR() // Στροφή δεξιά
{
  pwm.setPWM(0, 0, turnSpeed);
  pwm.setPWM(1, 0, 0);
  pwm.setPWM(2, 0, turnSpeed);
  pwm.setPWM(3, 0, 0);
  pwm.setPWM(4, 0, 0);
  pwm.setPWM(5, 0, turnSpeed);
  pwm.setPWM(6, 0, 0);
  pwm.setPWM(7, 0, turnSpeed);
}

void turnL() // Στροφή αριστερά
{
  pwm.setPWM(0, 0, 0);
  pwm.setPWM(1, 0, turnSpeed);
  pwm.setPWM(2, 0, 0);
  pwm.setPWM(3, 0, turnSpeed);
  pwm.setPWM(4, 0, turnSpeed);
  pwm.setPWM(5, 0, 0);
  pwm.setPWM(6, 0, turnSpeed);
  pwm.setPWM(7, 0, 0);
}

void stop() // Σταματάει
{
  pwm.setPWM(0, 0, 0);
  pwm.setPWM(1, 0, 0);
  pwm.setPWM(2, 0, 0);
  pwm.setPWM(3, 0, 0);
  pwm.setPWM(4, 0, 0);
  pwm.setPWM(5, 0, 0);
  pwm.setPWM(6, 0, 0);
  pwm.setPWM(7, 0, 0);
}

void back() // Κίνηση προς τα πίσω
{
  pwm.setPWM(0, 0, 0);
  pwm.setPWM(1, 0, speed);
  pwm.setPWM(2, 0, 0);
  pwm.setPWM(3, 0, speed);
  pwm.setPWM(4, 0, 0);
  pwm.setPWM(5, 0, speed);
  pwm.setPWM(6, 0, 0);
  pwm.setPWM(7, 0, speed);
}
```

**Εναλλακτικά:** Μέσα στον φάκελο «ΚΩΔΙΚΕΣ ARDUINO» μπορείτε να ανοίξετε το αρχείο κάνοντας διπλό κλικ πάνω στο αρχείο  M\_1.31 .

**(4) Αποτέλεσμα δοκιμής:**

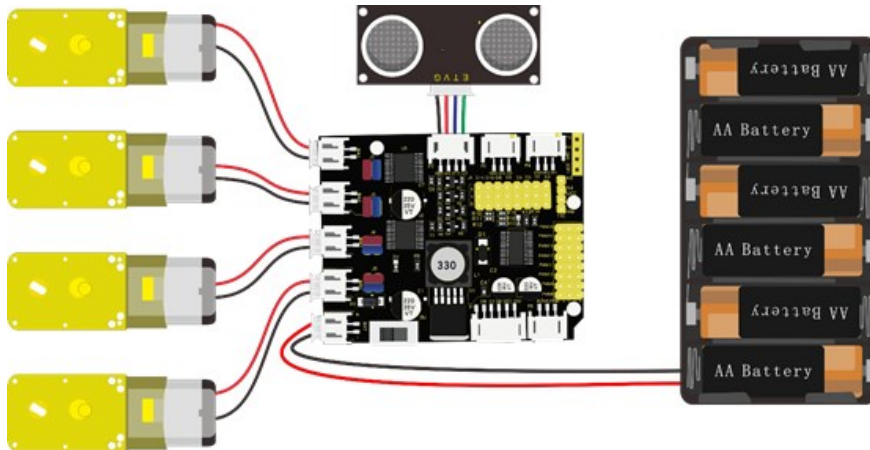
Ανεβάστε σωστά τον κώδικα χωρίς σφάλματα. Το ρομπότ R4 θα αρχίσει να κινείται σε ευθεία και όταν εντοπίσει το εμπόδιο, τότε θα χτυπήσει ο βομβητής και θα σταματήσει να κινείται.

## Δραστηριότητα 1.32: Έξυπνο αυτοκίνητο αποφυγής εμποδίων II

### (1) Περιγραφή:

Σε αυτή τη δραστηριότητα, σε συνέχεια της προηγούμενης, θα προγραμματίσετε το ρομπότ R4 να αποφεύγει το εμπόδιο χωρίς να σταματάει την κίνηση του. Πιο συγκεκριμένα όταν το αυτοκίνητο πλησιάσει το εμπόδιο σε απόσταση μικρότερη από 20 εκατοστά, τότε θα κινηθεί για λίγο προς τα πίσω και έπειτα θα στρίψει αριστερά και θα συνεχίσει τη κίνηση του.

### (2) Διάγραμμα καλωδίωσης:



Συνδέστε την πλακέτα οδήγησης στον υπολογιστή σας με το καλώδιο USB.

### (3) Κώδικας δοκιμής:

```

/*
 Δραστηριότητα 1.32
 Έξυπνο Αυτοκίνητο αποφυγής εμποδίων II
 */
#include <Wire.h>
#include <Adafruit_PWMServoDriver.h>
Adafruit_PWM_Servo_Driver pwm = Adafruit_PWM_Servo_Driver(0x47);
int BUZZER_PIN = 2; // Ορισμός βομβητή στο PIN 2
int speed = 2000; // Αρχικοποίηση της τιμής ταχύτητας
int turnSpeed = 2000; // Αρχικοποίηση της τιμής ταχύτητας για τις στροφές
int echoPin = 13; // Ορισμός μονάδα υπερήχων ECHO στο D13
int trigPin = 12; // Ορισμός μονάδα υπερήχων TRIG στο D12

int Ultrasonic_Ranging() {
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  int distance = pulseIn(echoPin, HIGH); // Ανάγνωση της διάρκειας των υψηλών επιπέδων
  distance = distance / 58; // Μετατροπή του χρόνου παλμού σε απόσταση
  delay(50);
  return distance;
}

```



```

}
void setup() {
  Serial.begin(9600);      // Έναρξη σειριακής θύρας στη συχνότητα 9600
  pwm.begin();
  pwm.setPWMFreq(60);
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  stop();
}

void loop() {
  int distance = Ultrasonic_Ranging();
  Serial.print("distance=");
  Serial.println(distance);
  if (distance < 30) { // Αν η απόσταση αντικειμένου από τον αισθητήρα είναι λιγότερη από 30
εκατοστά
    if (distance < 15) { // Αν η απόσταση αντικειμένου από τον αισθητήρα είναι λιγότερη από 15
εκατοστά
      digitalWrite(BUZZER_PIN, HIGH); // Ο βομβητής ενεργοποιείται
      stop();
      delay(100);
      digitalWrite(BUZZER_PIN, LOW); // Ο βομβητής απενεργοποιείται
    }else {
      stop();
      delay(100);
      turnL();
      delay(500);
    }
  }else {
    advance();
  }
}

void advance() // Κίνηση μπροστά
{
  pwm.setPWM(0, 0, speed);
  pwm.setPWM(1, 0, 0);
  pwm.setPWM(2, 0, speed);
  pwm.setPWM(3, 0, 0);
  pwm.setPWM(4, 0, speed);
  pwm.setPWM(5, 0, 0);
  pwm.setPWM(6, 0, speed);
  pwm.setPWM(7, 0, 0);
}

void turnR() // Στροφή δεξιά
{
  pwm.setPWM(0, 0, turnSpeed);
  pwm.setPWM(1, 0, 0);
  pwm.setPWM(2, 0, turnSpeed);
}

```

```

pwm.setPWM(3, 0, 0);
pwm.setPWM(4, 0, 0);
pwm.setPWM(5, 0, turnSpeed);
pwm.setPWM(6, 0, 0);
pwm.setPWM(7, 0, turnSpeed);
}

```

```

void turnL() // Στροφή αριστερά
{
pwm.setPWM(0, 0, 0);
pwm.setPWM(1, 0, turnSpeed);
pwm.setPWM(2, 0, 0);
pwm.setPWM(3, 0, turnSpeed);
pwm.setPWM(4, 0, turnSpeed);
pwm.setPWM(5, 0, 0);
pwm.setPWM(6, 0, turnSpeed);
pwm.setPWM(7, 0, 0);
}

```

```

void stop() // Σταματάει
{
pwm.setPWM(0, 0, 0);
pwm.setPWM(1, 0, 0);
pwm.setPWM(2, 0, 0);
pwm.setPWM(3, 0, 0);
pwm.setPWM(4, 0, 0);
pwm.setPWM(5, 0, 0);
pwm.setPWM(6, 0, 0);
pwm.setPWM(7, 0, 0);
}

```

```

void back() // Κίνηση προς τα πίσω
{
pwm.setPWM(0, 0, 0);
pwm.setPWM(1, 0, speed);
pwm.setPWM(2, 0, 0);
pwm.setPWM(3, 0, speed);
pwm.setPWM(4, 0, 0);
pwm.setPWM(5, 0, speed);
pwm.setPWM(6, 0, 0);
pwm.setPWM(7, 0, speed);
}

```


**Εναλλακτικά:** Μέσα στον φάκελο «ΚΩΔΙΚΕΣ ARDUINO» μπορείτε να ανοίξετε το αρχείο κάνοντας διπλό κλικ πάνω στο αρχείο  M\_1.32 .

(4) Αποτέλεσμα δοκιμής:


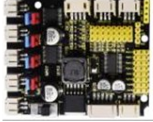




Ανεβάστε σωστά τον κώδικα χωρίς σφάλματα. Το ρομπότ R4 θα αποφύγει το εμπόδιο όταν το εντοπίσει καθώς κινείται.

## Δραστηριότητα 1.33: Έλεγχος έξυπνου αυτοκινήτου μέσω της εφαρμογής R4

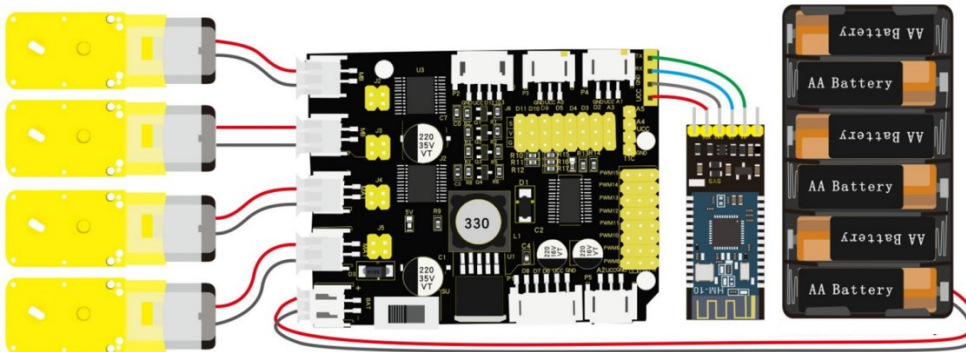
### (1) Περιγραφή:

Έχετε ήδη δει τη λειτουργία του Bluetooth. Σε αυτή τη δραστηριότητα, θα δείτε πως μπορείτε να προγραμματίσετε το αυτοκίνητο να ελέγχεται μέσω της εφαρμογής. Πιο συγκεκριμένα σε αυτή τη δραστηριότητα θα προγραμματίσετε τη μπροστινή κίνηση του ρομπότ με τη χρήση του πάνω βέλους της εφαρμογής .

### (2) Τι χρειάζεστε:

Ελεγκτής	Πλακέτα οδήγησης κινητήρων	Κινητήρας 4,5V 200rpm	Μονάδα Bluetooth	Βάση μπαταρίας	Καλώδιο USB
					

### (2) Διάγραμμα καλωδίωσης:



Τα VCC, GND, TXD και RXD της μονάδας Bluetooth συνδέονται με το VCC, το GNDRXD και το TXD της ασπίδας. Συνδέστε την πλακέτα οδήγησης στον υπολογιστή σας με το καλώδιο USB.

**ΣΗΜΕΙΩΣΗ:1.** Δώστε προσοχή στην κατεύθυνση των ακίδων κατά την εισαγωγή της μονάδας Bluetooth

**2. ΜΗΝ** εισάγετε τη μονάδα Bluetooth πριν ανεβάσετε τον κώδικα. (Για να αποφύγετε την αντίφαση μεταξύ σειριακής επικοινωνίας κώδικα και επικοινωνίας Bluetooth)

### (3) Κώδικας δοκιμής:

```
/*
 Δραστηριότητα 1.33
 Έλεγχος έξυπνου αυτοκινήτου μέσω της εφαρμογής R4
*/
```

```
#include <Wire.h>
#include <Adafruit_PWMServoDriver.h>
```

```

Adafruit_PWMServoDriver pwm = Adafruit_PWMServoDriver(0x47);
int speed = 2000; // Αρχικοποίηση της τιμής ταχύτητας
int turnSpeed = 2000; // Αρχικοποίηση της τιμής ταχύτητας για τις στροφές


void setup() {
  Serial.begin(9600); // Έναρξη σειριακής θύρας στη συχνότητα 9600
  pwm.begin();
  pwm.setPWMPFreq(60);
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  stop();
}

void loop() {
  if (Serial.available() > 0) {
    switch (Serial.read()) {
      case 'F': advance(); Serial.println("advance"); break; // Αν λάβει εντολή 'F', κάνει κίνηση μπροστά
      case 'S': stop(); Serial.println("stop"); break; // Αν λάβει εντολή 'S', σταματάει
      default : break;
    }
  }
}


void advance() // Κίνηση μπροστά
{
  pwm.setPWM(0, 0, speed);
  pwm.setPWM(1, 0, 0);
  pwm.setPWM(2, 0, speed);
  pwm.setPWM(3, 0, 0);
  pwm.setPWM(4, 0, speed);
  pwm.setPWM(5, 0, 0);
  pwm.setPWM(6, 0, speed);
  pwm.setPWM(7, 0, 0);
}

void stop() // Σταματάει
{
  pwm.setPWM(0, 0, 0);
  pwm.setPWM(1, 0, 0);
  pwm.setPWM(2, 0, 0);
  pwm.setPWM(3, 0, 0);
  pwm.setPWM(4, 0, 0);
  pwm.setPWM(5, 0, 0);
  pwm.setPWM(6, 0, 0);
  pwm.setPWM(7, 0, 0);
}

```


**Εναλλακτικά:** Μέσα στον φάκελο «ΚΩΔΙΚΕΣ ARDUINO» μπορείτε να ανοίξετε το αρχείο κάνοντας διπλό κλικ πάνω στο αρχείο  M\_1.33 .

(4) Αποτέλεσμα δοκιμής:


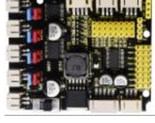




Ανεβάστε σωστά τον κώδικα χωρίς σφάλματα, έπειτα τοποθετήστε τη μονάδα Bluetooth και τρέξτε την εφαρμογή R4. Πατήστε το πάνω βέλος κίνησης του αυτοκινήτου για να κινήσετε το ρομπότ προς τα μπροστά .

## Δραστηριότητα 1.34: Έλεγχος έξυπνου αυτοκινήτου μέσω της εφαρμογής R4 II

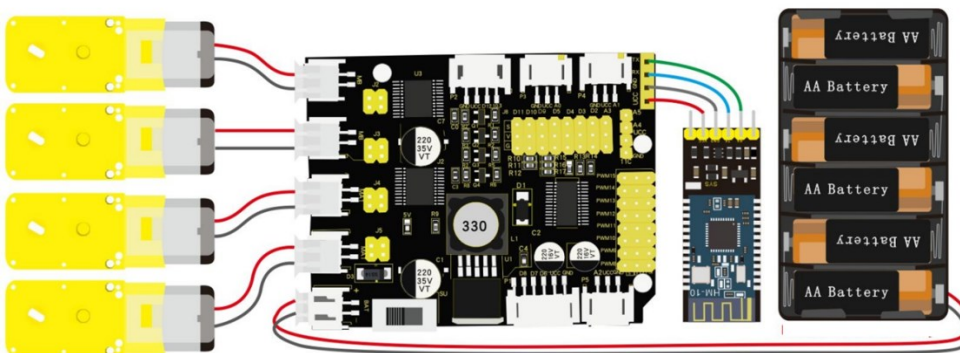
### (1) Περιγραφή:

Σε αυτή τη δραστηριότητα σε συνέχεια της προηγούμενης θα προσθέσετε τη πίσω κίνηση του αυτοκινήτου με τη χρήση του κάτω βέλους της εφαρμογής .

### (2) Τι χρειάζεστε:

Ελεγκτής	Πλακέτα οδήγησης κινητήρων	Κινητήρας 4,5V 200rpm	Μονάδα Bluetooth	Βάση μπαταρίας	Καλώδιο USB
					

### (2) Διάγραμμα καλωδίωσης:



Τα VCC, GND, TXD και RXD της μονάδας Bluetooth συνδέονται με το VCC, το GND, RXD και το TXD της ασπίδας. Συνδέστε την πλακέτα οδήγησης στον υπολογιστή σας με το καλώδιο USB.

**ΣΗΜΕΙΩΣΗ:1.** Δώστε προσοχή στην **κατεύθυνση** των ακίδων κατά την εισαγωγή της μονάδας Bluetooth

**2. ΜΗΝ** εισάγετε τη μονάδα Bluetooth πριν ανεβάσετε τον κώδικα. (Για να αποφύγετε την αντίφαση μεταξύ σειριακής επικοινωνίας κώδικα και επικοινωνίας Bluetooth)

### (3) Κώδικας δοκιμής:

/\*

Δραστηριότητα 1.34

Έλεγχος έξυπνου αυτοκινήτου μέσω της εφαρμογής R4 (2)

\*/

`#include <Wire.h>`

`#include <Adafruit_PWMServoDriver.h>`

```

Adafruit_PWMServoDriver pwm = Adafruit_PWMServoDriver(0x47);
int speed = 2000; // Αρχικοποίηση της τιμής ταχύτητας
int turnSpeed = 2000; // Αρχικοποίηση της τιμής ταχύτητας για τις στροφές

void setup() {
  Serial.begin(9600); // Έναρξη σειριακής θύρας στη συχνότητα 9600
  pwm.begin();
  pwm.setPWMPFreq(60);
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  stop();
}

void loop() {
  if (Serial.available() > 0) {
    switch (Serial.read()) {
      case 'F': advance(); Serial.println("advance"); break; // Αν λάβει εντολή 'F', κάνει κίνηση μπροστά
      case 'B': back(); Serial.println("back"); break; // Αν λάβει εντολή 'B', κάνει κίνηση πίσω
      case 'S': stop(); Serial.println("stop"); break; // Αν λάβει εντολή 'S', σταματάει
      default : break;
    }
  }
}


void advance() // Κίνηση μπροστά
{
  pwm.setPWM(0, 0, speed);
  pwm.setPWM(1, 0, 0);
  pwm.setPWM(2, 0, speed);
  pwm.setPWM(3, 0, 0);
  pwm.setPWM(4, 0, speed);
  pwm.setPWM(5, 0, 0);
  pwm.setPWM(6, 0, speed);
  pwm.setPWM(7, 0, 0);
}

void back() // Κίνηση προς τα πίσω
{
  pwm.setPWM(0, 0, 0);
  pwm.setPWM(1, 0, speed);
  pwm.setPWM(2, 0, 0);
  pwm.setPWM(3, 0, speed);
  pwm.setPWM(4, 0, 0);
  pwm.setPWM(5, 0, speed);
  pwm.setPWM(6, 0, 0);
  pwm.setPWM(7, 0, speed);
}



void stop() // Σταματάει
{

```

```
pwm.setPWM(0, 0, 0);  
pwm.setPWM(1, 0, 0);  
pwm.setPWM(2, 0, 0);  
pwm.setPWM(3, 0, 0);  
pwm.setPWM(4, 0, 0);  
pwm.setPWM(5, 0, 0);  
pwm.setPWM(6, 0, 0);  
pwm.setPWM(7, 0, 0);  
}
```


**Εναλλακτικά:** Μέσα στον φάκελο «ΚΩΔΙΚΕΣ ARDUINO» μπορείτε να ανοίξετε το αρχείο κάνοντας διπλό κλικ πάνω στο αρχείο  M\_1.34 .

#### (4) Αποτέλεσμα δοκιμής:


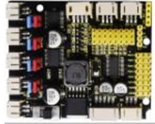




Ανεβάστε σωστά τον κώδικα χωρίς σφάλματα, έπειτα τοποθετήστε τη μονάδα Bluetooth και τρέξτε την εφαρμογή R4. Πατήστε το πάνω βέλος κίνησης του αυτοκινήτου για να κινήσετε το ρομπότ προς τα μπροστά  και το κάτω βέλος για το κινήσετε προς τα πίσω .

## Δραστηριότητα 1.35: Έλεγχος έξυπνου αυτοκινήτου μέσω της εφαρμογής R4 III

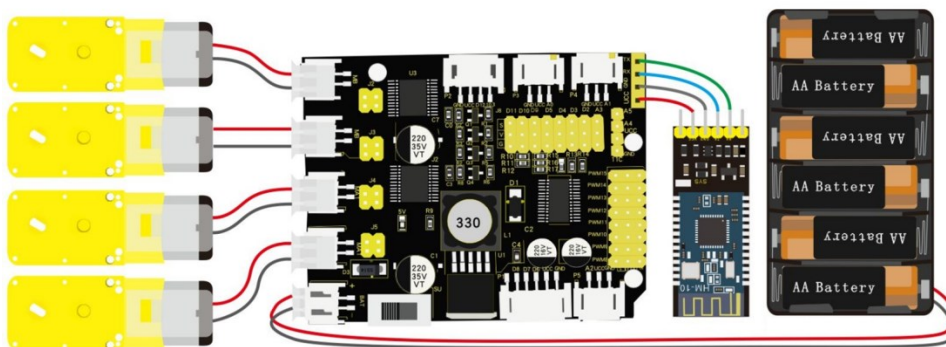
### (1) Περιγραφή:

Σε αυτή τη δραστηριότητα σε συνέχεια των δύο προηγούμενων δραστηριοτήτων, θα προσθέσετε τη δεξιά κίνηση του αυτοκινήτου με τη χρήση του δεξιού βέλους της εφαρμογής .

### (2) Τι χρειάζεστε:

Ελεγκτής	Πλακέτα οδήγησης κινητήρων	Κινητήρας 4,5V 200rpm	Μονάδα Bluetooth	Βάση μπαταρίας	Καλώδιο USB
					

### (2) Διάγραμμα καλωδίωσης:



Τα VCC, GND, TXD και RXD της μονάδας Bluetooth συνδέονται με το VCC, το GNDRXD και το TXD της ασπίδας.

Συνδέστε την πλακέτα οδήγησης στον υπολογιστή σας με το καλώδιο USB.

**ΣΗΜΕΙΩΣΗ:1.** Δώστε προσοχή στην **κατεύθυνση** των ακίδων κατά την εισαγωγή της μονάδας Bluetooth

**2. ΜΗΝ** εισάγετε τη μονάδα Bluetooth πριν ανεβάσετε τον κώδικα. (Για να αποφύγετε την αντίφαση μεταξύ σειριακής επικοινωνίας κώδικα και επικοινωνίας Bluetooth)



## (3) Κώδικας δοκιμής:

