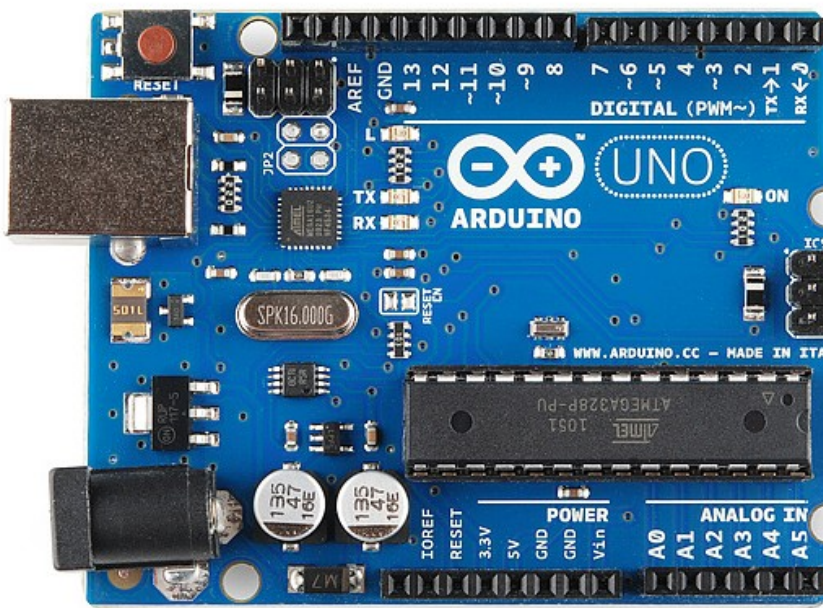


2017

# Κατασκευάζω και Προγραμματίζω με τον μικροελεγκτή Arduino



Αριστείδης Παλιούρας

e-mail: [arispaliouras@gmail.com](mailto:arispaliouras@gmail.com)

ISBN: 978-960-93-8945-7

Κατασκευάζω και Προγραμματίζω με τον μικροελεγκτή Arduino

Copyright © 2017 Αριστείδης Παλιούρας

**ISBN:** 978-960-93-8945-7

**Συγγραφέας:** Αριστείδης Παλιούρας

**Τηλ.** 694 622 5477

**e-mail:** arispaliouras@gmail.com

**Δικτυακός τόπος:** [www.robotics-edu.gr](http://www.robotics-edu.gr)

Ο εκδότης διατηρεί όλα τα πνευματικά του δικαιώματα. Κανένα μέρος αυτού του βιβλίου δεν επιτρέπεται να αναπαραχθεί, να διατηρηθεί σε κάποιο σύστημα ανάκτησης, ή να μεταδοθεί από κάποιο μέσο, ηλεκτρονικό, μηχανικό, φωτοτυπικό, ή όποιο άλλο μέσο, χωρίς την ενυπόγραφη άδεια του εκδότη. Μονολότι έχουν ληφθεί όλες οι απαραίτητες προφυλάξεις κατά την προετοιμασία αυτού του βιβλίου, ο εκδότης και συγγραφέας δεν φέρουν ευθύνη για λάθη ή παραλείψεις. Επίσης, δεν αναλαμβάνουν ευθύνη για οποιαδήποτε ζημιά προκληθεί από τη χρήση των πληροφοριών που περιέχονται σε αυτό το βιβλίο.

## Περιεχόμενα

Φύλλο εργασίας 1 - Λαμπάκι (LED) που αναβοσβήνει .....	4
Φύλλο εργασίας 2 - Φωτεινός σηματοδότης .....	9
Φύλλο εργασίας 3 - Ρύθμιση της φωτεινότητας ενός LED με ποτενσιόμετρο.....	13
Φύλλο εργασίας 4 - Αυτόματο φωτάκι νυκτός .....	16
Φύλλο εργασίας 5 - Χριστουγεννιάτικα φωτάκια (σταδιακή αύξηση και μείωση φωτεινότητας ενός LED) .....	21
Φύλλο εργασίας 6 - Θερμόμετρο εξωτερικού χώρου .....	25
Φύλλο εργασίας 7 - Δημιουργώ τα δικά μου χρώματα με το RGB LED .....	30
Φύλλο εργασίας 8 - Ανιχνευτής απόστασης .....	36
Φύλλο εργασίας 9 - Αυτόνομο ρομποτικό όχημα αποφυγής εμποδίων .....	41