```

/*
 Δραστηριότητα 1.35
 Έλεγχος έξυπνου αυτοκινήτου μέσω της εφαρμογής R4 (3)
*/
#include <Wire.h>
#include <Adafruit_PWMServoDriver.h>
Adafruit_PWMServoDriver pwm = Adafruit_PWMServoDriver(0x47);
int speed = 2000; // Αρχικοποίηση της τιμής ταχύτητας
int turnSpeed = 2000; // Αρχικοποίηση της τιμής ταχύτητας για τις στροφές

void setup() {
  Serial.begin(9600); // Έναρξη σειριακής θύρας στη συχνότητα 9600
  pwm.begin();
  pwm.setPWMPFreq(60);
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  stop();
}

void loop() {
  if (Serial.available() > 0) {
    switch (Serial.read()) {
      case 'F': advance(); Serial.println("advance"); break; // Αν λάβει εντολή 'F', κάνει κίνηση μπροστά
      case 'R': turnR(); Serial.println("turn right"); break; // Αν λάβει εντολή 'R', στροφή δεξιά
      case 'S': stop(); Serial.println("stop"); break; // Αν λάβει εντολή 'S', σταματάει
      default : break;
    }
  }
}

void advance() // Κίνηση μπροστά
{
  pwm.setPWM(0, 0, speed);
  pwm.setPWM(1, 0, 0);
  pwm.setPWM(2, 0, speed);
  pwm.setPWM(3, 0, 0);
  pwm.setPWM(4, 0, speed);
  pwm.setPWM(5, 0, 0);
  pwm.setPWM(6, 0, speed);
  pwm.setPWM(7, 0, 0);
}

void turnR() // Στροφή δεξιά
{
  pwm.setPWM(0, 0, turnSpeed);
  pwm.setPWM(1, 0, 0);
  pwm.setPWM(2, 0, turnSpeed);
  pwm.setPWM(3, 0, 0);
}

```

```


pwm.setPWM(4, 0, 0);
pwm.setPWM(5, 0, turnSpeed);
pwm.setPWM(6, 0, 0);
pwm.setPWM(7, 0, turnSpeed);
}

```



```

void stop() // Σταματάει
{
  pwm.setPWM(0, 0, 0);
  pwm.setPWM(1, 0, 0);
  pwm.setPWM(2, 0, 0);
  pwm.setPWM(3, 0, 0);
  pwm.setPWM(4, 0, 0);
  pwm.setPWM(5, 0, 0);
  pwm.setPWM(6, 0, 0);
  pwm.setPWM(7, 0, 0);
}

```


**Εναλλακτικά:** Μέσα στον φάκελο «ΚΩΔΙΚΕΣ ARDUINO» μπορείτε να ανοίξετε το αρχείο κάνοντας διπλό κλικ πάνω στο αρχείο  M\_1.35 .

#### (4) Αποτέλεσμα δοκιμής:


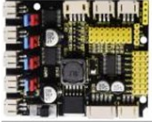




Ανεβάστε σωστά τον κώδικα χωρίς σφάλματα, έπειτα τοποθετήστε τη μονάδα Bluetooth και τρέξτε την εφαρμογή R4. Πατήστε το πάνω βέλος κίνησης του αυτοκινήτου για να κινήσετε το ρομπότ προς τα μπροστά  και προς το δεξί βέλος για να το κινήσετε προς τα δεξιά .

## Δραστηριότητα 1.36: Έλεγχος έξυπνου αυτοκινήτου μέσω της εφαρμογής R4 (4)

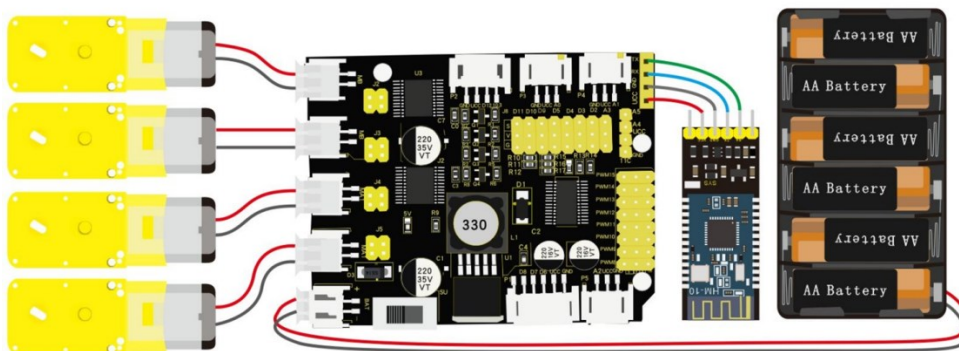
### (1) Περιγραφή:

Σε αυτή τη δραστηριότητα, σε συνέχεια των τριών προηγούμενων δραστηριοτήτων, θα προσθέσετε την αριστερή κίνηση του αυτοκινήτου με τη χρήση του αριστερού βέλους της εφαρμογής .

### (2) Τι χρειάζεστε:

Ελεγκτής	Πλακέτα οδήγησης κινητήρων	Κινητήρας 4,5V 200rpm	Μονάδα Bluetooth	Βάση μπαταρίας	Καλώδιο USB
					

### (2) Διάγραμμα καλωδίωσης:



Τα VCC, GND, TXD και RXD της μονάδας Bluetooth συνδέονται με το VCC, το GND, RXD και το TXD της ασπίδας. Συνδέστε την πλακέτα οδήγησης στον υπολογιστή σας με το καλώδιο USB.

**ΣΗΜΕΙΩΣΗ:1.** Δώστε προσοχή στην κατεύθυνση των ακίδων κατά την εισαγωγή της μονάδας Bluetooth

**2. ΜΗΝ** εισάγετε τη μονάδα Bluetooth πριν ανεβάσετε τον κώδικα. (Για να αποφύγετε την αντίφαση μεταξύ σειριακής επικοινωνίας κώδικα και επικοινωνίας Bluetooth)

**(3) Κώδικας δοκιμής:**

Δραστηριότητα 1.36

Έλεγχος έξυπνου αυτοκινήτου μέσω της εφαρμογής R4 (4)

```

*/
#include <Wire.h>
#include <Adafruit_PWMServoDriver.h>
Adafruit_PWMServoDriver pwm = Adafruit_PWMServoDriver(0x47);
int speed = 2000; // Αρχικοποίηση της τιμής ταχύτητας
int turnSpeed = 2000; // Αρχικοποίηση της τιμής ταχύτητας για τις στροφές

void setup() {
  Serial.begin(9600); // Έναρξη σειριακής θύρας στη συχνότητα 9600
  pwm.begin();
  pwm.setPWMPFreq(60);
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  stop();
}

void loop() {
  if (Serial.available() > 0) {
    switch (Serial.read()) {
      case 'F': advance(); Serial.println("advance"); break; // Αν λάβει εντολή 'F', κάνει κίνηση μπροστά
      case 'L': turnL(); Serial.println("turn left"); break; // Αν λάβει εντολή 'L', στροφή αριστερά
      case 'S': stop(); Serial.println("stop"); break; // Αν λάβει εντολή 'S', σταματάει
      default : break;
    }
  }
}


void advance() // Κίνηση μπροστά
{
  pwm.setPWM(0, 0, speed);
  pwm.setPWM(1, 0, 0);
  pwm.setPWM(2, 0, speed);
  pwm.setPWM(3, 0, 0);
  pwm.setPWM(4, 0, speed);
  pwm.setPWM(5, 0, 0);
  pwm.setPWM(6, 0, speed);
  pwm.setPWM(7, 0, 0);
}

void turnL() // Στροφή αριστερά
{
  pwm.setPWM(0, 0, 0);
  pwm.setPWM(1, 0, turnSpeed);
  pwm.setPWM(2, 0, 0);
  pwm.setPWM(3, 0, turnSpeed);
  pwm.setPWM(4, 0, turnSpeed);
  pwm.setPWM(5, 0, 0);
}



```

```
pwm.setPWM(6, 0, turnSpeed);  
pwm.setPWM(7, 0, 0);  
}
```

```
void stop() // Σταματάει  
{  
  pwm.setPWM(0, 0, 0);  
  pwm.setPWM(1, 0, 0);  
  pwm.setPWM(2, 0, 0);  
  pwm.setPWM(3, 0, 0);  
  pwm.setPWM(4, 0, 0);  
  pwm.setPWM(5, 0, 0);  
  pwm.setPWM(6, 0, 0);  
  pwm.setPWM(7, 0, 0);  
}
```

**Εναλλακτικά:** Μέσα στον φάκελο «ΚΩΔΙΚΕΣ ARDUINO» μπορείτε να ανοίξετε το αρχείο κάνοντας διπλό κλικ πάνω στο αρχείο  M\_1.36 .

#### (4) Αποτέλεσμα δοκιμής:


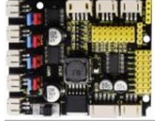




Ανεβάστε σωστά τον κώδικα χωρίς σφάλματα, έπειτα τοποθετήστε τη μονάδα Bluetooth και τρέξτε την εφαρμογή R4. Πατήστε το πάνω βέλος κίνησης του αυτοκινήτου για να κινήσετε το ρομπότ προς τα μπροστά  και το αριστερό βέλος για να το κινήσετε προς τα αριστερά .

## Δραστηριότητα 1.37: Έλεγχος έξυπνου αυτοκινήτου μέσω της εφαρμογής R4 (5)

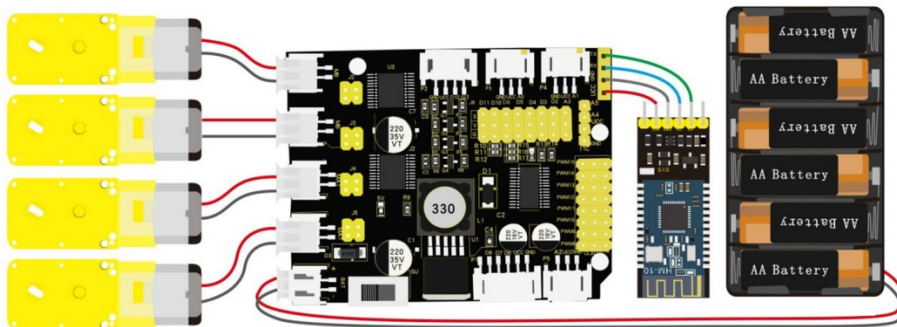
### (1) Περιγραφή:

Σε αυτή τη δραστηριότητα, σε συνέχεια των προηγούμενων, θα προσθέσετε όλες τις κινήσεις του ρομπότ και θα μπορείτε να χρησιμοποιήσετε όλα τα βέλη της εφαρμογής και να οδηγήσετε το ρομπότ προς οποιαδήποτε κατεύθυνση θέλετε.

### (2) Τι χρειάζεστε:

Ελεγκτής	Πλακέτα οδήγησης κινητήρων	Κινητήρας 4,5V 200rpm	Μονάδα Bluetooth	Βάση μπαταρίας	Καλώδιο USB
					

### (3) Διάγραμμα καλωδίωσης:



Τα VCC, GND, TXD και RXD της μονάδας Bluetooth συνδέονται με το VCC, το GND, TXD και το RXD της ασπίδας. Συνδέστε την πλακέτα οδήγησης στον υπολογιστή σας με το καλώδιο USB.

**ΣΗΜΕΙΩΣΗ:1.** Δώστε προσοχή στην **κατεύθυνση** των ακίδων κατά την εισαγωγή της μονάδας Bluetooth

**2. ΜΗΝ** εισάγετε τη μονάδα Bluetooth πριν ανεβάσετε τον κώδικα. (Για να αποφύγετε την αντίφαση μεταξύ σειριακής επικοινωνίας κώδικα και επικοινωνίας Bluetooth)

### (4) Κώδικας δοκιμής:

/\*

Δραστηριότητα 1.37

Έλεγχος έξυπνου αυτοκινήτου μέσω της εφαρμογής R4 (5)

\*/

```
#include <Wire.h>
```

```
#include <Adafruit_PWMServoDriver.h>
```

```
Adafruit_PWMServoDriver pwm = Adafruit_PWMServoDriver(0x47);
```

```
int speed = 2000; // Αρχικοποίηση της τιμής ταχύτητας
```

```
int turnSpeed = 2000; // Αρχικοποίηση της τιμής ταχύτητας για τις στροφές
```

```
void setup() {
```

```
133
```

```

Serial.begin(9600);    // Έναρξη σειριακής θύρας στη συχνότητα 9600
pwm.begin();
pwm.setPWMFreq(60);
pinMode(trigPin, OUTPUT);
pinMode(echoPin, INPUT);
stop();
}

void loop() {
  if (Serial.available() > 0) {
    switch (Serial.read()) {
      case 'F': advance(); Serial.println("advance"); break; // Αν λάβει εντολή 'F', κάνει κίνηση μπροστά
      case 'B': back(); Serial.println("back"); break; // Αν λάβει εντολή 'B', κάνει κίνηση προς τα πίσω
      case 'L': turnL(); Serial.println("turn left"); break; // Αν λάβει εντολή 'L', στροφή αριστερά
      case 'R': turnR(); Serial.println("turn right"); break; // Αν λάβει εντολή 'R', στροφή δεξιά
      case 'S': stop(); Serial.println("stop"); break; // Αν λάβει εντολή 'S', σταματάει
      default : break;
    }
  }
}

void advance() // Κίνηση μπροστά
{
  pwm.setPWM(0, 0, speed);
  pwm.setPWM(1, 0, 0);
  pwm.setPWM(2, 0, speed);
  pwm.setPWM(3, 0, 0);
  pwm.setPWM(4, 0, speed);
  pwm.setPWM(5, 0, 0);
  pwm.setPWM(6, 0, speed);
  pwm.setPWM(7, 0, 0);
}

void turnR() // Στροφή δεξιά
{
  pwm.setPWM(0, 0, turnSpeed);
  pwm.setPWM(1, 0, 0);
  pwm.setPWM(2, 0, turnSpeed);
  pwm.setPWM(3, 0, 0);
  pwm.setPWM(4, 0, 0);
  pwm.setPWM(5, 0, turnSpeed);
  pwm.setPWM(6, 0, 0);
  pwm.setPWM(7, 0, turnSpeed);
}

void turnL() // Στροφή αριστερά
{
  pwm.setPWM(0, 0, 0);
  pwm.setPWM(1, 0, turnSpeed);
  pwm.setPWM(2, 0, 0);
}

```

```

pwm.setPWM(3, 0, turnSpeed);
pwm.setPWM(4, 0, turnSpeed);
pwm.setPWM(5, 0, 0);
pwm.setPWM(6, 0, turnSpeed);
pwm.setPWM(7, 0, 0);
}

```

```

void stop() // Σταματάει
{
pwm.setPWM(0, 0, 0);
pwm.setPWM(1, 0, 0);
pwm.setPWM(2, 0, 0);
pwm.setPWM(3, 0, 0);
pwm.setPWM(4, 0, 0);
pwm.setPWM(5, 0, 0);
pwm.setPWM(6, 0, 0);
pwm.setPWM(7, 0, 0);
}

```

```

void back() // Κίνηση προς τα πίσω
{
pwm.setPWM(0, 0, 0);
pwm.setPWM(1, 0, speed);
pwm.setPWM(2, 0, 0);
pwm.setPWM(3, 0, speed);
pwm.setPWM(4, 0, 0);
pwm.setPWM(5, 0, speed);
pwm.setPWM(6, 0, 0);
pwm.setPWM(7, 0, speed);
}

```

**Εναλλακτικά:** Μέσα στον φάκελο «ΚΩΔΙΚΕΣ ARDUINO» μπορείτε να ανοίξετε το αρχείο κάνοντας διπλό κλικ πάνω στο αρχείο  M\_1.37 .

#### (5) Αποτέλεσμα δοκιμής:

Ανεβάστε σωστά τον κώδικα χωρίς σφάλματα, έπειτα τοποθετήστε τη μονάδα Bluetooth και τρέξτε την εφαρμογή R4. Κινείστε το ρομπότ χρησιμοποιώντας τα βέλη κατεύθυνσης.











## Δραστηριότητα 1.38: Έλεγχος αυτοκινήτου με το τηλεχειριστήριο IR

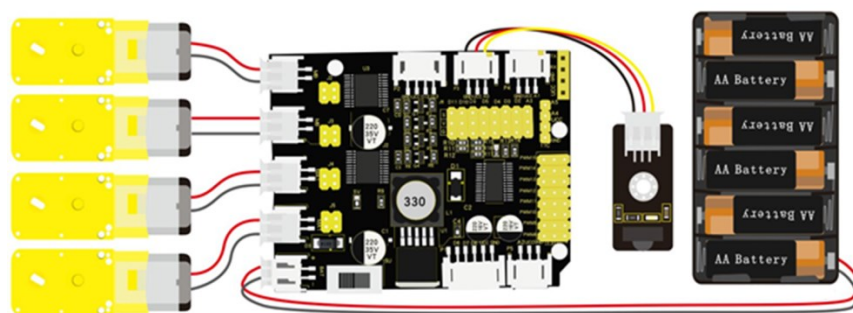
### (1) Περιγραφή:

Το τηλεχειριστήριο IR εφαρμόζεται ευρέως. Σε προηγούμενη δραστηριότητα μάθατε να ελέγχετε διάφορες συσκευές μέσω του τηλεχειριστηρίου IR όπως το LED και τον βομβητή. Σε αυτή τη δραστηριότητα και στις επόμενες θα μάθετε να προγραμματίζετε το ρομπότ έτσι ώστε να μπορείτε να το οδηγήσετε με τη χρήση του τηλεχειριστηρίου IR και των βελών κατεύθυνσης του. Πιο συγκεκριμένα σε αυτή τη δραστηριότητά θα προγραμματίσετε τη μπροστινή κίνηση του ρομπότ με τη χρήση του πάνω βέλους του χειριστηρίου.

### (2) Τι χρειάζεστε:

Ελεγκτής	Πλακέτα οδήγησης κινητήρων	Βάση μπαταρίας	Κινητήρας 4,5V 200rpm
			
Δέκτης υπερέυθρων	Καλώδιο USB	3pin F-F Dupont Line	
			

### (3) Διάγραμμα καλωδίωσης:



Συνδέστε την πλακέτα οδήγησης στον υπολογιστή σας με το καλώδιο USB.

## (4) Κώδικας δοκιμής:

```

/*
 Δραστηριότητα 1.38
 Έλεγχος αυτοκινήτου με το τηλεχειριστήριο IR
*/
#include <Wire.h>
#include <Adafruit_PWMServoDriver.h>
#include <IRremote.h>
Adafruit_PWMServoDriver pwm = Adafruit_PWMServoDriver(0x47);
int RECV_PIN = A0;
int speed = 2000; // Αρχικοποίηση της τιμής ταχύτητας

#define IR_Go 70
#define IR_Back 21
#define IR_Left 68
#define IR_Right 67
#define IR_Stop 64


void setup() {
  Serial.begin(9600);
  pwm.begin();
  pwm.setPWMPFreq(60);
  IrReceiver.begin(RECV_PIN); // ενεργοποίηση του δέκτη IR
  stop();
}

void loop() {
  if (IrReceiver.decode()) {
    switch (IrReceiver.decodedIRData.command) {
      case IR_Go: advance(); // Όταν πατηθεί το πάνω κουμπί, το ρομπότ κινείται προς τα μπρος
        break;
      case IR_Stop: stop(); // Όταν πατηθεί το 'OK' κουμπί, το ρομπότ σταματάει.
        break;
      default: break;
    }
    IrReceiver.resume();
  }
}

void advance() // Κίνηση μπροστά
{
  pwm.setPWM(0, 0, speed);
  pwm.setPWM(1, 0, 0);
  pwm.setPWM(2, 0, speed);
  pwm.setPWM(3, 0, 0);
  pwm.setPWM(4, 0, speed);
  pwm.setPWM(5, 0, 0);
  pwm.setPWM(6, 0, speed);
  pwm.setPWM(7, 0, 0);
}

```