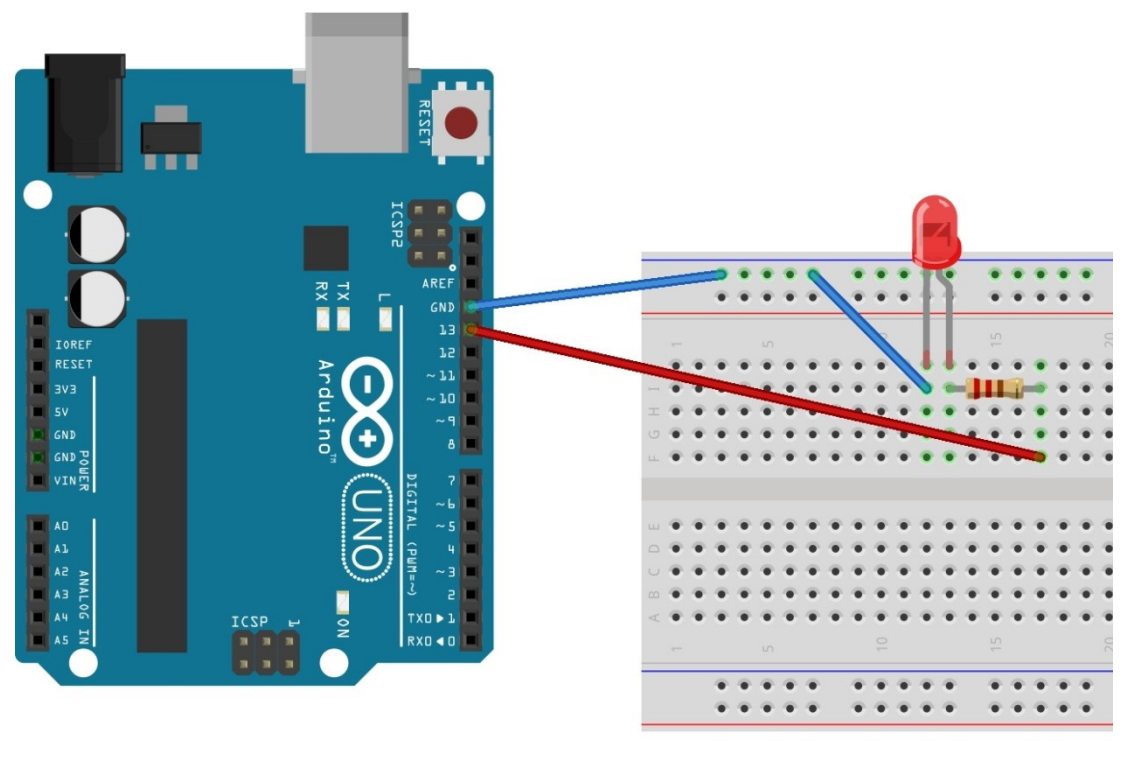
## Φύλλο εργασίας 1 - Λαμπάκι (LED) που αναβοσβήνει

Στην πρώτη μας δραστηριότητα θα συνδέσουμε ένα LED με την πλακέτα Arduino. Στη συνέχεια με το προγραμματιστικό περιβάλλον Ardublock θα προγραμματίσουμε τον μικροελεγκτή Arduino να αναβοσβήνει το LED ανά ένα δευτερόλεπτο.

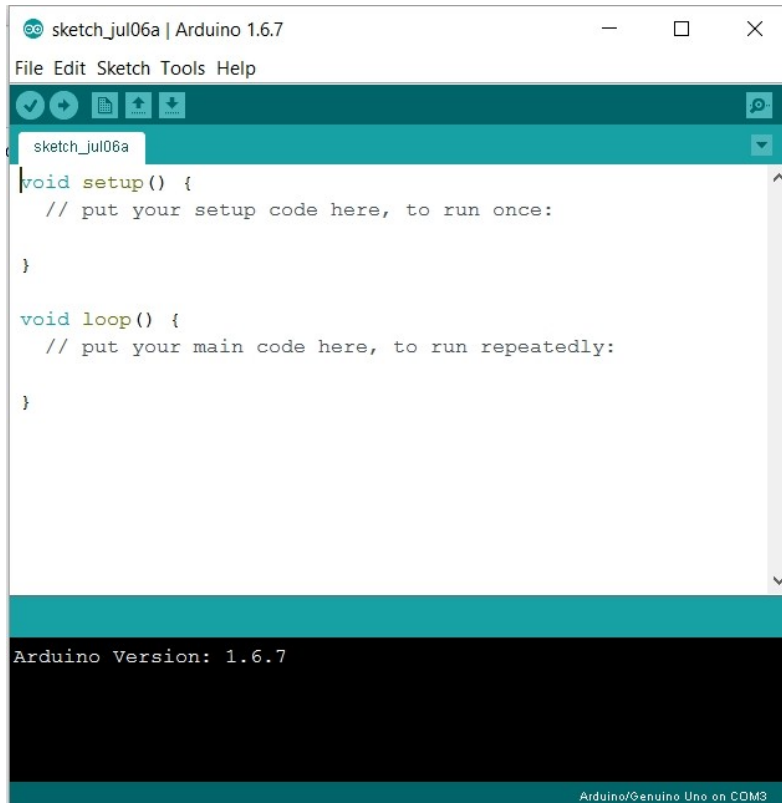
### Υλικά που θα χρειαστούμε:

- ✓ 1 Arduino UNO
- ✓ 1 Breadboard
- ✓ 1 LED Κόκκινο
- ✓ 1 αντίσταση 220 Ohm

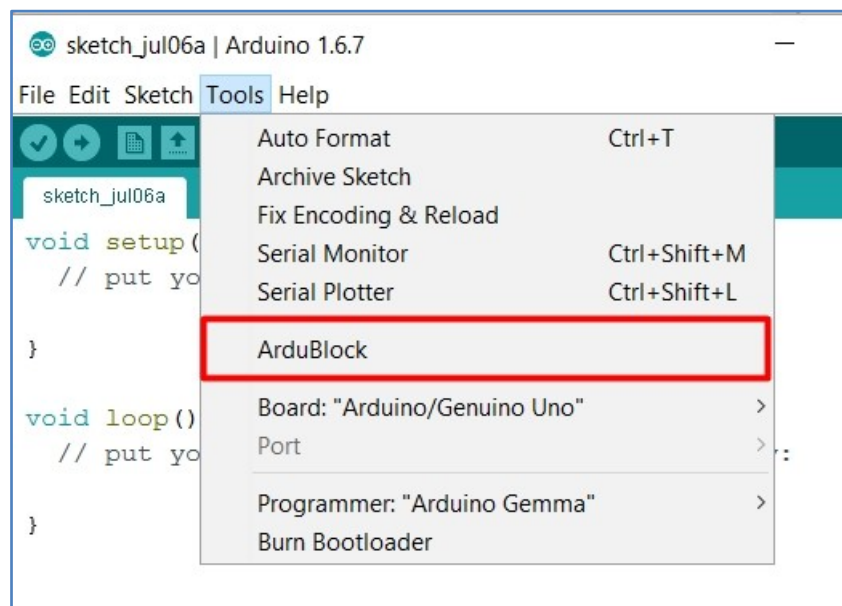
### Το κύκλωμα σε breadboard



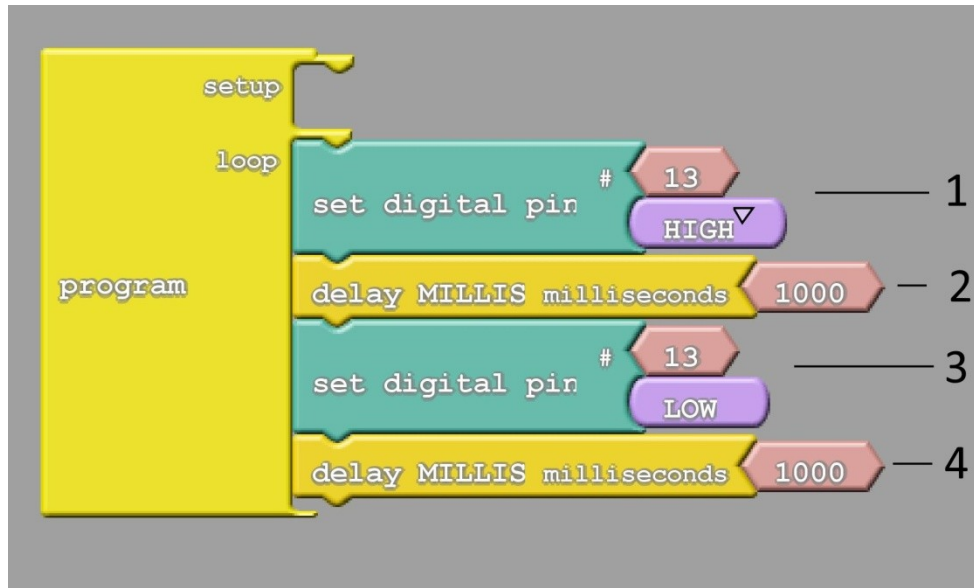
**1** Ανοίξτε το προγραμματιστικό περιβάλλον του Arduino.



**2** Από το μενού **Tools** επιλέξτε **ArduBlock** για να ξεκινήσετε το προγραμματιστικό περιβάλλον του ArduBlock.





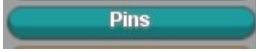



**3** Δημιουργήστε το παρακάτω πρόγραμμα στο Ardublock



\* Οι αριθμοί δίπλα σε κάθε εντολή χρησιμοποιούνται στην ανάλυση του προγράμματος που ακολουθεί σε επόμενη παράγραφο.

Εικόνα 1 - program1.adp

Στον παρακάτω πίνακα μπορείτε να δείτε σε ποια ομάδα εντολών ανήκει κάθε εντολή για να την εντοπίσετε πιο εύκολα.

Ομάδα εντολών	Εντολή
	
	
	