```
void stop() // Σταματάει  
{  
  pwm.setPWM(0, 0, 0);  
  pwm.setPWM(1, 0, 0);  
  pwm.setPWM(2, 0, 0);  
  pwm.setPWM(3, 0, 0);  
  pwm.setPWM(4, 0, 0);  
  pwm.setPWM(5, 0, 0);  
  pwm.setPWM(6, 0, 0);  
  pwm.setPWM(7, 0, 0);  
}
```

**Εναλλακτικά:** Μέσα στον φάκελο «ΚΩΔΙΚΕΣ ARDUINO» μπορείτε να ανοίξετε το αρχείο κάνοντας διπλό κλικ πάνω στο αρχείο  M\_1.38 .

#### (5) Αποτέλεσμα δοκιμής:


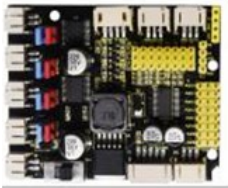





Ανεβάστε σωστά τον κώδικα χωρίς σφάλματα. Πατήστε το πάνω βέλος του χειριστηρίου για να κινήσετε προς τα μπροστά το ρομπότ.

## Δραστηριότητα 1.39: Έλεγχος αυτοκινήτου με το τηλεχειριστήριο IR (2)

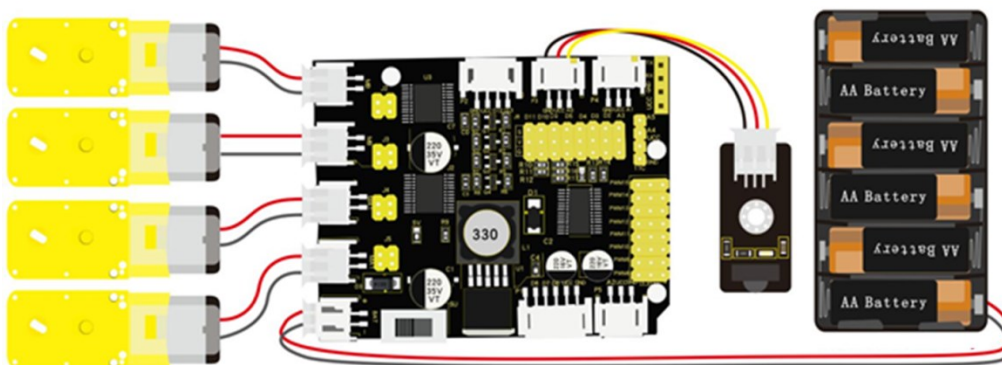
### (1) Περιγραφή:

Σε αυτή τη δραστηριότητα, σε συνέχεια της προηγούμενης, θα προσθέσετε την πίσω κίνηση του ρομπότ με τη χρήση του κάτω βέλους κίνησης του τηλεχειριστηρίου IR.

### (2) Τι χρειάζεστε:

Ελεγκτής	Πλακέτα οδήγησης κινητήρων	Βάση μπαταρίας	Κινητήρας 4,5V 200rpm
			
Δέκτης υπεράυθρων	Καλώδιο USB	3pin F-F Dupont Line	
			

### (3) Διάγραμμα καλωδίωσης:



Συνδέστε την πλακέτα οδήγησης στον υπολογιστή σας με το καλώδιο USB.

## (4) Κώδικας δοκιμής:

```

/*
 Δραστηριότητα 1.39
 Έλεγχος αυτοκινήτου με το τηλεχειριστήριο IR (2)
*/
#include <Wire.h>
#include <Adafruit_PWMServoDriver.h>
#include <IRremote.h>
Adafruit_PWMServoDriver pwm = Adafruit_PWMServoDriver(0x47);
int RECV_PIN = A0;
int speed = 2000; // Αρχικοποίηση της τιμής ταχύτητας

#define IR_Go 70
#define IR_Back 21
#define IR_Left 68
#define IR_Right 67
#define IR_Stop 64

void setup() {
  Serial.begin(9600);
  pwm.begin();
  pwm.setPWMPFreq(60);
  IrReceiver.begin(RECV_PIN); // ενεργοποίηση του δέκτη IR
  stop();
}

void loop() {
  if (IrReceiver.decode()) {
    switch (IrReceiver.decodedIRData.command) {
      case IR_Go: advance(); // Όταν πατηθεί το πάνω κουμπί, το ρομπότ κινείται προς τα μπρος
        break;
      case IR_Back: back(); // Όταν πατηθεί το κάτω κουμπί, το ρομπότ κινείται προς τα πίσω
        break;
      case IR_Stop: stop(); // Όταν πατηθεί το 'OK' κουμπί, το ρομπότ σταματάει.
        break;
      default: break;
    }
    IrReceiver.resume();
  }
}

void advance() // Κίνηση μπροστά
{
  pwm.setPWM(0, 0, speed);
  pwm.setPWM(1, 0, 0);
  pwm.setPWM(2, 0, speed);
  pwm.setPWM(3, 0, 0);
  pwm.setPWM(4, 0, speed);
  pwm.setPWM(5, 0, 0);
}

```

```

    pwm.setPWM(6, 0, speed);
    pwm.setPWM(7, 0, 0);
}

void stop() // Σταματάει
{
    pwm.setPWM(0, 0, 0);
    pwm.setPWM(1, 0, 0);
    pwm.setPWM(2, 0, 0);
    pwm.setPWM(3, 0, 0);
    pwm.setPWM(4, 0, 0);
    pwm.setPWM(5, 0, 0);
    pwm.setPWM(6, 0, 0);
    pwm.setPWM(7, 0, 0);
}

void back() // Κίνηση προς τα πίσω
{
    pwm.setPWM(0, 0, 0);
    pwm.setPWM(1, 0, speed);
    pwm.setPWM(2, 0, 0);
    pwm.setPWM(3, 0, speed);
    pwm.setPWM(4, 0, 0);
    pwm.setPWM(5, 0, speed);
    pwm.setPWM(6, 0, 0);
    pwm.setPWM(7, 0, speed);
}

```

**Εναλλακτικά:** Μέσα στον φάκελο «ΚΩΔΙΚΕΣ ARDUINO» μπορείτε να ανοίξετε το αρχείο κάνοντας διπλό κλικ πάνω στο αρχείο  M\_1.39 .

#### (5) Αποτέλεσμα δοκιμής:








Ανεβάστε σωστά τον κώδικα χωρίς σφάλματα. Πατήστε το πάνω και κάτω βέλος του χειριστηρίου για να κινήσετε προς τα μπροστά και προς τα πίσω το ρομπότ.

## Δραστηριότητα 1.40: Έλεγχος αυτοκινήτου με το τηλεχειριστήριο IR (3)

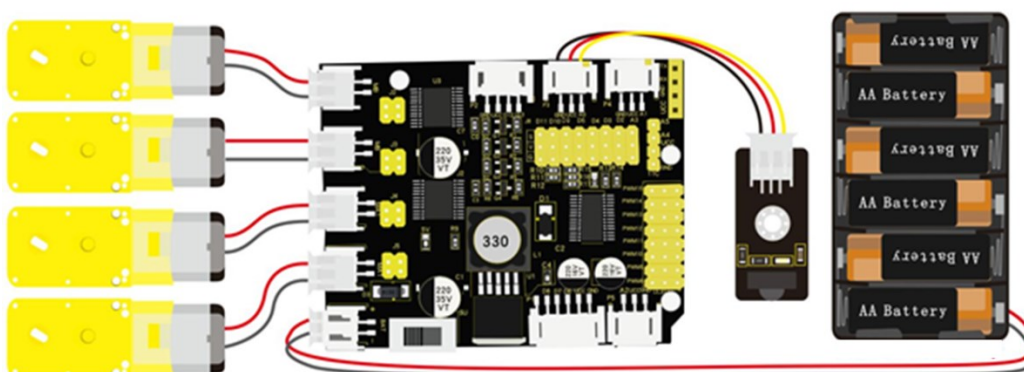
### (1) Περιγραφή:

Σε αυτή τη δραστηριότητα, σε συνέχεια της Δραστηριότητα 8.38: Έλεγχος αυτοκινήτου με το τηλεχειριστήριο IR, θα προσθέσετε την δεξιά κίνηση του ρομπότ με τη χρήση του δεξιού βέλους κίνησης του τηλεχειριστηρίου IR.

### (2) Τι χρειάζεστε:

Ελεγκτής	Πλακέτα οδήγησης κινητήρων	Βάση μπαταρίας	Κινητήρας 4,5V 200rpm
			
Δέκτης υπεράυθρων	Καλώδιο USB	3pin F-F Dupont Line	
			

### (3) Διάγραμμα καλωδίωσης:



Συνδέστε την πλακέτα οδήγησης στον υπολογιστή σας με το καλώδιο USB.

**(4) Κώδικας δοκιμής:**

```

/*
 Δραστηριότητα 1.40
 Έλεγχος αυτοκινήτου με το τηλεχειριστήριο IR (3)
*/
#include <Wire.h>
#include <Adafruit_PWMServoDriver.h>
#include <IRremote.h>
Adafruit_PWMServoDriver pwm = Adafruit_PWMServoDriver(0x47);
int RECV_PIN = A0;
int speed = 2000; // Αρχικοποίηση της τιμής ταχύτητας
int turnSpeed = 2000; // Αρχικοποίηση της τιμής ταχύτητας για τις στροφές

#define IR_Go 70
#define IR_Back 21
#define IR_Left 68
#define IR_Right 67
#define IR_Stop 64

void setup() {
  Serial.begin(9600);
  pwm.begin();
  pwm.setPWMPFreq(60);
  IrReceiver.begin(RECV_PIN); // ενεργοποίηση του δέκτη IR
  stop();
}

void loop() {
  if (IrReceiver.decode()) {
    switch (IrReceiver.decodedIRData.command) {
      case IR_Go: advance(); // Όταν πατηθεί το πάνω κουμπί, το ρομπότ κινείται προς τα μπρος
        break;
      case IR_Right: turnR(); // Όταν πατηθεί το δεξί κουμπί, το ρομπότ στρίβει προς τα δεξιά
        break;
      case IR_Stop: stop(); // Όταν πατηθεί το 'OK' κουμπί, το ρομπότ σταματάει.
        break;
      default: break;
    }
    IrReceiver.resume();
  }
}

void advance() // Κίνηση μπροστά
{
  pwm.setPWM(0, 0, speed);
  pwm.setPWM(1, 0, 0);
  pwm.setPWM(2, 0, speed);
  pwm.setPWM(3, 0, 0);
  pwm.setPWM(4, 0, speed);
  pwm.setPWM(5, 0, 0);
}

```




```

    pwm.setPWM(6, 0, speed);
    pwm.setPWM(7, 0, 0);
}
void turnR() // Στροφή δεξιά
{
    pwm.setPWM(0, 0, turnSpeed);
    pwm.setPWM(1, 0, 0);
    pwm.setPWM(2, 0, turnSpeed);
    pwm.setPWM(3, 0, 0);
    pwm.setPWM(4, 0, 0);
    pwm.setPWM(5, 0, turnSpeed);
    pwm.setPWM(6, 0, 0);
    pwm.setPWM(7, 0, turnSpeed);
}

void stop() // Σταματάει
{
    pwm.setPWM(0, 0, 0);
    pwm.setPWM(1, 0, 0);
    pwm.setPWM(2, 0, 0);
    pwm.setPWM(3, 0, 0);
    pwm.setPWM(4, 0, 0);
    pwm.setPWM(5, 0, 0);
    pwm.setPWM(6, 0, 0);
    pwm.setPWM(7, 0, 0);
}

```

**Εναλλακτικά:** Μέσα στον φάκελο «ΚΩΔΙΚΕΣ ARDUINO» μπορείτε να ανοίξετε το αρχείο κάνοντας διπλό κλικ πάνω στο αρχείο  M\_1.40 .

#### (5) Αποτέλεσμα δοκιμής:




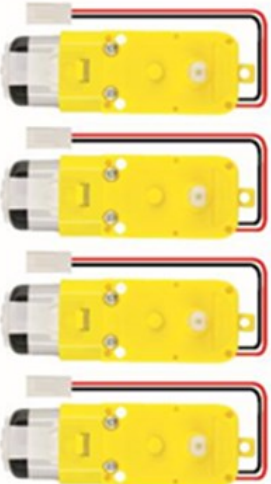



Ανεβάστε σωστά τον κώδικα χωρίς σφάλματα. Πατήστε το πάνω και δεξί βέλος του χειριστηρίου IR για να κινήσετε προς τα μπροστά και προς τα δεξιά το ρομπότ.

## Δραστηριότητα 1.41: Έλεγχος αυτοκινήτου με το τηλεχειριστήριο IR (4)

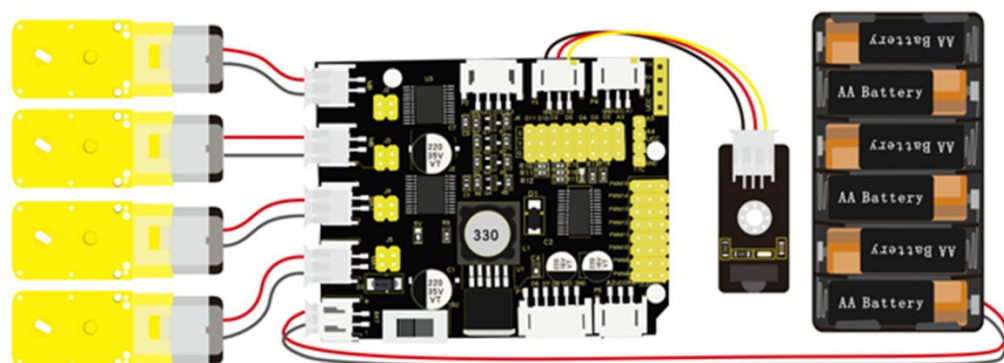
### (1) Περιγραφή:

Σε αυτή τη δραστηριότητα, σε συνέχεια της Δραστηριότητα 8.38: Έλεγχος αυτοκινήτου με το τηλεχειριστήριο IR, θα προσθέσετε την αριστερή κίνηση του ρομπότ με τη χρήση του αριστερού βέλους κίνησης του τηλεχειριστηρίου IR.

### (2) Τι χρειάζεστε:

Ελεγκτής	Πλακέτα οδήγησης κινητήρων	Βάση μπαταρίας	Κινητήρας 4,5V 200rpm
			
Δέκτης υπεράυθρων	Καλώδιο USB	3pin F-F Dupont Line	
			

### (3) Διάγραμμα καλωδίωσης:



Συνδέστε την πλακέτα οδήγησης στον υπολογιστή σας με το καλώδιο USB.

## (4) Κώδικας δοκιμής:

```

/*
 Δραστηριότητα 1.41
 Έλεγχος αυτοκινήτου με το τηλεχειριστήριο IR (4)
*/
#include <Wire.h>
#include <Adafruit_PWMServoDriver.h>
#include <IRremote.h>
Adafruit_PWMServoDriver pwm = Adafruit_PWMServoDriver(0x47);
int RECV_PIN = A0;
int speed = 2000; // Αρχικοποίηση της τιμής ταχύτητας
int turnSpeed = 2000; // Αρχικοποίηση της τιμής ταχύτητας για τις στροφές

#define IR_Go 70
#define IR_Back 21
#define IR_Left 68
#define IR_Right 67
#define IR_Stop 64

void setup() {
  Serial.begin(9600);
  pwm.begin();
  pwm.setPWMPFreq(60);
  IrReceiver.begin(RECV_PIN); // ενεργοποίηση του δέκτη IR
  stop();
}

void loop() {
  if (IrReceiver.decode()) {
    switch (IrReceiver.decodedIRData.command) {
      case IR_Go: advance(); // Όταν πατηθεί το πάνω κουμπί, το ρομπότ κινείται προς τα μπρος
        break;
      case IR_Left: turnL(); // Όταν πατηθεί το αριστερό κουμπί, το ρομπότ στρίβει προς τα αριστερά
        break;
      case IR_Stop: stop(); // Όταν πατηθεί το 'OK' κουμπί, το ρομπότ σταματάει.
        break;
      default: break;
    }
    IrReceiver.resume();
  }
}

void advance() // Κίνηση μπροστά
{
  pwm.setPWM(0, 0, speed);
  pwm.setPWM(1, 0, 0);
  pwm.setPWM(2, 0, speed);
  pwm.setPWM(3, 0, 0);
  pwm.setPWM(4, 0, speed);
}

```


```

    pwm.setPWM(5, 0, 0);
    pwm.setPWM(6, 0, speed);
    pwm.setPWM(7, 0, 0);
}

void turnL() // Στροφή αριστερά
{
    pwm.setPWM(0, 0, 0);
    pwm.setPWM(1, 0, turnSpeed);
    pwm.setPWM(2, 0, 0);
    pwm.setPWM(3, 0, turnSpeed);
    pwm.setPWM(4, 0, turnSpeed);
    pwm.setPWM(5, 0, 0);
    pwm.setPWM(6, 0, turnSpeed);
    pwm.setPWM(7, 0, 0);
}

void stop() // Σταματάει
{
    pwm.setPWM(0, 0, 0);
    pwm.setPWM(1, 0, 0);
    pwm.setPWM(2, 0, 0);
    pwm.setPWM(3, 0, 0);
    pwm.setPWM(4, 0, 0);
    pwm.setPWM(5, 0, 0);
    pwm.setPWM(6, 0, 0);
    pwm.setPWM(7, 0, 0);
}

```

**Εναλλακτικά:** Μέσα στον φάκελο «ΚΩΔΙΚΕΣ ARDUINO» μπορείτε να ανοίξετε το αρχείο κάνοντας διπλό κλικ πάνω στο αρχείο  M\_1.41 .

#### (5) Αποτέλεσμα δοκιμής:


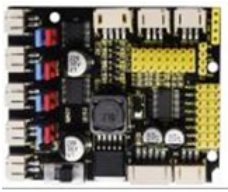

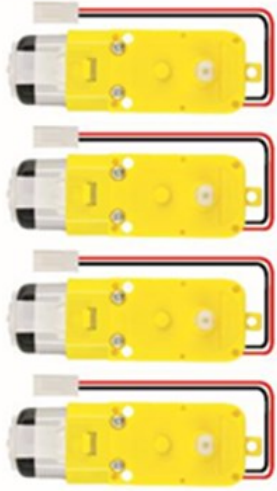



Ανεβάστε σωστά τον κώδικα χωρίς σφάλματα. Πατήστε το πάνω και αριστερό βέλος του χειριστηρίου IR για να κινήσετε προς τα μπροστά και προς τα αριστερά το ρομπότ.

## Δραστηριότητα 1.42: Έλεγχος αυτοκινήτου με το τηλεχειριστήριο IR (5)

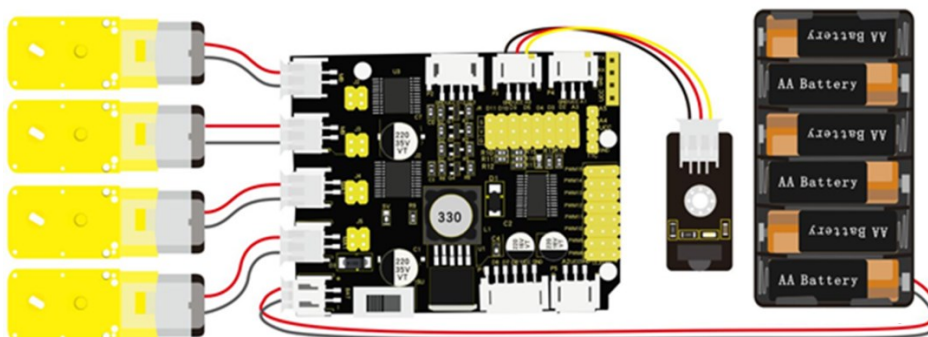
### (1) Περιγραφή:

Σε αυτή τη δραστηριότητα, σε συνέχεια των προηγούμενων, θα προγραμματίσετε το ρομπότ να κινείται προς όλες τις κατευθύνσεις με τη χρήση του τηλεχειριστηρίου IR. Έτσι θα μπορείτε να οδηγήσετε το ρομπότ προς οποιαδήποτε κατεύθυνση θέλετε χρησιμοποιώντας τα βέλη κατεύθυνσης του τηλεχειριστηρίου IR.

### (2) Τι χρειάζεστε:

Ελεγκτής	Πλακέτα οδήγησης κινητήρων	Βάση μπαταρίας	Κινητήρας 4,5V 200rpm
			
Δέκτης υπεράυθρων	Καλώδιο USB	3pin F-F Dupont Line	
			

### (3) Διάγραμμα καλωδίωσης:



Συνδέστε την πλακέτα οδήγησης στον υπολογιστή σας με το καλώδιο USB.

## (4) Κώδικας δοκιμής:

```

/*
 Δραστηριότητα 1.42
 Έλεγχος αυτοκινήτου με το τηλεχειριστήριο IR (5)
*/
#include <Wire.h>
#include <Adafruit_PWMServoDriver.h>
#include <IRremote.h>
Adafruit_PWMServoDriver pwm = Adafruit_PWMServoDriver(0x47);
int RECV_PIN = A0;
int speed = 2000; // Αρχικοποίηση της τιμής ταχύτητας
int turnSpeed = 2000; // Αρχικοποίηση της τιμής ταχύτητας για τις στροφές

#define IR_Go 70
#define IR_Back 21
#define IR_Left 68
#define IR_Right 67
#define IR_Stop 64

void setup() {
  Serial.begin(9600);
  pwm.begin();
  pwm.setPWMPFreq(60);
  IrReceiver.begin(RECV_PIN); // ενεργοποίηση του δέκτη IR
  stop();
}

void loop() {
  if (IrReceiver.decode()) {
    switch (IrReceiver.decodedIRData.command) {
      case IR_Go: advance(); // Όταν πατηθεί το πάνω κουμπί, το ρομπότ κινείται προς τα μπρος
        break;
      case IR_Back: back(); // Όταν πατηθεί το κάτω κουμπί, το ρομπότ κινείται προς τα πίσω
        break;
      case IR_Left: turnL(); // Όταν πατηθεί το αριστερό κουμπί, το ρομπότ στρίβει προς τα αριστερά
        break;
      case IR_Right: turnR(); // Όταν πατηθεί το δεξί κουμπί, το ρομπότ κινείται προς τα δεξιά
        break;
      case IR_Stop: stop(); // Όταν πατηθεί το 'OK' κουμπί, το ρομπότ σταματάει.
        break;
      default: break;
    }
    IrReceiver.resume();
  }
}

void advance() // Κίνηση μπροστά
{
  pwm.setPWM(0, 0, speed);
  pwm.setPWM(1, 0, 0);
}

```