**Ανάλυση προγράμματος (program1.adp)**

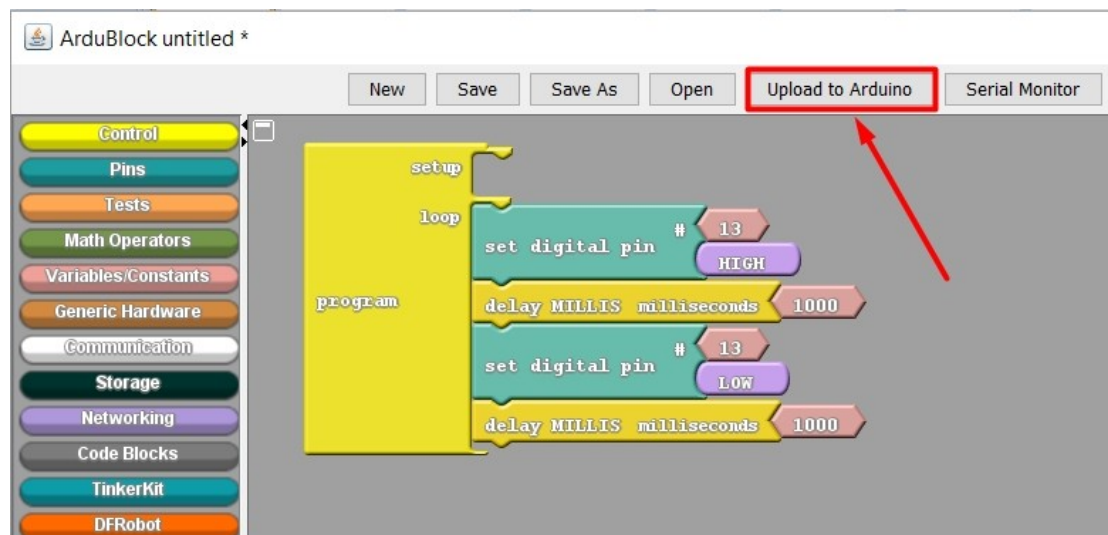
\* Κάθε γραμμή αντιστοιχεί σε μια εντολή (με τον ίδιο αριθμό) στην Εικόνα 1.

1. Η εντολή **set digital pin(13, HIGH)** στέλνει μια τάση 5 V στην ψηφιακή θύρα 13. Αυτό έχει σαν αποτέλεσμα να ανάψει το LED.
2. Με την εντολή **delay MILLIS milliseconds(1000)** σταματά η εκτέλεση του προγράμματος για χρονικό διάστημα 1000 ms ( 1 sec = 1000 ms ).

3. Η εντολή **set digital pin(13, LOW)**, στέλνει μια τάση 0 V στην ψηφιακή θύρα 13. Αυτό έχει σαν αποτέλεσμα να σβήσει το LED.
4. Με την εντολή **delay MILLIS milliseconds(1000)** σταματά η εκτέλεση του προγράμματος για χρονικό διάστημα 1000 ms.

Όλες οι παραπάνω εντολές βρίσκονται μέσα στην **loop**. Η **loop** είναι μια συνάρτηση την οποία εκτελεί ο μικροελεγκτής Arduino συνεχώς. Όταν ενεργοποιηθεί ο μικροελεγκτής Arduino θα καλέσει την συνάρτηση **setup** μια φορά και στη συνέχεια την συνάρτηση **loop**. Μόλις ολοκληρωθεί η εκτέλεση των εντολών της **loop**, ο μικροελεγκτής θα ξανακαλέσει την συνάρτηση **loop**. Αυτό θα γίνεται συνεχώς, όσο έχει ρεύμα ο μικροελεγκτής ή μέχρι να πατηθεί το πλήκτρο **reset**. Αν πατήσουμε το πλήκτρο reset του μικροελεγκτή τότε θα τρέξει η συνάρτηση **setup** μια φορά και μετά θα εκτελείται η **loop** συνεχώς.

**4** Κάντε κλικ στο κουμπί "**Upload to Arduino**" για να φορτώσετε το πρόγραμμα στον μικροελεγκτή Arduino. Αν έχουν γίνει όλα σωστά, το LED θα αναβοσβήνει ανά ένα δευτερόλεπτο.



## Δραστηριότητες

1. Μετακινήστε τις εντολές από την περιοχή loop στην περιοχή setup και φορτώστε το πρόγραμμα στο Arduino. Τι παρατηρείτε ;
2. Προσπαθήστε να προσθέσετε άλλο ένα LED στο παραπάνω κύκλωμα και να το κάνετε να αναβοσβήνει ανά ένα δευτερόλεπτο. Χρησιμοποιήστε την θύρα 12 του Arduino.
3. Τροποποιήστε το πρόγραμμα της προηγούμενης δραστηριότητας ώστε τα LED να αναβοσβήνουν ανά 5 δευτερόλεπτα.