```

    pwm.setPWM(2, 0, speed);
    pwm.setPWM(3, 0, 0);
    pwm.setPWM(4, 0, speed);
    pwm.setPWM(5, 0, 0);
    pwm.setPWM(6, 0, speed);
    pwm.setPWM(7, 0, 0);
}

```

```

void turnR() // Στροφή δεξιά
{
    pwm.setPWM(0, 0, turnSpeed);
    pwm.setPWM(1, 0, 0);
    pwm.setPWM(2, 0, turnSpeed);
    pwm.setPWM(3, 0, 0);
    pwm.setPWM(4, 0, 0);
    pwm.setPWM(5, 0, turnSpeed);
    pwm.setPWM(6, 0, 0);
    pwm.setPWM(7, 0, turnSpeed);
}

```

```

void turnL() // Στροφή αριστερά
{
    pwm.setPWM(0, 0, 0);
    pwm.setPWM(1, 0, turnSpeed);
    pwm.setPWM(2, 0, 0);
    pwm.setPWM(3, 0, turnSpeed);
    pwm.setPWM(4, 0, turnSpeed);
    pwm.setPWM(5, 0, 0);
    pwm.setPWM(6, 0, turnSpeed);
    pwm.setPWM(7, 0, 0);
}

```

```

void stop() // Σταματάει
{
    pwm.setPWM(0, 0, 0);
    pwm.setPWM(1, 0, 0);
    pwm.setPWM(2, 0, 0);
    pwm.setPWM(3, 0, 0);
    pwm.setPWM(4, 0, 0);
    pwm.setPWM(5, 0, 0);
    pwm.setPWM(6, 0, 0);
    pwm.setPWM(7, 0, 0);
}


```

```

void back() // Κίνηση προς τα πίσω
{
    pwm.setPWM(0, 0, 0);
    pwm.setPWM(1, 0, speed);
    pwm.setPWM(2, 0, 0);
    pwm.setPWM(3, 0, speed);
}

```

```
pwm.setPWM(4, 0, 0);  
pwm.setPWM(5, 0, speed);  
pwm.setPWM(6, 0, 0);  
pwm.setPWM(7, 0, speed);  
}
```

**Εναλλακτικά:** Μέσα στον φάκελο «ΚΩΔΙΚΕΣ ARDUINO» μπορείτε να ανοίξετε το αρχείο κάνοντας διπλό κλικ πάνω στο αρχείο  M\_1.42 .

#### (5) Αποτέλεσμα δοκιμής:

Ανεβάστε σωστά τον κώδικα χωρίς σφάλματα. Οδηγήστε το ρομπότ προς την κατεύθυνση που επιθυμείτε.









## Δραστηριότητα 1.43: Έλεγχος ταχύτητας έξυπνου αυτοκινήτου (1)

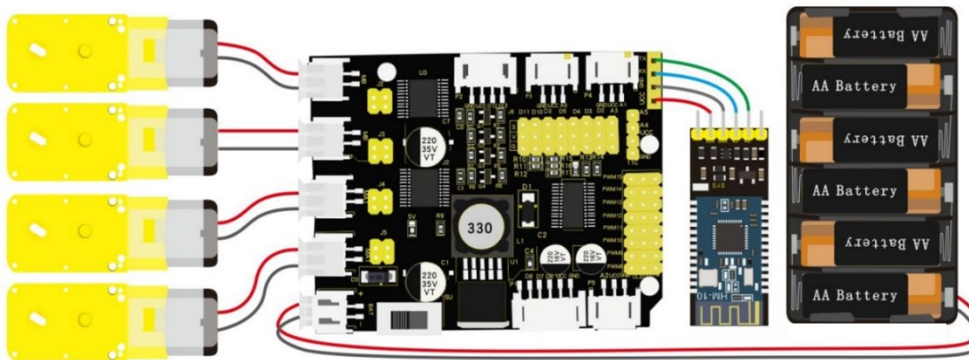
### (1) Περιγραφή:

Σε αυτή τη δραστηριότητα, όπως και στην επόμενη, θα προγραμματίσετε την ταχύτητα με την οποία κινείται το ρομπότ. Πιο συγκεκριμένα σε αυτή τη δραστηριότητα θα προγραμματίσετε την επιτάχυνση του ρομπότ και στην επόμενη θα προσθέσετε και την επιβράδυνση.

### (2) Τι χρειάζεστε:

Ελεγκτής	Πλακέτα οδήγησης κινητήρων	Μονάδα Bluetooth
		
Κινητήρας 4,5V 200rpm	Βάση μπαταρίας	Καλώδιο USB
		

### (3) Διάγραμμα καλωδίωσης:



Τα VCC, GND, TXD και RXD της μονάδας Bluetooth συνδέονται με το VCC, το GND, RXD και το TXD της ασπίδας. Συνδέστε την πλακέτα οδήγησης στον υπολογιστή σας με το καλώδιο USB.

**ΣΗΜΕΙΩΣΗ:** 1. Δώστε προσοχή στην **κατεύθυνση** των ακίδων κατά την εισαγωγή της μονάδας Bluetooth

2. **ΜΗΝ** εισάγετε τη μονάδα Bluetooth πριν ανεβάσετε τον κώδικα. (Για να αποφύγετε την αντίφαση μεταξύ σειριακής επικοινωνίας κώδικα και επικοινωνίας Bluetooth).

## (4) Κώδικας δοκιμής:

```

/*
 Δραστηριότητα 1.43
 Έλεγχος ταχύτητας έξυπνου αυτοκινήτου (1)
*/
#include <Wire.h>
#include <Adafruit_PWMServoDriver.h>
Adafruit_PWMServoDriver pwm = Adafruit_PWMServoDriver(0x47);
int speed = 2000; // Αρχικοποίηση της τιμής ταχύτητας
int turnSpeed = 2000; // Αρχικοποίηση της τιμής ταχύτητας για τις στροφές

void setup() {
  Serial.begin(9600); // Έναρξη σειριακής θύρας στη συχνότητα 9600
  pwm.begin();
  pwm.setPWMPFreq(60);
  stop();
}

void loop() {
  if (Serial.available() > 0) {
    switch (Serial.read()) {
      case 'F': advance(); Serial.println("advance"); break;
      case 'B': back(); Serial.println("back"); break;
      case 'L': turnL(); Serial.println("left"); break;
      case 'R': turnR(); Serial.println("right"); break;
      case 'S': stop(); Serial.println("stop"); break;
      case 'a': while ('a') { // Αν λάβει τον χαρακτήρα 'a', η ταχύτητα αυξάνεται
        Serial.println(speed);
        if (speed <= 2555) speed += 5; // Αυξάνει την ταχύτητα κατά 5, μέχρι να φτάσει 2555
        if (Serial.read() == 'S') {
          Serial.println(speed);
          break;
        }
      } break;
    }
  }
}

void advance() // Κίνηση μπροστά
{
  pwm.setPWM(0, 0, speed);
  pwm.setPWM(1, 0, 0);
  pwm.setPWM(2, 0, speed);
  pwm.setPWM(3, 0, 0);
  pwm.setPWM(4, 0, speed);
  pwm.setPWM(5, 0, 0);
  pwm.setPWM(6, 0, speed);
  pwm.setPWM(7, 0, 0);
}

```

```
void turnR() // Στροφή δεξιά
{
  pwm.setPWM(0, 0, turnSpeed);
  pwm.setPWM(1, 0, 0);
  pwm.setPWM(2, 0, turnSpeed);
  pwm.setPWM(3, 0, 0);
  pwm.setPWM(4, 0, 0);
  pwm.setPWM(5, 0, turnSpeed);
  pwm.setPWM(6, 0, 0);
  pwm.setPWM(7, 0, turnSpeed);
}


void turnL() // Στροφή αριστερά
{
  pwm.setPWM(0, 0, 0);
  pwm.setPWM(1, 0, turnSpeed);
  pwm.setPWM(2, 0, 0);
  pwm.setPWM(3, 0, turnSpeed);
  pwm.setPWM(4, 0, turnSpeed);
  pwm.setPWM(5, 0, 0);
  pwm.setPWM(6, 0, turnSpeed);
  pwm.setPWM(7, 0, 0);
}

void stop() // Σταματάει
{
  pwm.setPWM(0, 0, 0);
  pwm.setPWM(1, 0, 0);
  pwm.setPWM(2, 0, 0);
  pwm.setPWM(3, 0, 0);
  pwm.setPWM(4, 0, 0);
  pwm.setPWM(5, 0, 0);
  pwm.setPWM(6, 0, 0);
  pwm.setPWM(7, 0, 0);
}

void back() // Κίνηση προς τα πίσω
{
  pwm.setPWM(0, 0, 0);
  pwm.setPWM(1, 0, speed);
  pwm.setPWM(2, 0, 0);
  pwm.setPWM(3, 0, speed);
  pwm.setPWM(4, 0, 0);
  pwm.setPWM(5, 0, speed);
  pwm.setPWM(6, 0, 0);
  pwm.setPWM(7, 0, speed);
}
```

**Εναλλακτικά:** Μέσα στον φάκελο «ΚΩΔΙΚΕΣ ARDUINO» μπορείτε να ανοίξετε το αρχείο κάνοντας διπλό κλικ πάνω στο αρχείο  M\_1.43 .

**(5) Αποτέλεσμα δοκιμής:**


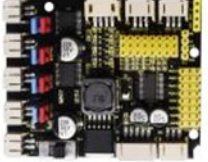




Ανεβάστε σωστά τον κώδικα χωρίς σφάλματα, έπειτα τοποθετήστε τη μονάδα Bluetooth και τρέξτε την εφαρμογή R4. Πατήστε το κουμπί παρατεταμένα  και δείτε το ρομπότ R4 να επιταχύνει στη μέγιστη ταχύτητα.

## Δραστηριότητα 1.44: Έλεγχος ταχύτητας έξυπνου αυτοκινήτου (2)

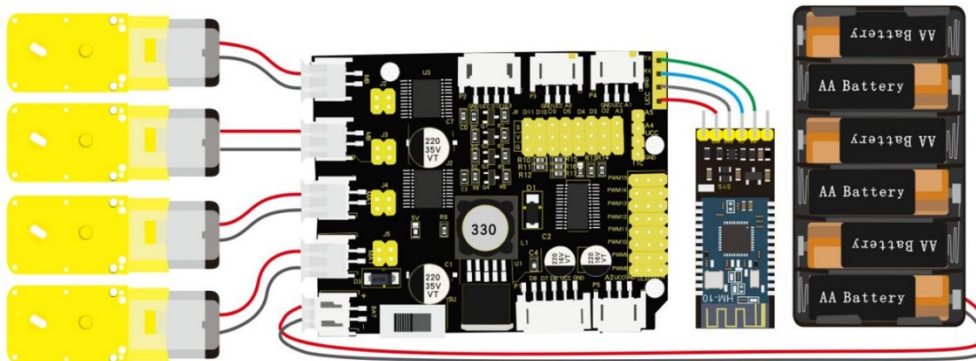
### (1) Περιγραφή:

Σε αυτή τη δραστηριότητα θα προσθέσετε και την επιβράδυνση.

### (2) Τι χρειάζεστε:

Ελεγκτής	Πλακέτα οδήγησης κινητήρων	Μονάδα Bluetooth
		
Κινητήρας 4,5V 200rpm	Βάση μπαταρίας	Καλώδιο USB
		

### (3) Διάγραμμα καλωδίωσης:



Τα VCC, GND, TXD και RXD της μονάδας Bluetooth συνδέονται με το VCC, το GND, TXD και το RXD της ασπίδας. Συνδέστε την πλακέτα οδήγησης στον υπολογιστή σας με το καλώδιο USB.

**ΣΗΜΕΙΩΣΗ:** 1. Δώστε προσοχή στην **κατεύθυνση** των ακίδων κατά την εισαγωγή της μονάδας Bluetooth

2. **ΜΗΝ** εισάγετε τη μονάδα Bluetooth πριν ανεβάσετε τον κώδικα. (Για να αποφύγετε την αντίφαση μεταξύ σειριακής επικοινωνίας κώδικα και επικοινωνίας Bluetooth)

## (4) Κώδικας δοκιμής:

```

/*
 Δραστηριότητα 1.44
 Έλεγχος ταχύτητας έξυπνου αυτοκινήτου (2)
*/
#include <Wire.h>
#include <Adafruit_PWMServoDriver.h>
Adafruit_PWMServoDriver pwm = Adafruit_PWMServoDriver(0x47);
int speed = 2000; // Αρχικοποίηση της τιμής ταχύτητας
int turnSpeed = 2000; // Αρχικοποίηση της τιμής ταχύτητας για τις στροφές

void setup() {
  Serial.begin(9600); // Έναρξη σειριακής θύρας στη συχνότητα 9600
  pwm.begin();
  pwm.setPWMPFreq(60);
  stop();
}

void loop() {
  if (Serial.available() > 0) {
    switch (Serial.read()) {
      case 'F': advance(); Serial.println("advance"); break;
      case 'B': back(); Serial.println("back"); break;
      case 'L': turnL(); Serial.println("left"); break;
      case 'R': turnR(); Serial.println("right"); break;
      case 'S': stop(); Serial.println("stop"); break;
      case 'd': while ('d') { // Αν λάβει 'd', η ταχύτητα μειώνεται
        Serial.println(speed);
        if (speed >= 5) speed -= 5; // η ταχύτητα μειώνεται κατά 5 μέχρι να ελάχιστη τιμή το 0
        if (Serial.read() == 'S') {
          Serial.println(speed);
          break;
        }
      } break;
      case 'a': while ('a') { // Αν λάβει τον χαρακτήρα 'a', η ταχύτητα αυξάνεται
        Serial.println(speed);
        if (speed <= 2555) speed += 5; // Αυξάνει την ταχύτητα κατά 5, μέχρι να φτάσει 2555
        if (Serial.read() == 'S') {
          Serial.println(speed);
          break;
        }
      } break;
    }
  }
}

void advance() // Κίνηση μπροστά
{
  pwm.setPWM(0, 0, speed);
}

```

```
pwm.setPWM(1, 0, 0);  
pwm.setPWM(2, 0, speed);  
pwm.setPWM(3, 0, 0);  
pwm.setPWM(4, 0, speed);  
pwm.setPWM(5, 0, 0);  
pwm.setPWM(6, 0, speed);  
pwm.setPWM(7, 0, 0);  
}
```

```
void turnR() // Στροφή δεξιά  
{  
  pwm.setPWM(0, 0, turnSpeed);  
  pwm.setPWM(1, 0, 0);  
  pwm.setPWM(2, 0, turnSpeed);  
  pwm.setPWM(3, 0, 0);  
  pwm.setPWM(4, 0, 0);  
  pwm.setPWM(5, 0, turnSpeed);  
  pwm.setPWM(6, 0, 0);  
  pwm.setPWM(7, 0, turnSpeed);  
}
```

```
void turnL() // Στροφή αριστερά  
{  
  pwm.setPWM(0, 0, 0);  
  pwm.setPWM(1, 0, turnSpeed);  
  pwm.setPWM(2, 0, 0);  
  pwm.setPWM(3, 0, turnSpeed);  
  pwm.setPWM(4, 0, turnSpeed);  
  pwm.setPWM(5, 0, 0);  
  pwm.setPWM(6, 0, turnSpeed);  
  pwm.setPWM(7, 0, 0);  
}
```


```
void stop() // Σταματάει  
{  
  pwm.setPWM(0, 0, 0);  
  pwm.setPWM(1, 0, 0);  
  pwm.setPWM(2, 0, 0);  
  pwm.setPWM(3, 0, 0);  
  pwm.setPWM(4, 0, 0);  
  pwm.setPWM(5, 0, 0);  
  pwm.setPWM(6, 0, 0);  
  pwm.setPWM(7, 0, 0);  
}
```

```
void back() // Κίνηση προς τα πίσω  
{  
  pwm.setPWM(0, 0, 0);  
  pwm.setPWM(1, 0, speed);  
  pwm.setPWM(2, 0, 0);  
}
```

```

pwm.setPWM(3, 0, speed);
pwm.setPWM(4, 0, 0);
pwm.setPWM(5, 0, speed);
pwm.setPWM(6, 0, 0);
pwm.setPWM(7, 0, speed);
}

```

**Εναλλακτικά:** Μέσα στον φάκελο «ΚΩΔΙΚΕΣ ARDUINO» μπορείτε να ανοίξετε το αρχείο κάνοντας διπλό κλικ πάνω στο αρχείο  M\_1.44 .

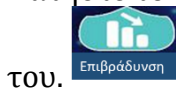
### (5) Αποτέλεσμα δοκιμής:

Ανεβάστε σωστά τον κώδικα χωρίς σφάλματα, έπειτα τοποθετήστε τη μονάδα Bluetooth και τρέξτε την εφαρμογή R4.

Πατήστε το κουμπί παρατεταμένα και το ρομπότ R4 θα επιταχύνει στη μέγιστη ταχύτητα.



Πατήστε το κουμπί παρατεταμένα, το αυτοκίνητο θα επιβραδύνει μέχρι να μηδενιστεί η ταχύτητα





του.


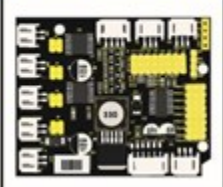






## Δραστηριότητα 1.45: Έλεγχος Σερβοκινητήρων μέσω Bluetooth(1)

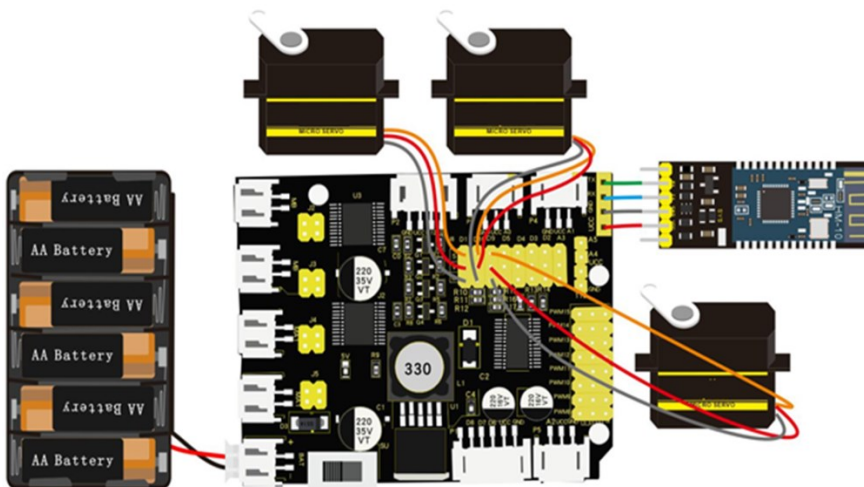
### (1) Περιγραφή:

Σε αυτή τη δραστηριότητα, όπως και στις επόμενες, θα μάθετε να ελέγχετε τους σερβοκινητήρες του ρομπότ μέσω της εφαρμογής. Στη συγκεκριμένη δραστηριότητα θα προγραμματίσετε τον έλεγχο του σερβοκινητήρα βάσης (Σερβοκινητήρα 3) προς τα δεξιά και αριστερά με τη χρήση των βελών  και .

### (2) Τι χρειάζεστε:

Control Board *1	Motor Shield*1	Bluetooth module *1	Servo motor *3	Battery Holder *1	USB Cable *1
					

### (3) Διάγραμμα καλωδίωσης:



Συνδέστε την πλακέτα οδήγησης στον υπολογιστή σας με το καλώδιο USB.

**ΣΗΜΕΙΩΣΗ:** 1. Δώστε προσοχή στην **κατεύθυνση** των ακίδων κατά την εισαγωγή της μονάδας Bluetooth

2. **ΜΗΝ** εισάγετε τη μονάδα Bluetooth πριν ανεβάσετε τον κώδικα. (Για να αποφύγετε την αντίφαση μεταξύ σειριακής επικοινωνίας κώδικα και επικοινωνίας Bluetooth)

**(4) Κώδικας δοκιμής:**

Δραστηριότητα 1.45

Έλεγχος Σερβοκινητήρων μέσω Bluetooth (1)

\*/

#include &lt;Servo.h&gt;

Servo myservo3;

int k3=90; // Αρχικοποίηση της τιμής της γωνίας του σερβοκινητήρα 3

void setup(){

Serial.begin(9600);

myservo3.attach(9);

myservo3.write(k3);

delay(1000);

}

void loop() {

if (Serial.available() &gt; 0) { // Λήψη σημάτων Bluetooth

switch (Serial.read()) {

case 'l':

while ('l') { // ο βραχίονας στρέφεται προς τα αριστερά

servo3left();

if (Serial.read() == 's') break;

}

break;

case 'r':

while ('r') { // ο βραχίονας στρέφεται προς τα δεξιά

servo3right();

if (Serial.read() == 's') break;

}

break;

}

}

}

void servo3left(){

myservo3.write(k3); // Περιστροφή προς K3

k3+=1;

delay(10);

if(k3&gt;180)k3=180; // Περιστροφή έως και 180°

}

void servo3right(){

myservo3.write(k3); // Περιστροφή προς K3

k3-=1;

delay(10);

if(k3&lt;1)k3=0; // Ελάχιστη περιστροφή 0°







}

**Εναλλακτικά:** Μέσα στον φάκελο «ΚΩΔΙΚΕΣ ARDUINO» μπορείτε να ανοίξετε το αρχείο κάνοντας διπλό κλικ πάνω στο αρχείο  M\_1.45 .

### (5) Αποτέλεσμα δοκιμής:



Ανεβάστε σωστά τον κώδικα χωρίς σφάλματα, έπειτα τοποθετήστε τη μονάδα Bluetooth και τρέξτε την εφαρμογή R4.

Πατήστε τα εικονίδια στην εφαρμογή  και  και ελέγξτε τη δεξιά και αριστερή κίνηση του βραχίονα του ρομπότ.


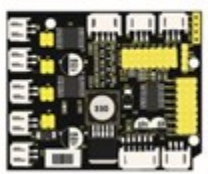


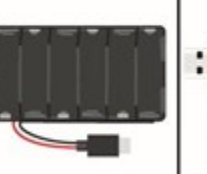

ΕΙΚΟΝΙΔΙΟ	ΣΥΜΒΟΛΟ	ΛΕΙΤΟΥΡΓΙΑ
	‘f’	Σηκώνει τον βραχίονα
	‘b’	Κατεβάζει τον βραχίονα
	‘l’	Στρίβει τον βραχίονα αριστερά
	‘r’	Στρίβει τον βραχίονα δεξιά
	‘Q’	Ανοίγει τη δαγκάνα
	‘E’	Κλείνει τη δαγκάνα

## Δραστηριότητα 1.46: Έλεγχος Σερβοκινητήρων μέσω Bluetooth (2)

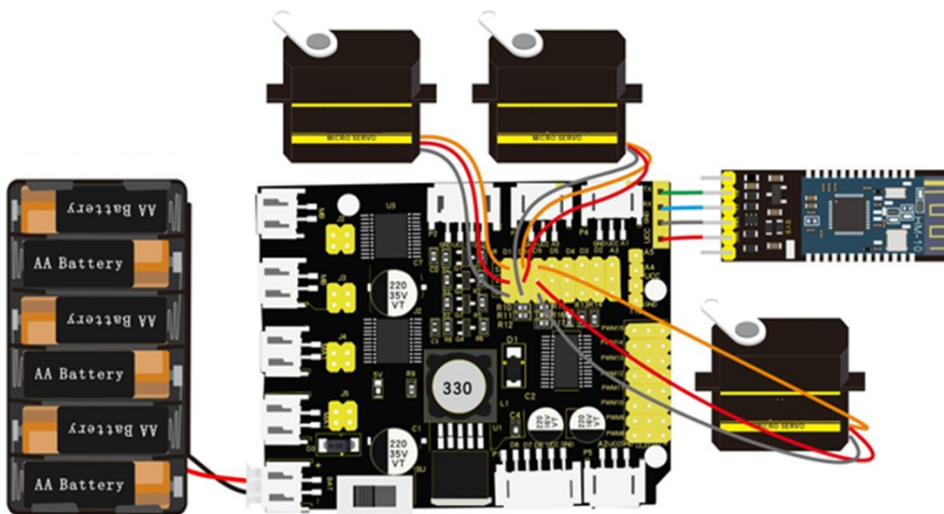
### (1) Περιγραφή:

Στη συγκεκριμένη δραστηριότητα θα προγραμματίσετε τον έλεγχο της κίνησης του σερβοκινητήρα του βραχίονα (Σερβοκινητήρα 2) προς τα πάνω και προς τα κάτω με τη χρήση των βελών  και .

### (2) Τι χρειάζεστε:

Control Board *1	Motor Shield*1	Bluetooth module *1	Servo motor *3	Battery Holder *1	USB Cable *1
					

### (3) Διάγραμμα καλωδίωσης:



Συνδέστε την πλακέτα οδήγησης στον υπολογιστή σας με το καλώδιο USB.

**ΣΗΜΕΙΩΣΗ:1.** Δώστε προσοχή στην **κατεύθυνση** των ακίδων κατά την εισαγωγή της μονάδας Bluetooth

**2.** ΜΗΝ εισάγετε τη μονάδα Bluetooth πριν ανεβάσετε τον κώδικα. (Για να αποφύγετε την αντίφαση μεταξύ σειριακής επικοινωνίας κώδικα και επικοινωνίας Bluetooth)

## (4) Κώδικας δοκιμής:

```

/*
 Δραστηριότητα 1.46
 Έλεγχος Σερβοκινητήρων μέσω Bluetooth (2)
*/
#include <Servo.h>
Servo myservo2;
int k2=120; // Αρχικοποίηση της τιμής της γωνίας του σερβοκινητήρα 2


void setup(){
  Serial.begin(9600);
  myservo2.attach(10);
  myservo2.write(k2);
  delay(1000);
}

void loop(){
  if(Serial.available()>0){ // Λήψη σημάτων Bluetooth
    switch(Serial.read()){
      case 'f' :
        while('f'){ // ο βραχίονας σηκώνεται
          servo2up();
          if(Serial.read()=='s') break;
        }
        break;
      case 'b':
        while('b'){ // ο βραχίονας χαμηλώνει
          servo2down();
          if(Serial.read()=='s') break;
        }
        break;
    }
  }
}

void servo2up(){
  myservo2.write(k2); // Περιστροφή προς K2
  k2+=1;
  delay(10);
  if(k2>120)k2=120; // Περιστροφή έως και 120°
}

void servo2down(){
  myservo2.write(k2); // Περιστροφή προς K2
  k2-=1;
  delay(10);
  if(k2<10)k2=10; // Ελάχιστη περιστροφή 10°
}







```

**Εναλλακτικά:** Μέσα στον φάκελο «ΚΩΔΙΚΕΣ ARDUINO» μπορείτε να ανοίξετε το αρχείο κάνοντας διπλό κλικ πάνω στο αρχείο  M\_1.46 .

#### (5) Αποτέλεσμα δοκιμής:

Ανεβάστε σωστά τον κώδικα χωρίς σφάλματα, έπειτα τοποθετήστε τη μονάδα Bluetooth και τρέξτε την εφαρμογή R4.

Πατήστε τα εικονίδια στην εφαρμογή  και  και ελέγξτε την πάνω και κάτω κίνηση του βραχίονα του ρομπότ.

ΕΙΚΟΝΙΔΙΟ	ΣΥΜΒΟΛΟ	ΛΕΙΤΟΥΡΓΙΑ
	'f'	Σηκώνει τον βραχίονα
	'b'	Κατεβάζει τον βραχίονα
	'l'	Στρίβει τον βραχίονα αριστερά
	'r'	Στρίβει τον βραχίονα δεξιά
	'Q'	Ανοίγει τη δαγκάνα
	'E'	Κλείνει τη δαγκάνα

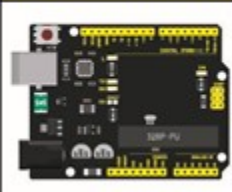
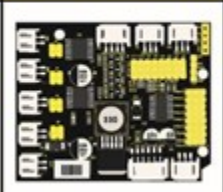




## Δραστηριότητα 1.47: Ελέγχος Σερβοκινητήρων μέσω Bluetooth (3)

### (1) Περιγραφή:

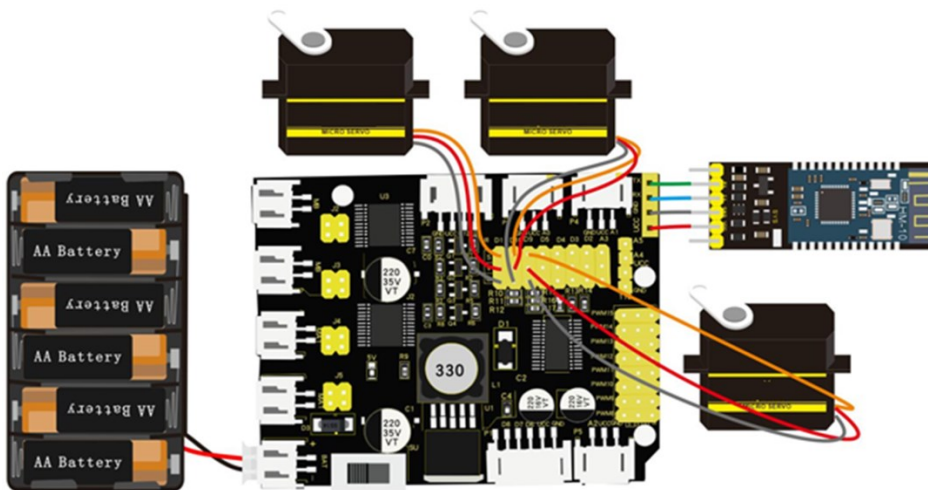
Στη συγκεκριμένη δραστηριότητα θα προγραμματίσετε τον έλεγχο της κίνησης του σερβοκινητήρα της δαγκάνας (Σερβοκινητήρα 1), θα προγραμματίσετε τον σερβοκινητήρα της δαγκάνας να ανοίγει

και να κλείνει με τη χρήση των κουμπιών **Άνοιξε** και **Κλείσε**

### (2) Τι χρειάζεστε:

Control Board *1	Motor Shield*1	Bluetooth module *1	Servo motor *3	Battery Holder *1	USB Cable *1
					

### (3) Διάγραμμα καλωδίωσης:



Συνδέστε την πλακέτα οδήγησης στον υπολογιστή σας με το καλώδιο USB.

**ΣΗΜΕΙΩΣΗ:1.** Δώστε προσοχή στην **κατεύθυνση** των ακίδων κατά την εισαγωγή της μονάδας Bluetooth

**2. ΜΗΝ** εισάγετε τη μονάδα Bluetooth πριν ανεβάσετε τον κώδικα. (Για να αποφύγετε την αντίφαση μεταξύ σειριακής επικοινωνίας κώδικα και επικοινωνίας Bluetooth)

## (4) Κώδικας δοκιμής:

```

/*
 Δραστηριότητα 1.47
 Έλεγχος Σερβοκινητήρων μέσω Bluetooth (3)
 */
#include <Servo.h>
Servo myservo1;

int k1=90; // Αρχικοποίηση της τιμής της γωνίας του σερβοκινητήρα 1

void setup(){
  Serial.begin(9600);
  myservo3.attach(11);
  myservo3.write(k1);
  delay(1000);
}

void loop(){
  if(Serial.available()>0){
    switch(Serial.read()){
      case 'Q':
        while('Q'){ // Η δαγκάνα ανοίγει
          servo1on();
          if(Serial.read()=='s')break;
        }
        break;
      case 'E':
        while('E'){ // Η δαγκάνα κλείνει
          servo1off();
          if(Serial.read()=='s')break;
        }
        break;
    }
  }
}

void servo1on(){
  myservo1.write(k1);
  k1-=1;
  delay(5);
  if(k1<20)k1=20;
}

void servo1off(){
  myservo1.write(k1);
  k1+=1;
  delay(5);
  if(k1>80)k1=90;
}

```





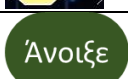
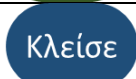


**Εναλλακτικά:** Μέσα στον φάκελο «ΚΩΔΙΚΕΣ ARDUINO» μπορείτε να ανοίξετε το αρχείο κάνοντας διπλό κλικ πάνω στο αρχείο  M\_1.47 .

### (5) Αποτέλεσμα δοκιμής:

Ανεβάστε σωστά τον κώδικα χωρίς σφάλματα, έπειτα τοποθετήστε τη μονάδα Bluetooth και τρέξτε την εφαρμογή R4.

Πατήστε τα εικονίδια στην εφαρμογή  και  και ελέγξτε τη κίνηση της δαγκάνας του ρομπότ.


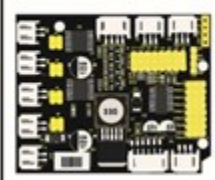




ΕΙΚΟΝΙΔΙΟ	ΣΥΜΒΟΛΟ	ΛΕΙΤΟΥΡΓΙΑ
	‘f’	Σηκώνει τον βραχίονα
	‘b’	Κατεβάζει τον βραχίονα
	‘l’	Στρίβει τον βραχίονα αριστερά
	‘r’	Στρίβει τον βραχίονα δεξιά
	‘Q’	Ανοίγει τη δαγκάνα
	‘E’	Κλείνει τη δαγκάνα

## Δραστηριότητα 1.48: Έλεγχος Σερβοκινητήρων βραχίονα μέσω Bluetooth

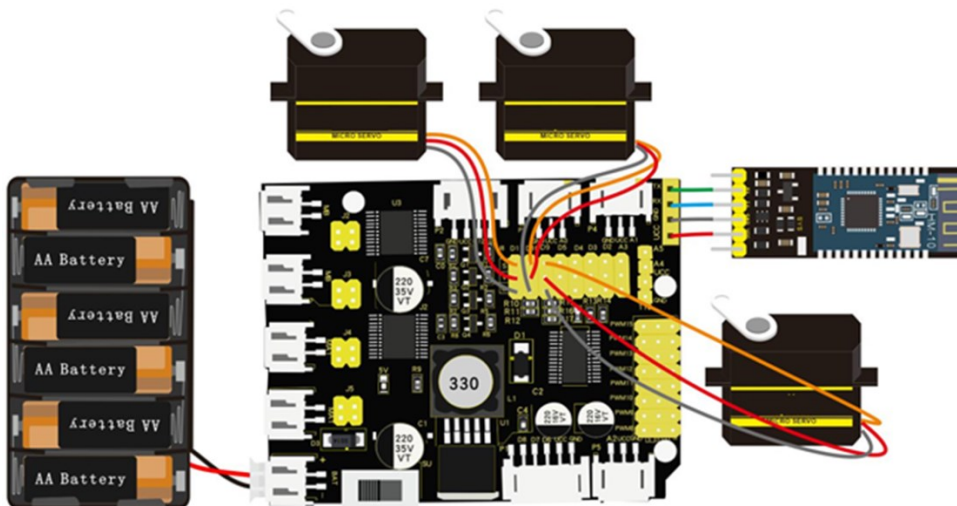
### (1) Περιγραφή:

Αφού μάθατε να προγραμματίζετε τον έλεγχο κάθε σερβοκινητήρα ξεχωριστά σε αυτή τη δραστηριότητα θα μάθετε να προγραμματίζετε τον έλεγχο όλων των σερβοκινητήρων ταυτόχρονα.

### (2) Τι χρειάζεστε:

Control Board *1	Motor Shield*1	Bluetooth module *1	Servo motor *3	Battery Holder *1	USB Cable *1
					

### (3) Διάγραμμα καλωδίωσης:



Συνδέστε την πλακέτα οδήγησης στον υπολογιστή σας με το καλώδιο USB.

**ΣΗΜΕΙΩΣΗ:1.** Δώστε προσοχή στην **κατεύθυνση** των ακίδων κατά την εισαγωγή της μονάδας Bluetooth

**2. ΜΗΝ** εισάγετε τη μονάδα Bluetooth πριν ανεβάσετε τον κώδικα. (Για να αποφύγετε την αντίφαση μεταξύ σειριακής επικοινωνίας κώδικα και επικοινωνίας Bluetooth)

## (4) Κώδικας δοκιμής:

```

/*
 Δραστηριότητα 1.48
 Έλεγχος Σερβοκινητήρων βραχίονα μέσω Bluetooth
 */
#include <Servo.h>
Servo myservo1;
Servo myservo2;
Servo myservo3;
int k1=80,k2=120,k3=90; // Αρχικοποίηση των τιμών των σερβοκινητήρων

void setup(){
  Serial.begin(9600);
  myservo1.attach(11);
  myservo2.attach(10);
  myservo3.attach(9);
  myservo1.write(k1);
  delay(1000);
  myservo2.write(k2);
  delay(1000);
  myservo3.write(k3);
  delay(1000);
}

void loop(){
  if(Serial.available()>0){ // Λήψη σημάτων Bluetooth
    switch(Serial.read()){
      case 'f':
        while('f'){ // Ο βραχίονας ανυψώνεται
          servo2up();
          if(Serial.read()=='s')break;
        }
        break;
      case 'b':
        while('b'){ // Ο βραχίονας χαμηλώνει
          servo2down();
          if(Serial.read()=='s')break;
        }
        break;
      case 'l':
        while('l'){ // Ο βραχίονας στρέφεται προς τα αριστερά
          servo3left();
          if(Serial.read()=='s')break;
        }
        break;
      case 'r':
        while('r'){ // Ο βραχίονας στρέφεται προς τα δεξιά
          servo3right();
          if(Serial.read()=='s')break;
        }
    }
  }
}

```

```

    }
    break;
case 'Q':
    while('Q'){ // Η δαγκάνα ανοίγει
        servo1on();
        if(Serial.read()=='s')break;
    }
    break;
case 'E':
    while('E'){ // Η δαγκάνα κλείνει
        servo1off();
        if(Serial.read()=='s')break;
    }
    break;
}
}
}

void servo1on(){
    myservo1.write(k1); // Περιστρέφει το K1
    k1-=1;
    delay(5);
    if(k1<20)k1=20; // Ελάχιστη περιστροφή 20°
}
void servo1off(){
    myservo1.write(k1); // Περιστρέφει το K1
    k1+=1;
    delay(5);
    if(k1>80)k1=80; // Μέγιστη περιστροφή 80°
}
void servo2up(){
    myservo2.write(k2); // Περιστρέφει το K2
    k2+=1;
    delay(10);
    if(k2>120)k2=120; // Μέγιστη περιστροφή 120°
}
void servo2down(){
    myservo2.write(k2); // Περιστρέφει το K2
    k2-=1;
    delay(10);
    if(k2<10)k2=10; // Ελάχιστη περιστροφή 10°
}
void servo3left(){
    myservo3.write(k3); // Περιστρέφει το K3
    k3+=1;
    delay(10);
    if(k3>180)k3=180; // Μέγιστη περιστροφή 180°
}
void servo3right(){
    myservo3.write(k3); // Περιστρέφει το K3

```

```

k3-=1;
delay(10);
if(k3<1)k3=0; // Ελάχιστη περιστροφή 0°
}






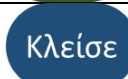
```

**Εναλλακτικά:** Μέσα στον φάκελο «ΚΩΔΙΚΕΣ ARDUINO» μπορείτε να ανοίξετε το αρχείο κάνοντας διπλό κλικ πάνω στο αρχείο  M\_1.48 .

#### (5) Αποτέλεσμα δοκιμής:

Ανεβάστε σωστά τον κώδικα χωρίς σφάλματα, έπειτα τοποθετήστε τη μονάδα Bluetooth και τρέξτε την εφαρμογή R4.

Πατήστε τα εικονίδια ελέγχου του βραχίονα στην εφαρμογή και δοκιμάστε κάθε πιθανή κίνηση του βραχίονα που μπορείτε να σκεφτείτε.

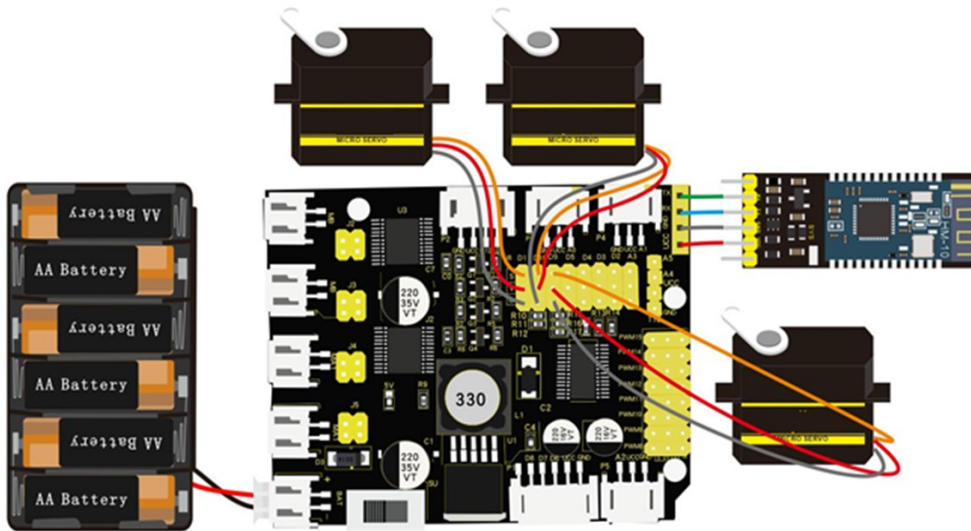
ΕΙΚΟΝΙΔΙΟ	ΣΥΜΒΟΛΟ	ΛΕΙΤΟΥΡΓΙΑ
	'f'	Σηκώνει τον βραχίονα
	'b'	Κατεβάζει τον βραχίονα
	'l'	Στρίβει τον βραχίονα αριστερά
	'r'	Στρίβει τον βραχίονα δεξιά
	'Q'	Ανοίγει τη δαγκάνα
	'E'	Κλείνει τη δαγκάνα

## Δραστηριότητα 1.49: Λειτουργία Μνήμης (1)

### (1) Περιγραφή:

Η λειτουργία μνήμης είναι ένα από τα χαρακτηριστικά του βραχίονα ρομπότ, που καταγράφει και προγραμματίζει τη συχνότητα, το χρόνο και το πλάτος της κίνησης. Σε αυτή τη δραστηριότητα θα προγραμματίσετε τη λειτουργία μνήμης για την κίνηση του βραχίονα του ρομπότ προς τα μπροστά, προς τα πίσω και συλλογή αντικειμένου.

### (2) Διάγραμμα καλωδίωσης:



Συνδέστε την πλακέτα οδήγησης στον υπολογιστή σας με το καλώδιο USB.

**ΣΗΜΕΙΩΣΗ:1.** Δώστε προσοχή στην **κατεύθυνση** των ακίδων κατά την εισαγωγή της μονάδας Bluetooth

**2. ΜΗΝ** εισάγετε τη μονάδα Bluetooth πριν ανεβάσετε τον κώδικα. (Για να αποφύγετε την αντίφαση μεταξύ σειριακής επικοινωνίας κώδικα και επικοινωνίας Bluetooth)

### (3) Κώδικας δοκιμής:

```
/*
Δραστηριότητα 1.49
Λειτουργία Μνήμης (1)
*/
#include <Servo.h>
Servo myservo1;      // ορίζει τον σερβοκινητήρα της δαγκάνας
Servo myservo2;      // ορίζει τον σερβοκινητήρα 2
Servo myservo3;      // ορίζει τον σερβοκινητήρα της βάσης
int k1 = 80, k2 = 120, k3 = 90; // αρχικοποίηση της τιμής γωνίας των σερβοκινητήρων
int M1[20], M2[20], M3[20];    // χρησιμοποιείται για την αποθήκευση της τιμής της γωνίας
// ανάγνωσης
int i = 0, j = 0, t = 0;      // i χρησιμοποιείται για την αποθήκευση του πίνακα, j για την αποθήκευση
// της μέγιστης τιμής, t για τον βρόχο while
```

```

void setup() {
  Serial.begin(9600); // Ορισμός ρυθμού baud σε 9600
  myservo1.attach(11);
  myservo2.attach(10);
  myservo3.attach(9);
  myservo1.write(k1);
  delay(1000);
  myservo2.write(k2);
  delay(1000);
  myservo3.write(k3);
  delay(1000);
}

void loop(){
  if(Serial.available()>0){ // Λήψη σημάτων Bluetooth
    switch(Serial.read()){
      case 'f':
        while('f'){ // Ο βραχίονας ανυψώνεται
          servo2up();
          if(Serial.read()=='s')break;
        }
        break;
      case 'b':
        while('b'){ // Ο βραχίονας χαμηλώνει
          servo2down();
          if(Serial.read()=='s')break;
        }
        break;
      case 'l':
        while('l'){ // Ο βραχίονας στρέφεται προς τα αριστερά
          servo3left();
          if(Serial.read()=='s')break;
        }
        break;
      case 'r':
        while('r'){ // Ο βραχίονας στρέφεται προς τα δεξιά
          servo3right();
          if(Serial.read()=='s')break;
        }
        break;
      case 'Q':
        while('Q'){ // Η δαγκάνα ανοίγει
          servo1on();
          if(Serial.read()=='s')break;
        }
        break;
      case 'E':
        while('E'){ // Η δαγκάνα κλείνει
          servo1off();
          if(Serial.read()=='s')break;
        }
    }
  }
}

```

```

    }
    break;
case 't':
{
    M1[i] = myservo1.read(); // αποθήκευση της τιμής της γωνίας των σερβοκινητήρων σε πίνακα
    delay(100);
    M2[i] = myservo2.read();
    delay(100);
    M3[i] = myservo3.read();
    delay(100);
    i++; // i προσθέτει 1
    j = i; // ορίζει την τιμή του i στο j
    delay(200);
}
break;
case 'i':
{
    i = 0;
    t = 1;
    k1 = myservo1.read(); // διαβάζει τις τιμές των γωνιών και τις θέτει στην ανάλογη μεταβλητή
    k2 = myservo2.read();
    k3 = myservo3.read();
    while (t) {
        for (int k = 0; k < j; k++) { // επανάληψη j φορές
            if (k1 < M1[k]) { // εάν η τρέχουσα τιμή γωνίας του σερβοκινητήρα 1 είναι μικρότερη
                από την τιμή του πίνακα 1
                    while (k1 < M1[k]) { // ο σερβοκινητήρας περιστρέφεται στη θέση όπου είναι
                        αποθηκευμένος στον πίνακα
                            myservo1.write(k1); // ο σερβοκινητήρας 1 εκτελεί την κίνηση
                                k1++; // K1 προσθέτει 1
                                    delay(10); // καθυστέρηση για 10ms / έλεγχος της ταχύτητας περιστροφής του
                                        σερβοκινητήρα
                                            }
                                                } else { // όταν η τιμή της γωνίας του σερβοκινητήρα 1 είναι μεγαλύτερη από την τιμή
                                                    που έχει αποθηκευτεί στον πίνακα 1
                                                        while (k1 > M1[k]) { // ο σερβοκινητήρας περιστρέφεται στη θέση όπου είναι
                                                            αποθηκευμένος στον πίνακα
                                                                myservo1.write(k1); // ο σερβοκινητήρας 1 εκτελεί την κίνηση
                                                                    k1--; // k1 αφαιρεί 1
                                                                        delay(10); // καθυστέρηση για 10ms / έλεγχος της ταχύτητας περιστροφής του
                                                                            σερβοκινητήρα
                                                                                }
                                                                                    }
                                                                                        // το ίδιο παρακάτω
                                                                                            if (k2 < M2[k]) {
                                                                                                while (k2 < M2[k]) {
                                                                                                    myservo2.write(k2);
                                                                                                        k2++;
                                                                                                            delay(10);
                                                                                                                }
                                                                                                                    }

```



```

    } else {
      while (k2 > M2[k]) {
        myservo2.write(k2);
        k2--;
        delay(10);
      }
    }
    if (k3 < M3[k]) {
      while (k3 < M3[k]) {
        myservo3.write(k3);
        k3++;
        delay(10);
      }
    } else {
      while (k3 > M3[k]) {
        myservo3.write(k3);
        k3--;
        delay(10);
      }
    }
  }
}
if (Serial.read() == 't') { // έξοδος απο τον βρόχο while
  t = 0;
  break;
}
}
}
break;
}
}
}

void servo1on(){
  myservo1.write(k1); // Περιστρέφει το K1
  k1-=1;
  delay(5);
  If (k1<20) k1=20; // Ελάχιστη περιστροφή 20°
}

void servo1off(){
  myservo1.write(k1); // Περιστρέφει το K1
  k1+=1;
  delay(5);
  If (k1>80) k1=80; // Μέγιστη περιστροφή 80°
}

```

```

void servo2up(){
  myservo2.write(k2); // Περιστρέφει το K2
  k2+=1;
  delay(10);
  If (k2>120) k2=120; // Μέγιστη περιστροφή 120°
}

void servo2down(){
  myservo2.write(k2); // Περιστρέφει το K2
  k2-=1;
  delay(10);
  If (k2<10) k2=10; // Ελάχιστη περιστροφή 10°
}

void servo3left(){
  myservo3.write(k3); // Περιστρέφει το K3
  k3+=1;
  delay(10);
  if(k3>180)k3=180; // Μέγιστη περιστροφή 180°
}
void servo3right(){
  myservo3.write(k3); // Περιστρέφει το K3
  k3-=1;
  delay(10);
  if(k3<1)k3=0; // Ελάχιστη περιστροφή 0°
}


```


**Εναλλακτικά:** Μέσα στον φάκελο «ΚΩΔΙΚΕΣ ARDUINO» μπορείτε να ανοίξετε το αρχείο κάνοντας διπλό κλικ πάνω στο αρχείο  M\_1.49 .


#### (4) Αποτέλεσμα δοκιμής:

Ανεβάστε σωστά τον κώδικα χωρίς σφάλματα, έπειτα τοποθετήστε τη μονάδα Bluetooth και τρέξτε την εφαρμογή R4.

Τοποθετήστε ένα αντικείμενο μπροστά από το ρομπότ.

Πατήστε το εικονίδιο  και κάντε μια κίνηση μπροστά του βραχίονα για να πλησιάσετε το

αντικείμενο. Πατήστε πάλι το εικονίδιο  και πάρτε το αντικείμενο. Επαναλάβετε για να πάτε το αντικείμενο προς τα πίσω και να το αφήσετε πατώντας το εικονίδιο αποθήκευσης για κάθε διαφορετική κίνηση του βραχίονα.

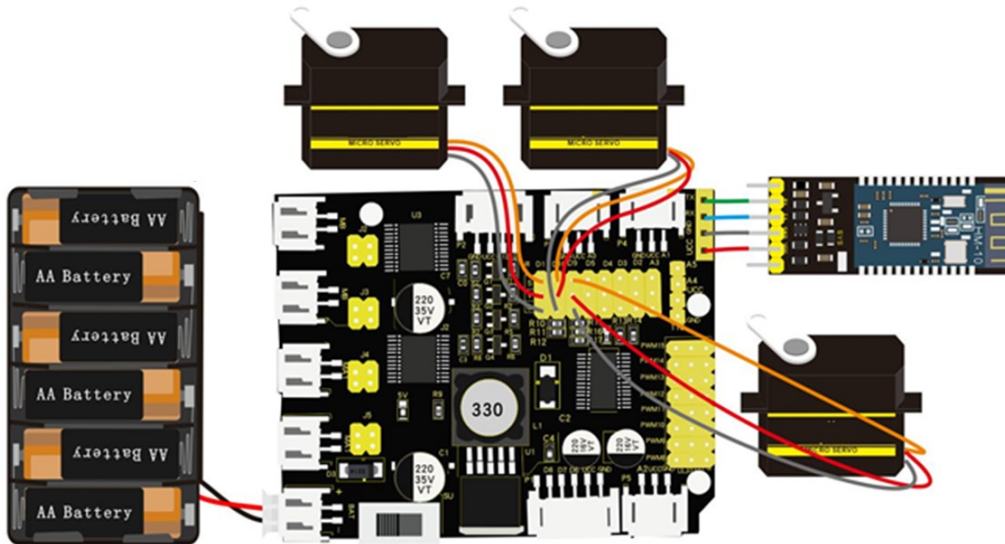
Στη συνέχεια πατήστε το πλήκτρο  για να εκτελέσετε τις αποθηκευμένες κινήσεις του βραχίονα.

## Δραστηριότητα 1.50: Λειτουργία Μνήμης (2)

### (1) Περιγραφή:

Σε αυτή τη δραστηριότητα σε συνέχεια της προηγούμενης θα προγραμματίσετε τη λειτουργία μνήμης για τη κίνηση του βραχίονα δεξιά-αριστερά και συλλογή αντικειμένου.

### (2) Διάγραμμα καλωδίωσης:



Συνδέστε την πλακέτα οδήγησης στον υπολογιστή σας με το καλώδιο USB.

**ΣΗΜΕΙΩΣΗ:1.** Δώστε προσοχή στην **κατεύθυνση** των ακίδων κατά την εισαγωγή της μονάδας Bluetooth

**2. ΜΗΝ** εισάγετε τη μονάδα Bluetooth πριν ανεβάσετε τον κώδικα. (Για να αποφύγετε την αντίφαση μεταξύ σειριακής επικοινωνίας κώδικα και επικοινωνίας Bluetooth)

### (3) Κώδικας δοκιμής:

```
/*
Δραστηριότητα 1.50
Λειτουργία Μνήμης (2)
*/
#include <Servo.h>
Servo myservo1;           // ορίζει τον σερβομηχανισμό της δαγκάνας
Servo myservo2;           // ορίζει τον σερβοκινητήρα 2
Servo myservo3;           // ορίζει τον σερβοκινητήρα της βάσης
int k1 = 80, k2 = 120, k3 = 90; // αρχικοποίηση της τιμής γωνίας των σερβοκινητήρων
int M1[20], M2[20], M3[20]; // χρησιμοποιείται για την αποθήκευση της τιμής της γωνίας
// ανάγνωσης
int i = 0, j = 0, t = 0;    // i χρησιμοποιείται για την αποθήκευση του πίνακα, j για την αποθήκευση
// της μέγιστης τιμής, t για τον βρόχο while

void setup() {
  Serial.begin(9600); // Ορισμός ρυθμού baud σε 9600
  myservo1.attach(11);

```

```

myservo2.attach(10);
myservo3.attach(9);
myservo1.write(k1);
delay(1000);
myservo2.write(k2);
delay(1000);
myservo3.write(k3);
delay(1000);
}

void loop(){
if(Serial.available()>0){ // Λήψη σημάτων Bluetooth
switch(Serial.read()){
case 'f':
while('f'){ // Ο βραχίονας ανυψώνεται
servo2up();
if(Serial.read()=='s')break;
}
break;
case 'b':
while('b'){ // Ο βραχίονας χαμηλώνει
servo2down();
if(Serial.read()=='s')break;
}
break;
case 'l':
while('l'){ // Ο βραχίονας στρέφεται προς τα αριστερά
servo3left();
if(Serial.read()=='s')break;
}
break;
case 'r':
while('r'){ // Ο βραχίονας στρέφεται προς τα δεξιά
servo3right();
if(Serial.read()=='s')break;
}
break;
case 'Q':
while('Q'){ // Η δαγκάνα ανοίγει
servo1on();
if(Serial.read()=='s')break;
}
break;
case 'E':
while('E'){ // Η δαγκάνα κλείνει
servo1off();
if(Serial.read()=='s')break;
}
break;
case 't':

```

```

{
  M1[i] = myservo1.read(); // αποθήκευση της τιμής της γωνίας των σερβοκινητήρων σε πίνακα
  delay(100);
  M2[i] = myservo2.read();
  delay(100);
  M3[i] = myservo3.read();
  delay(100);
  i++; // i προσθέτει 1
  j = i; // ορίζει την τιμή του i στο j
  delay(200);
}
break;
case 'i':
{
  i = 0;
  t = 1;
  k1 = myservo1.read(); // διαβάζει τις τιμές των γωνιών και τις θέτει στην ανάλογη μεταβλητή
  k2 = myservo2.read();
  k3 = myservo3.read();
  while (t) {
    for (int k = 0; k < j; k++) { // επανάληψη j φορές
      if (k1 < M1[k]) { // εάν η τρέχουσα τιμή γωνίας του σερβοκινητήρα 1 είναι μικρότερη
        από την τιμή του πίνακα 1
          while (k1 < M1[k]) { // ο σερβοκινητήρας περιστρέφεται στη θέση όπου είναι
            αποθηκευμένος στον πίνακα
              myservo1.write(k1); // ο σερβοκινητήρας 1 εκτελεί την κίνηση
              k1++; // K1 προσθέτει 1
              delay(10); // καθυστέρηση για 10ms / έλεγχος της ταχύτητας περιστροφής του
            σερβοκινητήρα
          }
        } else { // όταν η τιμή της γωνίας του σερβοκινητήρα 1 είναι μεγαλύτερη από την τιμή
          που έχει αποθηκευτεί στον πίνακα 1
            while (k1 > M1[k]) { // ο σερβοκινητήρας περιστρέφεται στη θέση όπου είναι
              αποθηκευμένος στον πίνακα
                myservo1.write(k1); // ο σερβοκινητήρας 1 εκτελεί την κίνηση
                k1--; // k1 αφαιρεί 1
                delay(10); // καθυστέρηση για 10ms / έλεγχος της ταχύτητας περιστροφής του
              σερβοκινητήρα
            }
          }
        // το ίδιο παρακάτω
        if (k2 < M2[k]) {
          while (k2 < M2[k]) {
            my2.write(k2);
            k2++;
            delay(10);
          }
        } else {
          while (k2 > M2[k]) {
            myservo2.write(k2);

```

```

        k2--;
        delay(10);
    }
}
if (k3 < M3[k]) {
    while (k3 < M3[k]) {
        myservo3.write(k3);
        k3++;
        delay(10);
    }
} else {
    while (k3 > M3[k]) {
        myservo3.write(k3);
        k3--;
        delay(10);
    }
}
}
if (Serial.read() == 't') { // έξοδος από τον βρόχο while
    t = 0;
    break;
}
}
}
break;
}
}
}

void servo1on(){
    myservo1.write(k1); // Περιστρέφει το K1
    k1-=1;
    delay(5);
    If (k1<20) k1=20; // Ελάχιστη περιστροφή 20°
}

void servo1off(){
    myservo1.write(k1); // Περιστρέφει το K1
    k1+=1;
    delay(5);
    If (k1>80) k1=80; // Μέγιστη περιστροφή 80°
}
void servo2up(){
    myservo2.write(k2); // Περιστρέφει το K2
    k2+=1;
    delay(10);
    If (k2>120) k2=120; // Μέγιστη περιστροφή 120°
}
}


```

```

void servo2down(){
  myservo2.write(k2); // Περιστρέφει το K2
  k2-=1;
  delay(10);
  If (k2<10) k2=10; // Ελάχιστη περιστροφή 10°
}

void servo3left(){
  myservo3.write(k3); // Περιστρέφει το K3
  k3+=1;
  delay(10);
  if(k3>180)k3=180; // Μέγιστη περιστροφή 180°
}
void servo3right(){
  myservo3.write(k3); // Περιστρέφει το K3
  k3-=1;
  delay(10);
  if(k3<1)k3=0; // Ελάχιστη περιστροφή 0°
}


```

**Εναλλακτικά:** Μέσα στον φάκελο «ΚΩΔΙΚΕΣ ARDUINO» μπορείτε να ανοίξετε το αρχείο κάνοντας διπλό κλικ πάνω στο αρχείο  M\_1.50 .

#### (4) Αποτέλεσμα δοκιμής:

Ανεβάστε σωστά τον κώδικα χωρίς σφάλματα, έπειτα τοποθετήστε τη μονάδα Bluetooth και τρέξτε την εφαρμογή R4.

Τοποθετήστε ένα αντικείμενο σε ένα σημείο στα αριστερά του ρομπότ.

Πατήστε το εικονίδιο  για κάθε διαφορετική κίνηση του βραχίονα του ρομπότ.

Στη συνέχεια πατήστε το πλήκτρο



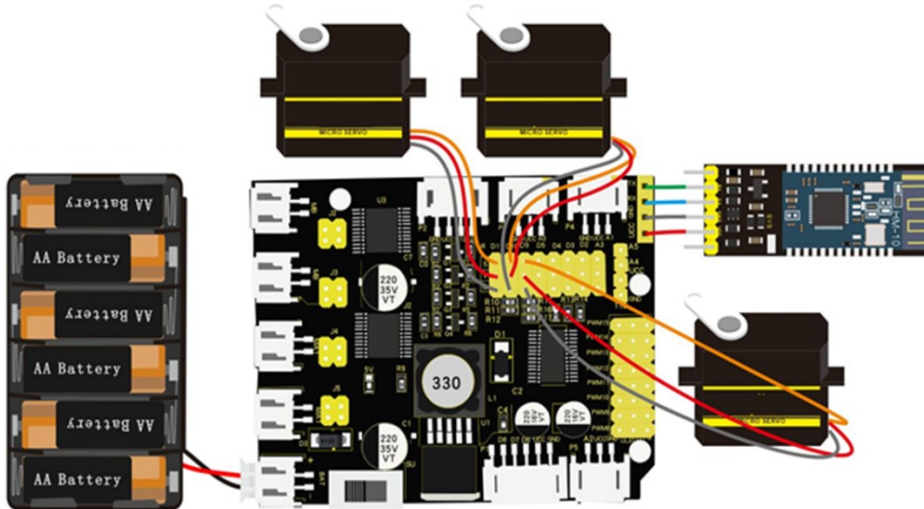
και δείτε το ρομπότ να επαναλαμβάνει τις κινήσεις που αποθηκεύσατε.

## Δραστηριότητα 1.51: Λειτουργία Μνήμης (3)

### (1) Περιγραφή:

Σε αυτή τη δραστηριότητα, σε συνέχεια της προηγούμενης, θα προγραμματίσετε τη λειτουργία μνήμης για την κίνηση του βραχίονα του ρομπότ προς όλες τις κατευθύνσεις.

### (2) Διάγραμμα καλωδίωσης:



Συνδέστε την πλακέτα οδήγησης στον υπολογιστή σας με το καλώδιο USB.

**ΣΗΜΕΙΩΣΗ:1.** Δώστε προσοχή στην **κατεύθυνση** των ακίδων κατά την εισαγωγή της μονάδας Bluetooth

**2. ΜΗΝ** εισάγετε τη μονάδα Bluetooth πριν ανεβάσετε τον κώδικα. (Για να αποφύγετε την αντίφαση μεταξύ σειριακής επικοινωνίας κώδικα και επικοινωνίας Bluetooth)

### (3) Κώδικας δοκιμής:

```

/*
Δραστηριότητα 1.51
Λειτουργία Μνήμης (3)
*/
#include <Servo.h>
Servo myservo1;           // ορίζει τον σερβοκινητήρα της δαγκάνας
Servo myservo2;           // ορίζει τον σερβοκινητήρα 2
Servo myservo3;           // ορίζει τον σερβοκινητήρα της βάσης
int k1 = 80, k2 = 120, k3 = 90; // αρχικοποίηση της τιμής γωνίας των σερβοκινητήρων
int M1[20], M2[20], M3[20]; // χρησιμοποιείται για την αποθήκευση της τιμής της γωνίας
// ανάγνωσης
int i = 0, j = 0, t = 0;    // i χρησιμοποιείται για την αποθήκευση του πίνακα, j για την αποθήκευση
// της μέγιστης τιμής, t για τον βρόχο while

void setup() {
  Serial.begin(9600); // Ορισμός ρυθμού baud σε 9600
  myservo1.attach(11);
  myservo2.attach(10);
  myservo3.attach(9);
}

```



```

myservo1.write(k1);
delay(1000);
myservo2.write(k2);
delay(1000);
myservo3.write(k3);
delay(1000);
}

void loop(){
if(Serial.available()>0){ // Λήψη σημάτων Bluetooth
switch(Serial.read()){
case 'f':
while('f'){ // Ο βραχίονας ανυψώνεται
servo2up();
if(Serial.read()=='s')break;
}
break;
case 'b':
while('b'){ // Ο βραχίονας χαμηλώνει
servo2down();
if(Serial.read()=='s')break;
}
break;
case 'l':
while('l'){ // Ο βραχίονας στρέφεται προς τα αριστερά
servo3left();
if(Serial.read()=='s')break;
}
break;
case 'r':
while('r'){ // Ο βραχίονας στρέφεται προς τα δεξιά
servo3right();
if(Serial.read()=='s')break;
}
break;
case 'Q':
while('Q'){ // Η δαγκάνα ανοίγει
servo1on();
if(Serial.read()=='s')break;
}
break;
case 'E':
while('E'){ // Η δαγκάνα κλείνει
servo1off();
if(Serial.read()=='s')break;
}
break;
case 't':
{
M1[i] = myservo1.read(); // αποθήκευση της τιμής της γωνίας των σερβοκινητήρων σε πίνακα

```