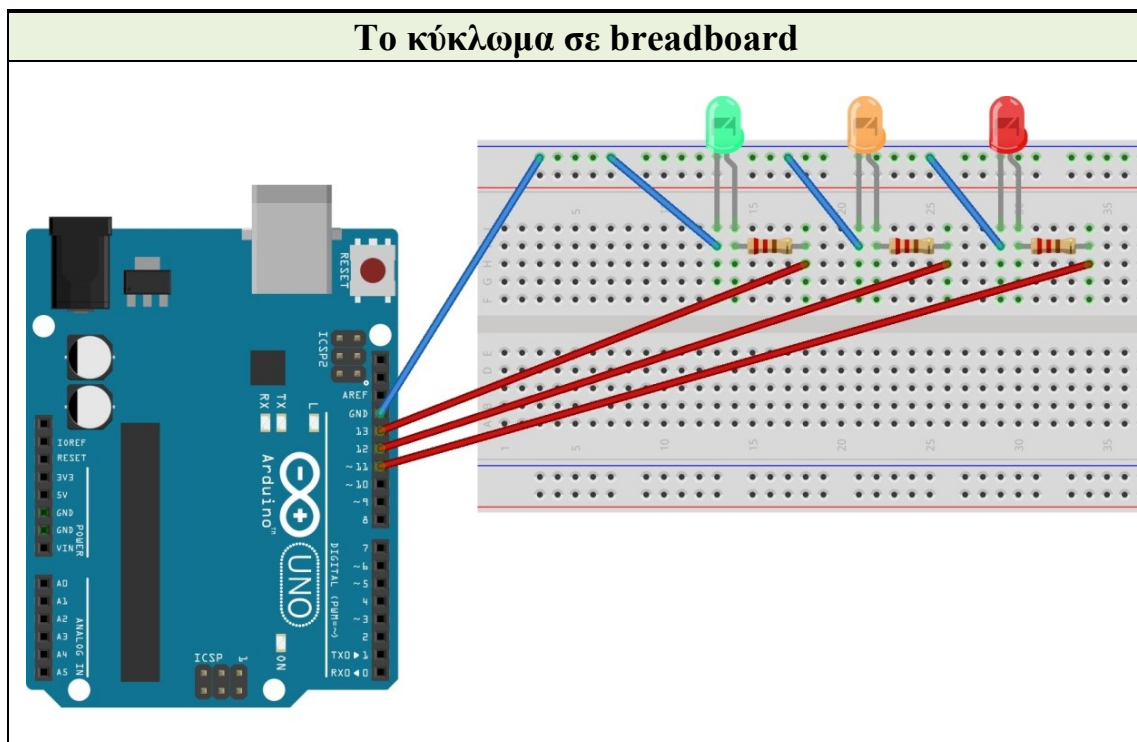


## Φύλλο εργασίας 2 - Φωτεινός σηματοδότης

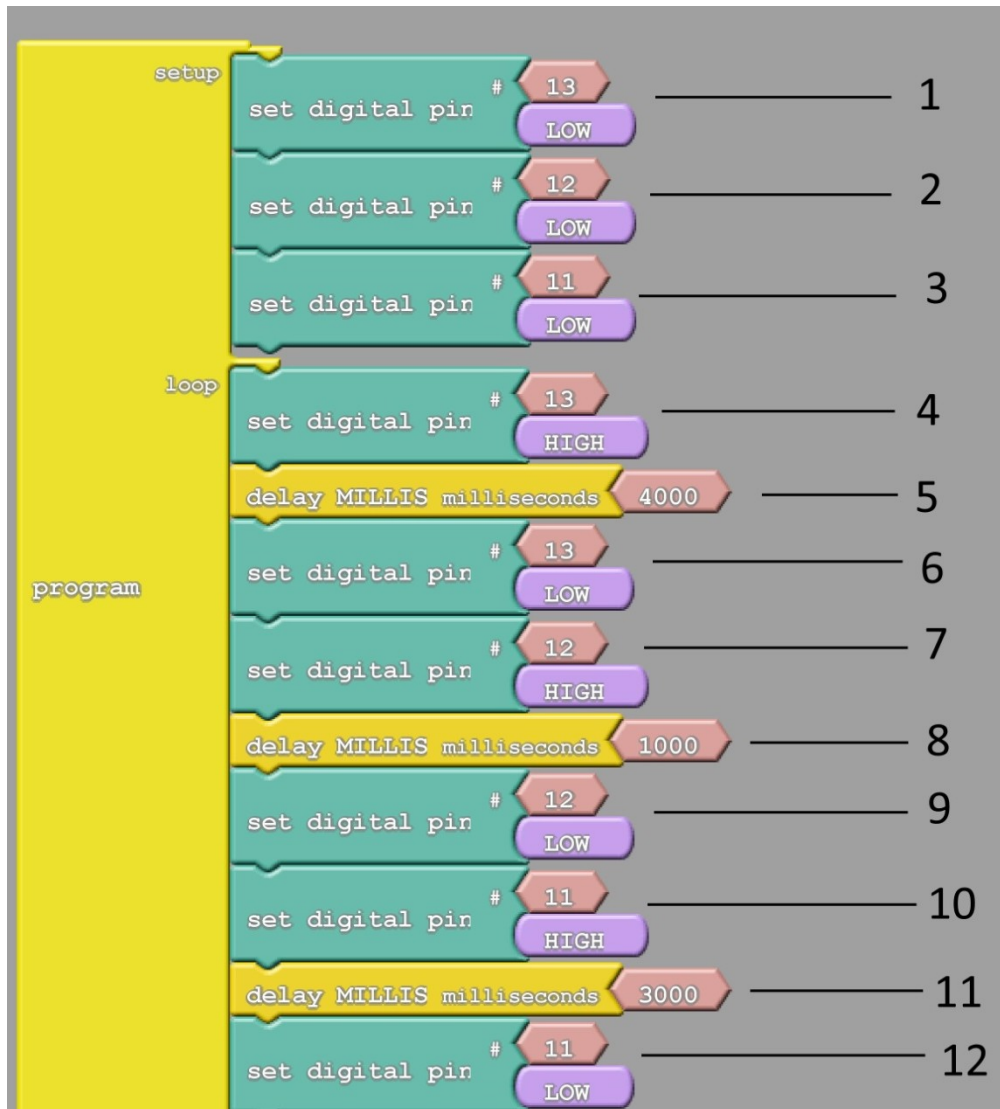
Στην δραστηριότητα αυτή θα προσομοιώσουμε τη λειτουργία ενός φωτεινού σηματοδότη (φανάρι κυκλοφορίας). Αρχικά θα ανάβει το πράσινο φανάρι για 4 δευτερόλεπτα. Στη συνέχεια θα ανάβει το κίτρινο φανάρι για 1 δευτερόλεπτο και αμέσως μετά το κόκκινο για 3 δευτερόλεπτα. Η διαδικασία αυτή θα επαναλαμβάνεται συνεχώς.