```

delay(100);
M2[i] = myservo2.read();
delay(100);
M3[i] = myservo3.read();
delay(100);
i++; // i προσθέτει 1
j = i; // ορίζει την τιμή του i στο j
delay(200);
}
break;
case 'i':
{
i = 0;
t = 1;
k1 = myservo1.read(); // διαβάζει τις τιμές των γωνιών και τις θέτει στην ανάλογη μεταβλητή
k2 = myservo2.read();
k3 = myservo3.read();
while (t) {
for (int k = 0; k < j; k++) { // επανάληψη j φορές
if (k1 < M1[k]) { // εάν η τρέχουσα τιμή γωνίας του σερβοκινητήρα 1 είναι μικρότερη
από την τιμή του πίνακα 1
while (k1 < M1[k]) { // ο σερβοκινητήρας περιστρέφεται στη θέση όπου είναι
αποθηκευμένος στον πίνακα
myservo1.write(k1); // ο σερβοκινητήρας 1 εκτελεί την κίνηση
k1++; // K1 προσθέτει 1
delay(10); // καθυστέρηση για 10ms / έλεγχος της ταχύτητας περιστροφής του
σερβοκινητήρα
}
} else { // όταν η τιμή της γωνίας του σερβοκινητήρα 1 είναι μεγαλύτερη από την τιμή
που έχει αποθηκευτεί στον πίνακα 1
while (k1 > M1[k]) { // ο σερβοκινητήρας περιστρέφεται στη θέση όπου είναι
αποθηκευμένος στον πίνακα
myservo1.write(k1); // ο σερβοκινητήρας 1 εκτελεί την κίνηση
k1--; // k1 αφαιρεί 1
delay(10); // καθυστέρηση για 10ms / έλεγχος της ταχύτητας περιστροφής του
σερβοκινητήρα
}
}
// το ίδιο παρακάτω
if (k2 < M2[k]) {
while (k2 < M2[k]) {
myservo2.write(k2);
k2++;
delay(10);
}
} else {
while (k2 > M2[k]) {
myservo2.write(k2);
k2--;
delay(10);
}
}
}
}

```

```

    }
  }
  if (k3 < M3[k]) {
    while (k3 < M3[k]) {
      myservo3.write(k3);
      k3++;
      delay(10);
    }
  } else {
    while (k3 > M3[k]) {
      myservo3.write(k3);
      k3--;
      delay(10);
    }
  }
}
}
}
if (Serial.read() == 't') { // έξοδος απο τον βρόχο while
  t = 0;
  break;
}
}
}
break;
}
}
}

```

```

void servo1on(){
  myservo1.write(k1); // Περιστρέφει το K1
  k1-=1;
  delay(5);
  If (k1<20) k1=20; // Ελάχιστη περιστροφή 20°
}

```

```

void servo1off(){
  myservo1.write(k1); // Περιστρέφει το K1
  k1+=1;
  delay(5);
  If (k1>80) k1=80; // Μέγιστη περιστροφή 80°
}

```

```

void servo2up(){
  myservo2.write(k2); // Περιστρέφει το K2
  k2+=1;
  delay(10);
  If (k2>120) k2=120; // Μέγιστη περιστροφή 120°
}


```

```

void servo2down(){
  myservo2.write(k2); // Περιστρέφει το K2
  k2-=1;
  delay(10);
  if (k2<10) k2=10; // Ελάχιστη περιστροφή 10°
}

void servo3left(){
  myservo3.write(k3); // Περιστρέφει το K3
  k3+=1;
  delay(10);
  if(k3>180)k3=180; // Μέγιστη περιστροφή 180°
}
void servo3right(){
  myservo3.write(k3); // Περιστρέφει το K3
  k3-=1;
  delay(10);
  if(k3<1)k3=0; // Ελάχιστη περιστροφή 0°
}

```


**Εναλλακτικά:** Μέσα στον φάκελο «ΚΩΔΙΚΕΣ ARDUINO» μπορείτε να ανοίξετε το αρχείο κάνοντας διπλό κλικ πάνω στο αρχείο  M\_1.51 .

#### (4) Αποτέλεσμα δοκιμής:

Ανεβάστε σωστά τον κώδικα χωρίς σφάλματα, έπειτα τοποθετήστε τη μονάδα Bluetooth και τρέξτε την εφαρμογή R4.

Τοποθετήστε ένα αντικείμενο σε ένα σημείο στα αριστερά και ένα μπροστά από το ρομπότ.

Αποθηκεύστε τις κινήσεις: Συλλογή αντικειμένου από μπροστά και τοποθέτηση του δεξιά, συλλογή

αντικειμένου από αριστερά και τοποθέτηση του μπροστά. Πατήστε το εικονίδιο  για κάθε διαφορετική κίνηση του βραχίονα του ρομπότ.

Στη συνέχεια πατήστε το πλήκτρο



και δείτε το ρομπότ να επαναλαμβάνει τις κινήσεις που







αποθηκεύσατε.

## Δραστηριότητα 1.52: Λειτουργία Μνήμης (4)

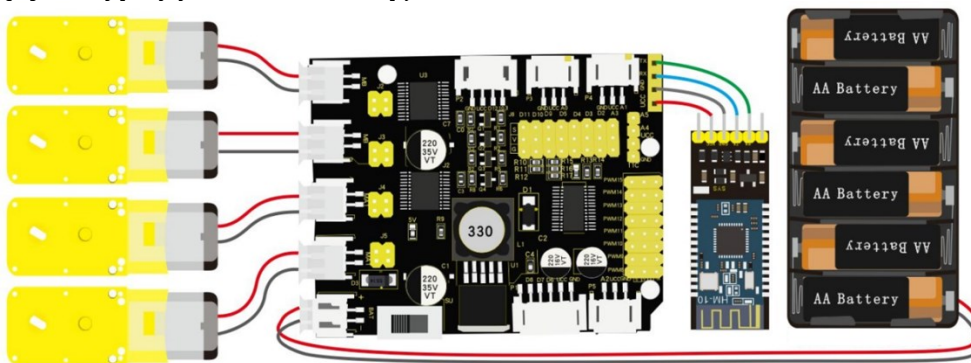
### (1) Περιγραφή:

Σε αυτή τη δραστηριότητα, σε συνέχεια της προηγούμενης, θα προγραμματίσετε τη λειτουργία μνήμης για ένα διαφορετικό συνδυασμό κινήσεων του βραχίονα του ρομπότ. Θα τοποθετήσετε τρία αντικείμενα αριστερά, δεξιά και μπροστά και θα τα μεταφέρετε διαδοχικά σε διαφορετική θέση. Το δεξί αντικείμενο στη μέση, το μεσαίο αντικείμενο αριστερά και το αριστερό αντικείμενο δεξιά.

### (2) Τι χρειάζεστε:

Ελεγκτής	Πλακέτα οδήγησης κινητήρων	Μονάδα Bluetooth
		
Κινητήρας 4,5V 200rpm	Βάση μπαταρίας	Καλώδιο USB
		

### (3) Διάγραμμα καλωδίωσης:



Συνδέστε την πλακέτα οδήγησης στον υπολογιστή σας με το καλώδιο USB.

**ΣΗΜΕΙΩΣΗ:1.** Δώστε προσοχή στην **κατεύθυνση** των ακίδων κατά την εισαγωγή της μονάδας Bluetooth

**2. ΜΗΝ** εισάγετε τη μονάδα Bluetooth πριν ανεβάσετε τον κώδικα. (Για να αποφύγετε την αντίφαση μεταξύ σειριακής επικοινωνίας κώδικα και επικοινωνίας Bluetooth)

## (4) Κώδικας δοκιμής:

```

/*
 Δραστηριότητα 1.52
 Λειτουργία Μνήμης (4)
*/
#include <Servo.h>
Servo myservo1;           // ορίζει τον σερβοκινητήρα της δαγκάνας
Servo myservo2;           // ορίζει τον σερβοκινητήρα 2
Servo myservo3;           // ορίζει τον σερβοκινητήρα της βάσης
int k1 = 80, k2 = 120, k3 = 90; // αρχικοποίηση της τιμής γωνίας των σερβοκινητήρων
int M1[20], M2[20], M3[20]; // χρησιμοποιείται για την αποθήκευση της τιμής της γωνίας
ανάγνωσης
int i = 0, j = 0, t = 0;    // i χρησιμοποιείται για την αποθήκευση του πίνακα, j για την αποθήκευση
της μέγιστης τιμής, t για τον βρόχο while

void setup() {
  Serial.begin(9600); // Ορισμός ρυθμού baud σε 9600
  myservo1.attach(11);
  myservo2.attach(10);
  myservo3.attach(9);
  myservo1.write(k1);
  delay(1000);
  myservo2.write(k2);
  delay(1000);
  myservo3.write(k3);
  delay(1000);
}

void loop(){
  if(Serial.available()>0){ // Λήψη σημάτων Bluetooth
    switch(Serial.read()){
      case 'f':
        while('f'){ // Ο βραχίονας ανυψώνεται
          servo2up();
          if(Serial.read()=='s')break;
        }
        break;
      case 'b':
        while('b'){ // Ο βραχίονας χαμηλώνει
          servo2down();
          if(Serial.read()=='s')break;
        }
        break;
      case 'l':
        while('l'){ // Ο βραχίονας στρέφεται προς τα αριστερά
          servo3left();
          if(Serial.read()=='s')break;
        }
        break;
    }
  }
}

```

```

case 'r':
    while('r'){ // Ο βραχίονας στρέφεται προς τα δεξιά
        servo3right();
        if(Serial.read()=='s')break;
    }
    break;
case 'Q':
    while('Q'){ // Η δαγκάνα ανοίγει
        servo1on();
        if(Serial.read()=='s')break;
    }
    break;
case 'E':
    while('E'){ // Η δαγκάνα κλείνει
        servo1off();
        if(Serial.read()=='s')break;
    }
    break;
case 't':
    {
        M1[i] = myservo1.read(); // αποθήκευση της τιμής της γωνίας των σερβοκινητήρων σε πίνακα
        delay(100);
        M2[i] = myservo2.read();
        delay(100);
        M3[i] = myservo3.read();
        delay(100);
        i++; // i προσθέτει 1
        j = i; // ορίζει την τιμή του i στο j
        delay(200);
    }
    break;
case 'i':
    {
        i = 0;
        t = 1;
        k1 = myservo1.read(); // διαβάζει τις τιμές των γωνιών και τις θέτει στην ανάλογη μεταβλητή
        k2 = myservo2.read();
        k3 = myservo3.read();
        while (t) {
            for (int k = 0; k < j; k++) { // επανάληψη j φορές
                if (k1 < M1[k]) { // εάν η τρέχουσα τιμή γωνίας του σερβοκινητήρα 1 είναι μικρότερη
                    από την τιμή του πίνακα 1
                        while (k1 < M1[k]) { // ο σερβοκινητήρας περιστρέφεται στη θέση όπου είναι
                            αποθηκευμένος στον πίνακα
                                myservo1.write(k1); // ο σερβοκινητήρας 1 εκτελεί την κίνηση
                                k1++; // K1 προσθέτει 1
                                delay(10); // καθυστέρηση για 10ms / έλεγχος της ταχύτητας περιστροφής του
                                    σερβοκινητήρα
                            }
                        }
                    }
                }
            }
        }
    }

```

```

    } else { // όταν η τιμή της γωνίας του σερβοκινητήρα 1 είναι μεγαλύτερη από την τιμή
που έχει αποθηκευτεί στον πίνακα 1
        while (k1 > M1[k]) { // ο σερβοκινητήρας περιστρέφεται στη θέση όπου είναι
αποθηκευμένος στον πίνακα
            myservo1.write(k1); // ο σερβοκινητήρας 1 εκτελεί την κίνηση
            k1--; // k1 αφαιρεί 1
            delay(10); // καθυστέρηση για 10ms / έλεγχος της ταχύτητας περιστροφής του
σερβοκινητήρα
        }
    }
// το ίδιο παρακάτω
if (k2 < M2[k]) {
    while (k2 < M2[k]) {
        myservo2.write(k2);
        k2++;
        delay(10);
    }
} else {
    while (k2 > M2[k]) {
        myservo2.write(k2);
        k2--;
        delay(10);
    }
}
if (k3 < M3[k]) {
    while (k3 < M3[k]) {
        myservo3.write(k3);
        k3++;
        delay(10);
    }
} else {
    while (k3 > M3[k]) {
        myservo3.write(k3);
        k3--;
        delay(10);
    }
}
}
if (Serial.read() == 't') { // έξοδος απο τον βρόχο while
    t = 0;
    break;
}
}
}
break;
}
}
}

```

```
void servo1on(){
```



```

myservo1.write(k1); // Περιστρέφει το K1
k1-=1;
delay(5);
If (k1<20) k1=20; // Ελάχιστη περιστροφή 20°
}

void servo1off(){
myservo1.write(k1); // Περιστρέφει το K1
k1+=1;
delay(5);
If (k1>80) k1=80; // Μέγιστη περιστροφή 80°
}


void servo2up(){
myservo2.write(k2); // Περιστρέφει το K2
k2+=1;
delay(10);
If (k2>120) k2=120; // Μέγιστη περιστροφή 120°
}

void servo2down(){
myservo2.write(k2); // Περιστρέφει το K2
k2-=1;
delay(10);
If (k2<10) k2=10; // Ελάχιστη περιστροφή 10°
}

void servo3left(){
myservo3.write(k3); // Περιστρέφει το K3
k3+=1;
delay(10);
if(k3>180)k3=180; // Μέγιστη περιστροφή 180°
}

void servo3right(){
myservo3.write(k3); // Περιστρέφει το K3
k3-=1;
delay(10);
if(k3<1)k3=0; // Ελάχιστη περιστροφή 0°
}

```

**Εναλλακτικά:** Μέσα στον φάκελο «ΚΩΔΙΚΕΣ ARDUINO» μπορείτε να ανοίξετε το αρχείο κάνοντας διπλό κλικ πάνω στο αρχείο  M\_1.52 .

#### (5) Αποτέλεσμα δοκιμής:

Ανεβάστε σωστά τον κώδικα χωρίς σφάλματα, έπειτα τοποθετήστε τη μονάδα Bluetooth και τρέξτε την εφαρμογή R4.

Τοποθετήστε ένα αντικείμενο αριστερά, ένα μπροστά και ένα δεξιά του ρομπότ.

Αποθηκεύστε τις κινήσεις: Συλλογή αντικειμένου από δεξιά και τοποθέτηση του στο κέντρο, συλλογή αντικειμένου από κέντρο και τοποθέτηση του αριστερά, συλλογή αντικειμένου από αριστερά και

τοποθέτηση του στο κέντρο. Πατήστε το εικονίδιο  για κάθε διαφορετική κίνηση του βραχίονα του ρομπότ.

Στη συνέχεια πατήστε το πλήκτρο



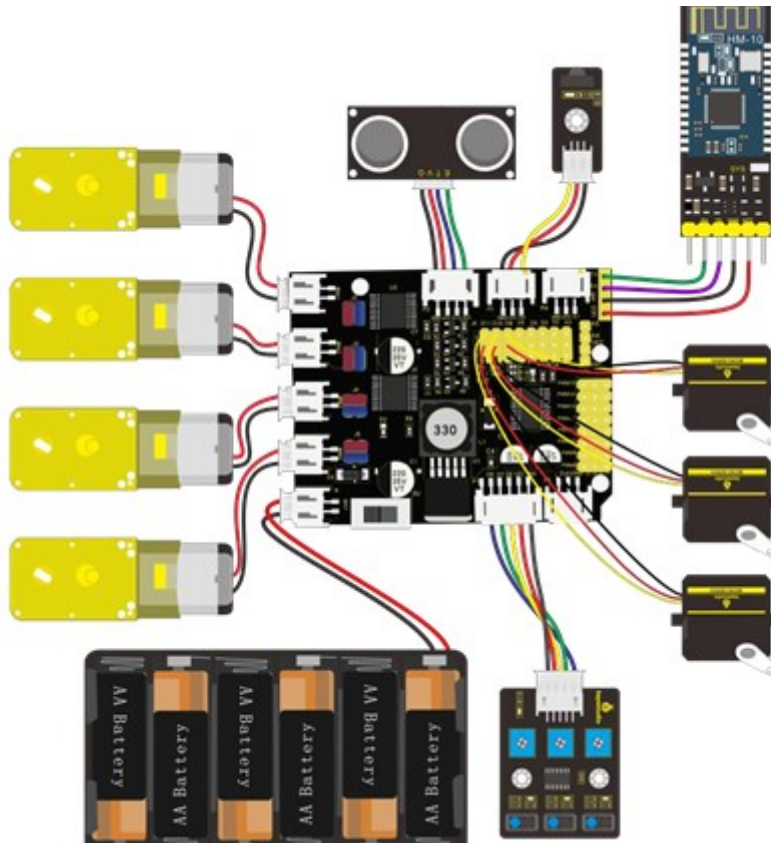
και δείτε το ρομπότ να επαναλαμβάνει τις κινήσεις που αποθηκεύσατε.

## Δραστηριότητα 1.53: Έξυπνο αυτοκίνητο ελέγχου ρομποτικού βραχίονα

### (1) Περιγραφή:

Μέχρι τώρα, έχουμε γνωρίσει πολλαπλές λειτουργίες του ρομπότ R4 και του μηχανικού βραχίονα. Μάλιστα, θα μπορούσαμε να τα συνδυάσουμε μαζί. Εφαρμόστε τον ακόλουθο κώδικα, για να κάνετε το ρομπότ R4 να επιδεικνύει κάθε λειτουργία που γνωρίσαμε.

### (2) Διάγραμμα καλωδίωσης:



Συνδέστε την πλακέτα οδήγησης στον υπολογιστή σας με το καλώδιο USB.

**ΣΗΜΕΙΩΣΗ:** 1. Δώστε προσοχή στην **κατεύθυνση** των ακίδων κατά την εισαγωγή της μονάδας Bluetooth

2. **ΜΗΝ** εισάγετε τη μονάδα Bluetooth πριν ανεβάσετε τον κώδικα. (Για να αποφύγετε την αντίφαση μεταξύ σειριακής επικοινωνίας κώδικα και επικοινωνίας Bluetooth)

### (3) Κώδικας δοκιμής:

/\*

Δραστηριότητα 8.53

Έξυπνο αυτοκίνητο ελέγχου ρομποτικού βραχίονα

\*/

```
#include <Wire.h>
```

```
#include <Adafruit_PWMServoDriver.h>
```

```
Adafruit_PWMServoDriver pwm = Adafruit_PWMServoDriver(0x47);
```

```
#include <Servo.h>
```

```
#include <IRremote.h>
```

```
int RECV_PIN = A0;
```

194

```

int echoPin = 13; // Ορισμός μονάδα υπερήχων ECHO στο D13
int trigPin = 12; // Ορισμός μονάδα υπερήχων TRIG στο D12

#define IR_Go 70
#define IR_Back 21
#define IR_Left 68
#define IR_Right 67
#define IR_Stop 64

#define SensorLeft 6 // καρφίτσα εισαγωγής του αριστερού αισθητήρα
#define SensorMiddle 7 // καρφίτσα εισαγωγής του μεσαίου αισθητήρα
#define SensorRight 8 // καρφίτσα εισαγωγής του δεξιού αισθητήρα
unsigned char SL; // κατάσταση αριστερού αισθητήρα
unsigned char SM; // κατάσταση μεσαίου αισθητήρα
unsigned char SR; // κατάσταση δεξιού αισθητήρα

Servo myservo1;
Servo myservo2;
Servo myservo3;

int k1 = 80, k2 = 120, k3 = 90; // αρχικοποίηση της τιμής γωνίας των σερβοκινητήρων
int M1[20], M2[20], M3[20];
int i = 0, j = 0, t = 0, speed = 2000, turnSpeed = 2000;
char blue_val;

void setup() {
  Serial.begin(9600);
  pwm.begin();
  pwm.setPWMPfreq(60);
  myservo1.attach(11);
  myservo2.attach(10);
  myservo3.attach(9);
  myservo1.write(k1);
  delay(1000);
  myservo2.write(k2);
  delay(1000);
  myservo3.write(k3);
  delay(1000);
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  pinMode(SensorLeft, INPUT);
  pinMode(SensorMiddle, INPUT);
  pinMode(SensorRight, INPUT);
  stop();
  IrReceiver.begin(RECV_PIN); // ενεργοποίηση του δέκτη IR
}

void loop() {
  if (Serial.available() > 0) { // Λήψη σημάτων Bluetooth
    blue_val = Serial.read();

```