### Υλικά που θα χρειαστούμε:

- ✓ 1 Arduino UNO
- ✓ 1 Breadboard
- ✓ 1 LED Κόκκινο (κόκκινο φανάρι)
- ✓ 1 LED Κίτρινο ή Πορτοκαλί (κίτρινο φανάρι)
- ✓ 1 LED Πράσινο (πράσινο φανάρι)
- ✓ 3 αντιστάσεις 220 Ohm



- 1 Ανοίξτε το προγραμματιστικό περιβάλλον του Arduino.
- 2 Από το μενού **Tools** επίλεξε **Ardublock** για να ξεκινήσετε το προγραμματιστικό περιβάλλον του Ardublock.
- 3 Δημιουργήστε το παρακάτω πρόγραμμα στο περιβάλλον του Ardublock.



\* Οι αριθμοί δίπλα σε κάθε εντολή χρησιμοποιούνται στην ανάλυση του προγράμματος που ακολουθεί στην επόμενη παράγραφο.

Εικόνα 2 - program2.adp

### Ανάλυση προγράμματος (program2.adp)

\* Κάθε γραμμή αντιστοιχεί σε μια εντολή (με τον ίδιο αριθμό) στην Εικόνα 2.

1. Η εντολή **set digital pin(13, LOW)** στέλνει μια τάση 0 V στην ψηφιακή θύρα 13. Αυτό έχει σαν αποτέλεσμα να σβήσει το πράσινο LED.
2. Η εντολή **set digital pin(12, LOW)** στέλνει μια τάση 0 V στην ψηφιακή θύρα 12. Αυτό έχει σαν αποτέλεσμα να σβήσει το πορτοκαλί LED.
3. Η εντολή **set digital pin(11, LOW)** στέλνει μια τάση 0 V στην ψηφιακή θύρα 11. Αυτό έχει σαν αποτέλεσμα να σβήσει το κόκκινο LED.
4. Η εντολή **set digital pin(13, HIGH)** στέλνει μια τάση 5 V στην ψηφιακή θύρα 13. Αυτό έχει σαν αποτέλεσμα να ανάψει το πράσινο LED.
5. Με την εντολή **delay MILLIS milliseconds(4000)** σταματά η εκτέλεση του προγράμματος για χρονικό διάστημα 4000 ms ( 1 sec = 1000 ms ).
6. Η εντολή **set digital pin(13, LOW)** στέλνει μια τάση 0 V στην ψηφιακή θύρα 13. Αυτό έχει σαν αποτέλεσμα να σβήσει το πράσινο LED.
7. Η εντολή **set digital pin(12, HIGH)** στέλνει μια τάση 5 V στην ψηφιακή θύρα 12. Αυτό έχει σαν αποτέλεσμα να ανάψει το πορτοκαλί LED.
8. Με την εντολή **delay MILLIS milliseconds(1000)** σταματά η εκτέλεση του προγράμματος για χρονικό διάστημα 1000 ms ( 1 sec = 1000 ms ).
9. Η εντολή **set digital pin(12, LOW)** στέλνει μια τάση 0 V στην ψηφιακή θύρα 12. Αυτό έχει σαν αποτέλεσμα να σβήσει το πορτοκαλί LED.
10. Η εντολή **set digital pin(11, HIGH)** στέλνει μια τάση 5 V στην ψηφιακή θύρα 11. Αυτό έχει σαν αποτέλεσμα να ανάψει το κόκκινο LED.
11. Με την εντολή **delay MILLIS milliseconds(3000)** σταματά η εκτέλεση του προγράμματος για χρονικό διάστημα 3000 ms ( 1 sec = 1000 ms ).
12. Η εντολή **set digital pin(11, LOW)** στέλνει μια τάση 0 V στην ψηφιακή θύρα 11. Αυτό έχει σαν αποτέλεσμα να σβήσει το κόκκινο LED.

Οι εντολές 1, 2, 3 βρίσκονται μέσα την **setup**. Η **setup** είναι μια συνάρτηση την οποία εκτελεί ο μικροελεγκτής μια φορά (όταν ο μικροελεγκτής τροφοδοτηθεί με ρεύμα την πρώτη φορά ή όταν πατηθεί το πλήκτρο reset). Αν έχουμε κλειστό τον μικροελεγκτή και τον ανοίξουμε εκτελείται η **setup** μια φορά και αμέσως μετά εκτελείται η **loop** συνεχώς.