```

Serial.println(blue_val);
switch (blue_val) {
  case 'F':
    advance();
    Serial.println("advance");
    break; // Αν λάβει εντολή 'F', κάνει κίνηση μπροστά

  case 'B':
    back();
    Serial.println("back");
    break; // Αν λάβει εντολή 'B', κάνει κίνηση πίσω
  case 'L':
    turnL();
    Serial.println("turn left");
    break; // Αν λάβει εντολή 'L', στροφή αριστερά
  case 'R':
    turnR();
    Serial.println("turn right");
    break; // Αν λάβει εντολή 'R', στροφή δεξιά
  case 'S':
    stop();
    Serial.println("stop");
    break; // Αν λάβει εντολή 'S', σταματάει
  case 'a': speeds_a(); break; // Αν λάβει εντολή 'a', επιταχύνει

  case 'd': speeds_d(); break; // Αν λάβει εντολή 'd', επιβραδύνει

  case 'f': servo2up(); break; // // Ο βραχίονας ανυψώνεται

  case 'b': servo2down(); break; // // Ο βραχίονας χαμηλώνει

  case 'l': servo3left(); break; // Ο βραχίονας στρέφεται προς τα αριστερά

  case 'r': servo3right(); break; // Ο βραχίονας στρέφεται προς τα δεξιά

  case 'Q': servo1on(); break; // Η δαγκάνα ανοίγει

  case 'E': servo1off(); break; // Η δαγκάνα κλείνει

  case 't': read_servo(); break; // Αν λάβει εντολή 't', καταγράφει κίνηση

  case 'i': do_servo(); break; // Αν λάβει εντολή 'i', εκτελεί κίνηση

  case 'Y':
    avoid();
    Serial.println("avoid");
    break; // Αν λάβει εντολή 'Y' είσοδο στη λειτουργία αποφυγής εμποδίων

  case 'X':
    tracking();

```

```

    Serial.println("tracking");
    break; // Αν λάβει εντολή 'X' είσοδος σε λειτουργία παρακολούθησης γραμμής

case 'U':
    follow();
    Serial.println("follow");
    break; // Αν λάβει εντολή 'U' είσοδος στη λειτουργία παρακολούθησης υπερήχων
case 'G':
    Fall();
    Serial.println("Fall");
    break; // Αν λάβει εντολή 'G' εισέλθει σε κατάσταση αντι-πτώσης

case 'H':
    Home();
    Serial.println("Fall");
default: break;
}
}

if (IrReceiver.decode()) { // Λήψη σημάτων IR
    switch (IrReceiver.decodedIRData.command) {
        case IR_Go: advance(); break;
        case IR_Back: back(); break;
        case IR_Left: turnL(); break;
        case IR_Right: turnR(); break;
        case IR_Stop: stop(); break;
        default: break;
    }
    IrReceiver.resume();
}
}

// Μέθοδοι κίνησης//

void advance() // Κίνηση μπροστά
{
    pwm.setPWM(0, 0, speed);
    pwm.setPWM(1, 0, 0);
    pwm.setPWM(2, 0, speed);
    pwm.setPWM(3, 0, 0);
    pwm.setPWM(4, 0, speed);
    pwm.setPWM(5, 0, 0);
    pwm.setPWM(6, 0, speed);
    pwm.setPWM(7, 0, 0);
}

void turnR() // Στροφή δεξιά
{
    pwm.setPWM(0, 0, turnSpeed);
    pwm.setPWM(1, 0, 0);
}

```

```

    pwm.setPWM(2, 0, turnSpeed);
    pwm.setPWM(3, 0, 0);
    pwm.setPWM(4, 0, 0);
    pwm.setPWM(5, 0, turnSpeed);
    pwm.setPWM(6, 0, 0);
    pwm.setPWM(7, 0, turnSpeed);
}

void turnL() // Στροφή αριστερά
{
    pwm.setPWM(0, 0, 0);
    pwm.setPWM(1, 0, turnSpeed);
    pwm.setPWM(2, 0, 0);
    pwm.setPWM(3, 0, turnSpeed);
    pwm.setPWM(4, 0, turnSpeed);
    pwm.setPWM(5, 0, 0);
    pwm.setPWM(6, 0, turnSpeed);
    pwm.setPWM(7, 0, 0);
}

void stop() // Σταματάει
{
    pwm.setPWM(0, 0, 0);
    pwm.setPWM(1, 0, 0);
    pwm.setPWM(2, 0, 0);
    pwm.setPWM(3, 0, 0);
    pwm.setPWM(4, 0, 0);
    pwm.setPWM(5, 0, 0);
    pwm.setPWM(6, 0, 0);
    pwm.setPWM(7, 0, 0);
}

void back() // Κίνηση προς τα πίσω
{
    pwm.setPWM(0, 0, 0);
    pwm.setPWM(1, 0, speed);
    pwm.setPWM(2, 0, 0);
    pwm.setPWM(3, 0, speed);
    pwm.setPWM(4, 0, 0);
    pwm.setPWM(5, 0, speed);
    pwm.setPWM(6, 0, 0);
    pwm.setPWM(7, 0, speed);
}

//Μεθοδοι ελέγχου ταχύτητας//

//Επιτάχυνση
void speeds_a() {
    bool a_flag = 1;
    while (a_flag) {

```

```

    Serial.println(speed);
    if (speed <= 2555) { // Αυξάνει την ταχύτητα κατά 5, μέχρι να φτάσει 2555
        speed += 5;
        delay(2); // Αλλάξτε το χρόνο καθυστέρησης για να αλλάξετε την ταχύτητα του έξυπνου
        αυτοκινήτου
    }
    if (Serial.read() == 'S') a_flag = 0; // 'S' για να σταματήσει η επιτάχυνση
}
}

//Επιβράδυνση
void speeds_d() {
    bool d_flag = 1;
    while (d_flag) {
        Serial.println(speed);
        if (speed >= 5) {
            speed -= 5;
            delay(2);
        }
        if (Serial.read() == 'S') d_flag = 0;
    }
}

//Μέθοδοι ελέγχου βραχίονα//
void servo1on() { // Η δαγκάνα ανοίγει
    bool servo1on_flag = 1;
    while (servo1on_flag) {
        k1 -= 1;
        myservo1.write(k1);
        delay(5);
        if (k1 < 20) {
            k1 = 20;
        }
        if (Serial.read() == 's') servo1on_flag = 0;
    }
}

void servo1off() { // Η δαγκάνα κλείνει
    bool servo1off_flag = 1;
    while (servo1off_flag) {
        k1 += 1; // έλεγχος της ακρίβειας περιστροφής
        myservo1.write(k1);
        delay(5); // έλεγχος της ταχύτητας περιστροφής του σερβοκινητήρα
        if (k1 > 80) {
            k1 = 80;
        }
        if (Serial.read() == 's') servo1off_flag = 0;
    }
}
}

```



```

void servo2up() { // ο ρομποτικός βραχίονας ανυψώνεται
  bool servo2up_flag = 1;
  while (servo2up_flag) {
    k2 += 1; // έλεγχος της ακρίβειας περιστροφής
    myservo2.write(k2);
    delay(10); // έλεγχος της ταχύτητας περιστροφής του σερβοκινητήρα
    if (k2 > 120) {
      k2 = 120;
    }
    if (Serial.read() == 's') servo2up_flag = 0;
  }
}

void servo3up() { // ο ρομποτικός βραχίονας ανυψώνεται
  k2 += 1; // έλεγχος της ακρίβειας περιστροφής
  myservo2.write(k2);
  delay(10); // έλεγχος της ταχύτητας περιστροφής του σερβοκινητήρα
}

void servo2down() { // ο ρομποτικός βραχίονας χαμηλώνει
  bool servo2down_flag = 1;
  while (servo2down_flag) {
    k2 -= 1; // έλεγχος της ακρίβειας περιστροφής
    myservo2.write(k2);
    delay(10); // έλεγχος της ταχύτητας περιστροφής του σερβοκινητήρα
    if (k2 < 10) {
      k2 = 10;
    }
    if (Serial.read() == 's') servo2down_flag = 0;
  }
}

void servo3left() { // ο ρομποτικός βραχίονας στρέφεται αριστερόστροφα
  bool servo3left_flag = 1;
  while (servo3left_flag) {
    k3 += 1; // έλεγχος της ακρίβειας περιστροφής
    myservo3.write(k3);
    delay(10); // έλεγχος της ταχύτητας περιστροφής του σερβοκινητήρα
    if (k3 > 180) {
      k3 = 180;
    }
    if (Serial.read() == 's') servo3left_flag = 0;
  }
}

void servo3right() { // ο ρομποτικός βραχίονας στρέφεται δεξιόστροφα
  bool servo3right_flag = 1;
  while (servo3right_flag) {
    k3 -= 1; // έλεγχος της ακρίβειας περιστροφής
    myservo3.write(k3);

```

```

    delay(10); // έλεγχος της ταχύτητας περιστροφής του σερβοκινητήρα
    if (k3 < 1) {
        k3 = 0;
    }
    if (Serial.read() == 's') servo3right_flag = 0;
}
}

//Μέθοδοι Μνήμης//
void read_servo() {
    M1[i] = myservo1.read(); // αποθήκευση της τιμής της γωνίας των σερβοκινητήρων σε
πίνακα
    delay(100); // χρόνος καθυστέρησης για να αποθηκευτεί η τιμή της γωνίας
    M2[i] = myservo2.read();
    delay(100);
    M3[i] = myservo3.read();
    delay(100);
    i++; // i προσθέτει 1 κάθε φορά που αποθηκεύεται i
    j = i; // ορίστε την τιμή του i στο j
    delay(200);
}

void do_servo() {
    i = 0;
    t = 1;
    k1 = myservo1.read(); // διαβάζει τις τιμές των γωνιών και τις θέτει στην ανάλογη
μεταβλητή
    k2 = myservo2.read();
    k3 = myservo3.read();

    while (t) {
        for (int k = 0; k < j; k++) { // επανάληψη j φορές
            if (k1 < M1[k]) { // εάν η τρέχουσα τιμή γωνίας του σερβοκινητήρα 1 είναι
μικρότερη από την τιμή του πίνακα 1
                while (k1 < M1[k]) { // ο σερβοκινητήρας περιστρέφεται στη θέση όπου είναι
αποθηκευμένος στον πίνακα
                    myservo1.write(k1); // ο σερβοκινητήρας 1 εκτελεί την κίνηση
                    k1++; // k1 προσθέτει 1
                    delay(10); // καθυστέρηση για 10ms / έλεγχος της ταχύτητας
περιστροφής του σερβομηχανισμού
                }
            } else { // όταν η τιμή της γωνίας του σερβοκινητήρα 1 είναι
μεγαλύτερη από την τιμή που έχει αποθηκευτεί στον πίνακα 1
                while (k1 > M1[k]) { // ο σερβοκινητήρας περιστρέφεται στη θέση όπου είναι
αποθηκευμένος στον πίνακα
                    myservo1.write(k1); // ο σερβοκινητήρας 1 εκτελεί την κίνηση
                    k1--; // k1 αφαιρεί 1
                    delay(10); // καθυστέρηση για 10ms / έλεγχος της ταχύτητας περιστροφής
του σερβοκινητήρα
                }
            }
        }
    }
}

```

```

}
// το ίδιο παρακάτω
if (k2 < M2[k]) {
  while (k2 < M2[k]) {
    myservo2.write(k2);
    k2++;
    delay(10);
  }
} else {
  while (k2 > M2[k]) {
    myservo2.write(k2);
    k2--;
    delay(10);
  }
}

if (k3 < M3[k]) {
  while (k3 < M3[k]) {
    myservo3.write(k3);
    k3++;
    delay(10);
  }
} else {
  while (k3 > M3[k]) {
    myservo3.write(k3);
    k3--;
    delay(10);
  }
}

if (Serial.available() > 0) { // για την έξοδο από τον βρόχο
  if (Serial.read() == 't') { // λαμβάνει 'i' τότε 't'
    t = 0; // θέτει το 't' σε 0, έξοδος από τον βρόχο
    break;
  }
}
}
}
}
}

```

//Μεθοδος αποφυγής εμποδίων//

```

int Ultrasonic_Ranging() {
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  int distance = pulseIn(echoPin, HIGH); // Ανάγνωση της διάρκειας των υψηλών επιπέδων
  distance = distance / 58; // Μετατροπή του χρόνου παλμού σε απόσταση
}

```

```

    delay(50);
    return distance;
}

void avoid() { //αυτόνομη κίνηση, αποφυγή εμποδίων με υπερήχους
    bool avoid_flag = 1;
    while (avoid_flag) {
        int distance = Ultrasonic_Ranging();
        Serial.print("distance=");
        Serial.println(distance);
        if (distance < 30) { // υποθέτοντας ότι η μπροστινή απόσταση είναι μικρότερη από 30
cm
            if (distance < 15) { // υποθέτοντας ότι η μπροστινή απόσταση είναι μικρότερη από 15
cm
                stop();
                delay(100);
                back();
                delay(300);
            } else { // όταν 10 < απόσταση < 30, στροφή δεξιά
                stop();
                delay(100);
                turnR();
                delay(500);
            }
        } else { // όταν η απόσταση > 30, κίνηση μπροστά
            advance();
        }
        blue_val = Serial.read();
        if (blue_val == 'S') {
            stop();
            avoid_flag = 0;
        }
    }
}

//Μέθοδος παρακολούθησης γραμμής
void tracking(void) {
    bool tracking_flag = 1;
    while (tracking_flag) {
        SL = digitalRead(SensorLeft);
        SM = digitalRead(SensorMiddle);
        SR = digitalRead(SensorRight);
        if (SM == HIGH) {
            if (SL == LOW && SR == HIGH) { // Αν ανιχνευτεί μαύρο στα δεξιά και λευκό στα
αριστερά, στρίβει δεξιά
                turnR();
            } else if (SR == LOW && SL == HIGH) { // αν ανιχνευτεί μαύρο στα αριστερά και λευκό
στα δεξιά, στρίβει αριστερά
                turnL();
            } else { // Αν ανιχνευτεί λευκό και στις δύο άκρες, κινείται ευθεία

```

```

        advance();
    }
} else {
    if (SL == LOW && SR == HIGH) { // Αν ανιχνευτεί μαύρο στα δεξιά και λευκό στα
αριστερά, στρίβει δεξιά
        turnR();
    } else if (SR == LOW && SL == HIGH) { // Αν ανιχνευτεί μαύρο στα αριστερά και λευκό
στα δεξιά, στρίβει αριστερά
        turnL();
    } else { // Αν όλοι οι αισθητήρες ανιχνεύουν λευκό, το αυτοκίνητο σταματάει
        stop();
    }
}
blue_val = Serial.read();
if (blue_val == 'S') {
    stop();
    tracking_flag = 0;
}
}
}

```

//Μέθοδος αισθητήρα υπέρηχων

```

void follow() {
    bool follow_flag = 1;
    while (follow_flag) {
        int distance = Ultrasonic_Ranging();
        Serial.print("distance=");
        Serial.println(distance);
        if (distance < 15 && distance != 0) { // Αν η απόσταση είναι μικρότερη απο 15
εκατοστά, κίνηση προς τα πίσω
            back();
        } else if (distance >= 15 && distance < 20) { // Αν η απόσταση είναι μεταξύ 15 και 20
εκατοστά, σταμάτα
            stop();
        } else if (distance >= 20 && distance < 40) { // Αν η απόσταση είναι μεταξύ 20 και 40
εκατοστά, κίνηση μπροστά
            advance();
        } else {
            stop(); // αλλιώς σταμάτα
        }
        blue_val = Serial.read();
        if (blue_val == 'S') {
            stop();
            follow_flag = 0;
        }
    }
}
}

```

```


void Home(){
k1 = 80, k2 = 120, k3 = 90;

```

```
myservo1.attach(11);
myservo2.attach(10);
myservo3.attach(9);
myservo1.write(k1);
delay(1000);
myservo2.write(k2);
delay(1000);
myservo3.write(k3);
delay(1000);
}
```

//Μέθοδος αντι-πτώσης

```
void Fall(void) {
  bool Fall_flag = 1;
  while (Fall_flag) {
    SL = digitalRead(SensorLeft);
    SM = digitalRead(SensorMiddle);
    SR = digitalRead(SensorRight);
    if (SM == HIGH || SL == HIGH || SR == HIGH) {
      back();
      delay(500);
      turnL();
      delay(500);
    } else {
      advance();
    }
  }
  blue_val = Serial.read();
  if (blue_val == 'S') {
    stop();
    Fall_flag = 0;
  }
}
}
```






**Εναλλακτικά:** Μέσα στον φάκελο «ΚΩΔΙΚΕΣ ARDUINO» μπορείτε να ανοίξετε το αρχείο κάνοντας διπλό κλικ πάνω στο αρχείο  M\_1.53 .

#### (4) Αποτέλεσμα δοκιμής:




Ανεβάστε σωστά τον κώδικα χωρίς σφάλματα, έπειτα τοποθετήστε τη μονάδα Bluetooth και τρέξτε την εφαρμογή R4. Τώρα μπορείτε να κατευθύνετε τόσο το ρομποτικό βραχίονα όσο και το αυτοκίνητο προς οποιαδήποτε κατεύθυνση θέλετε. Μπορείτε επίσης να χρησιμοποιήσετε όλες τις λειτουργίες της εφαρμογής (έλεγχος ταχύτητας, μνήμη, παρακολούθηση γραμμής, αποφυγή

εμποδίων κτλ). Τώρα που έχετε τον πλήρη έλεγχο των δυνατοτήτων του αυτοκινήτου μπορείτε να σκεφτείτε διάφορες κινήσεις και παραδείγματα για να εξοικειωθείτε με τη χρήση του. Πιο κάτω μπορείτε να δείτε τον πίνακα με τις λειτουργίες των διάφορων κουμπιών της εφαρμογής.

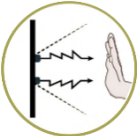
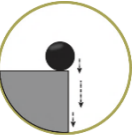

	Χαρακτήρας Ελέγχου Κώδικα	Λειτουργία
	Άνοιγμα: F Κλείσιμο: S	Πατήστε το κουμπί, το ρομπότ πηγαίνει μπροστά. Αφήστε το για να σταματήσει.
	Άνοιγμα: L Κλείσιμο: S	Πατήστε το κουμπί, το ρομπότ στρίβει αριστερά. Αφήστε το για να σταματήσει.
	Άνοιγμα: R Κλείσιμο: S	Πατήστε το κουμπί, το ρομπότ στρίβει δεξιά. Αφήστε το για να σταματήσει.
	Άνοιγμα: B Κλείσιμο: S	Πατήστε το κουμπί, το ρομπότ πηγαίνει προς τα πίσω. Αφήστε το για να σταματήσει.
 Επιτάχυνση	Άνοιγμα: a Κλείσιμο: S	Πατήστε για επιτάχυνση. Το ρομπότ θα κινηθεί με μεγαλύτερη ταχύτητα. Αφήστε το για να σταματήσει.
 Επιβράδυνση	Άνοιγμα: d Κλείσιμο: S	Πατήστε για να επιβραδύνετε. Το ρομπότ θα κινηθεί με μικρότερη ταχύτητα. Αφήστε το για να σταματήσει.
	Άνοιγμα: Q Κλείσιμο: S	Πατήστε για να ανοίξει η δαγκάνα. Αφήστε για να σταματήσει.
	Άνοιγμα: E Κλείσιμο: S	Πατήστε για να κλείσει η δαγκάνα. Αφήστε το για να σταματήσει.
	Άνοιγμα: f Κλείσιμο: s	Πιέστε για να κινήσετε τον βραχίονα προς τα μπροστά. Αφήστε το για να σταματήσει.

	Άνοιγμα: b Κλείσιμο: s	Πατήστε για να κινήσετε τον βραχίονα προς τα πίσω. Αφήστε το για να σταματήσει.
	Άνοιγμα: l Κλείσιμο: s	Πατήστε για να στρίψει ο βραχίονας αριστερά. Αφήστε το για να σταματήσει.
	Άνοιγμα: r Κλείσιμο: s	Πατήστε για να στρίψει ο βραχίονας δεξιά. Αφήστε το για να σταματήσει.
	Άνοιγμα: t Κλείσιμο: s	Πατήστε το εικονίδιο για να αποθηκεύσετε την τρέχουσα τιμή γωνίας του βραχίονα (εγγραφή κίνησης).
	Άνοιγμα: i Κλείσιμο: s	Πιέστε για να εκτελέσετε την αποθηκευμένη τιμή της γωνίας του βραχίονα (εκτέλεση κίνησης).

## ΕΠΙΠΛΕΟΝ ΛΕΙΤΟΥΡΓΙΕΣ

 Αίσθηση βαρύτητας	Κάντε κλικ για να ξεκινήσει η κινητή ανίχνευση βαρύτητας. Κινήστε το κινητό σας αλλάζοντας την κατεύθυνση του κινητού πάνω, κάτω, αριστερά και δεξιά και οδηγήστε το ρομπότ. Κάντε ξανά κλικ για έξοδο.	
 Αυτόματη μεταφορά	Κάντε κλικ για να επαναφέρεται τον βραχίονα του ρομπότ στην αρχική του θέση. Κουμπί reset.	
 Αποφυγή εμποδίου	Κάντε κλικ για αποστολή "Y" και μετά κάντε κλικ στο "S"	Πατήστε το εικονίδιο για να εισέλθετε στη λειτουργία αποφυγής εμποδίων και πατήστε ξανά για έξοδο.



 <p>Ακολούθηση εμποδίου</p>	<p>Κάντε κλικ για να στείλετε το "U" και μετά κάντε κλικ στο "S"</p>	<p>Εκκίνηση της λειτουργίας παρακολούθησης υπερήχων. Πατήστε ξανά για έξοδο.</p>
 <p>Αποφυγή πτώσης</p>	<p>Κάντε κλικ για αποστολή "G" και μετά κάντε κλικ στο "S"</p>	<p>Πατήστε το εικονίδιο για να ενεργοποιήσετε τη λειτουργία προστασίας από πτώση, πατήστε ξανά για έξοδο.</p>
 <p>Ιχνηλάτηση</p>	<p>Κάντε κλικ για να στείλετε το "X" και μετά κάντε κλικ στο "S"</p>	<p>Πατήστε το εικονίδιο για να ενεργοποιήσετε τη λειτουργία παρακολούθησης γραμμής, πατήστε ξανά για να τερματίσετε</p>