Οι εντολές 4, 5, 6, 7, 8, 9, 10, 11, 12 βρίσκονται μέσα στην **loop**. Η **loop** είναι μια συνάρτηση την οποία εκτελεί ο μικροελεγκτής Arduino συνεχώς. Όταν ενεργοποιηθεί ο μικροελεγκτής Arduino, μετά την εκτέλεση των εντολών της **setup**, θα καλέσει την συνάρτηση **loop**. Μόλις ολοκληρωθεί η εκτέλεση των εντολών της **loop**, ο μικροελεγκτής θα ξανακαλέσει την συνάρτηση **loop**. Αυτό θα γίνεται συνεχώς, όσο ο μικροελεγκτής έχει ρεύμα. Αν πατήσουμε το πλήκτρο **reset** του μικροελεγκτή τότε θα τρέξει η συνάρτηση **setup** μια φορά και μετά θα εκτελείται η **loop** συνεχώς.

## Δραστηριότητες

**1.** Προσπαθήστε να προσθέσετε άλλο δύο LED στο παραπάνω κύκλωμα, ένα κόκκινο και ένα πράσινο, τα οποία θα προσομοιώνουν τα φανάρια των πεζών. Δηλαδή, όταν το πράσινο φανάρι είναι αναμμένο (πράσινο LED στη θύρα 13) τότε το κόκκινο LED που θα προσθέσετε θα είναι αναμμένο. Επίσης, όταν το κόκκινο φανάρι είναι αναμμένο (κόκκινο LED στη θύρα 11) τότε το πράσινο LED που θα προσθέσετε θα είναι αναμμένο.

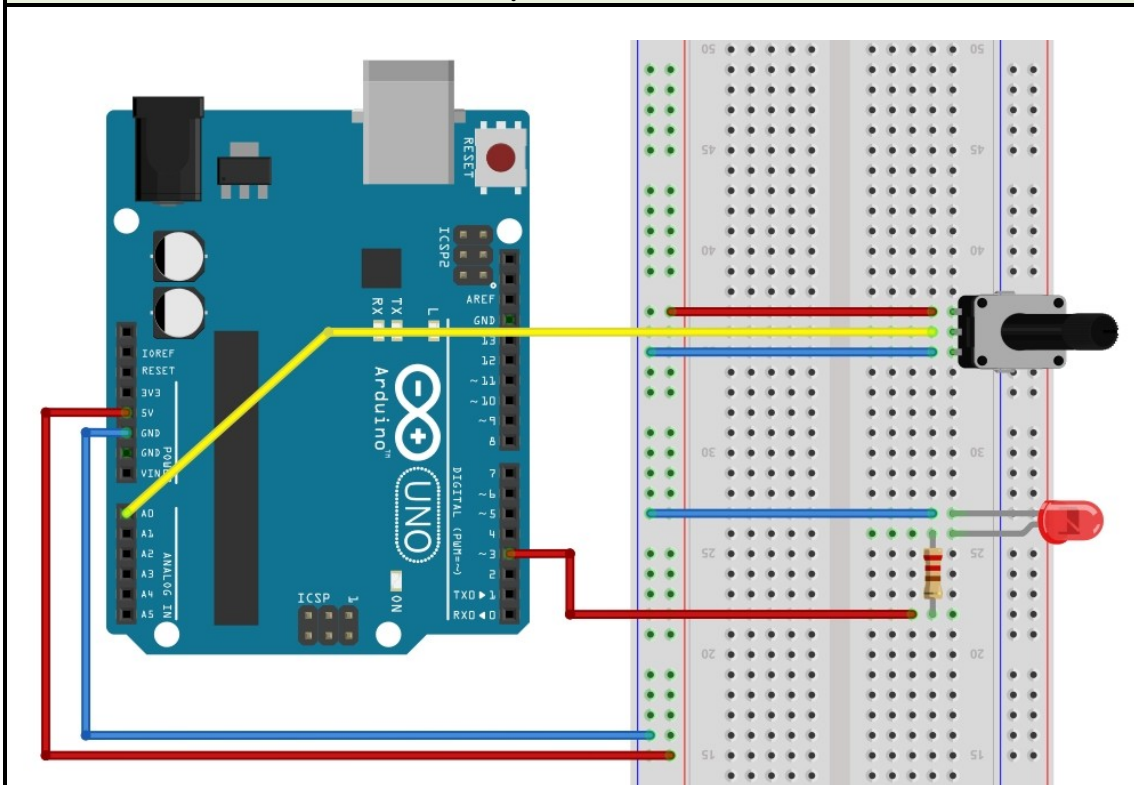
### Φύλλο εργασίας 3 - Ρύθμιση της φωτεινότητας ενός LED με ποτενσιόμετρο

Στη δραστηριότητα αυτή θα κατασκευάσουμε ένα κύκλωμα με το οποίο θα ρυθμίζουμε τη φωτεινότητα ενός LED με τη χρήση ενός ποτενσιόμετρου. Το ποτενσιόμετρο είναι ένα αναλογικό ηλεκτρονικό εξάρτημα που μας παρέχει μια μεταβλητή τάση εξόδου. Το βασικό του στοιχείο είναι μια μεταβλητή αντίσταση πάνω στην οποία κινείται μια επαφή η οποία συνδέεται με το μεσαίο ακροδέκτη (pin) του. Η θέση της επαφής στη μεταβλητή αντίσταση καθορίζει την τάση στο μεσαίο ακροδέκτη.

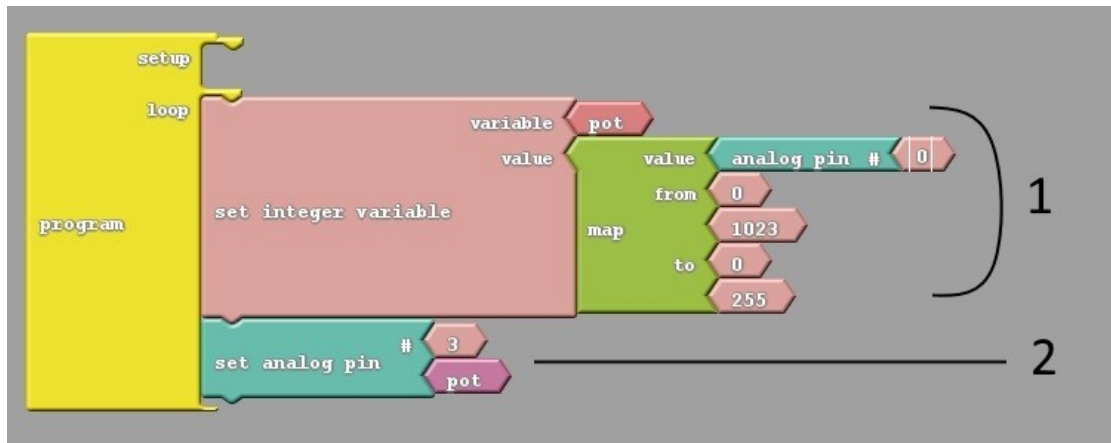
#### Υλικά που θα χρειαστούμε:

- ✓ 1 Arduino UNO
- ✓ 1 Breadboard
- ✓ 1 περιστροφικό ποτενσιόμετρο
- ✓ 1 LED
- ✓ 1 αντίσταση 220 Ohm

#### Το κύκλωμα σε breadboard



- 1 Ανοίξτε το προγραμματιστικό περιβάλλον του Arduino.
- 2 Από το μενού **Tools** επίλεξε **Ardublock** για να ξεκινήσετε το προγραμματιστικό περιβάλλον του Ardublock.
- 3 Δημιουργήστε το παρακάτω πρόγραμμα στο περιβάλλον του Ardublock.



Εικόνα 3 – program3.adp

\* Οι αριθμοί δίπλα σε κάθε εντολή χρησιμοποιούνται στην ανάλυση του προγράμματος που ακολουθεί στην επόμενη παράγραφο.

### Ανάλυση προγράμματος (program3.adp)

\* Κάθε γραμμή αντιστοιχεί σε μια εντολή (με τον ίδιο αριθμό) στην Εικόνα 3.

1. Η εντολή **set integer variable** θέτει στην μεταβλητή **pot** μια τιμή (value). Στη συγκεκριμένη περίπτωση η τιμή είναι το αποτέλεσμα της συνάρτησης **map**. Η συνάρτηση **map** παίρνει την τιμή που επιστρέφει το **ποτενσιόμετρο** (με την εντολή **analog pin # 0**), και η οποία ανήκει στο διάστημα [0, 1023], και την αντιστοιχεί σε έναν αριθμό στο διάστημα [0, 255]. Η μετατροπή αυτή είναι απαραίτητη γιατί οι τιμές που επιτρέπεται να στείλουμε στον ακροδέκτη (pin) 3, στον οποίο είναι συνδεδεμένο το LED, πρέπει να ανήκουν στο διάστημα [0, 255].
2. Η εντολή **set analog pin(3, pot)** στέλνει την τιμή της μεταβλητής **pot** στην ψηφιακή θύρα 3. Η τιμή της μεταβλητής **pot** είναι μια τιμή από 0 έως 255. Οπότε, ανάλογα με την τιμή της **pot** θα έχουμε και το αντίστοιχο επίπεδο φωτεινότητας στο LED.

*Η set analog pin λειτουργεί μόνο στις PWM ψηφιακές θύρες οι οποίες στις περισσότερες πλακέτες είναι οι 3, 5, 6, 9, 10, 11. Δηλαδή στις θύρες αυτές*

μπορούμε να παράγουμε ενδιάμεσες τάσεις στο διάστημα  $[0, 5]$  Volt στέλνοντας τιμές από 0 έως 255.



## Δραστηριότητες

1. Προσθέστε άλλο ένα LED στο παραπάνω κύκλωμα και ρυθμίστε τη φωτεινότητα του με το ποτενσιόμετρο. Δηλαδή, με το ποτενσιόμετρο να ρυθμίζετε τη φωτεινότητα και των δύο LED.
2. Τροποποιήστε το πρόγραμμα που αναπτύξατε στην προηγούμενη άσκηση ώστε η φωτεινότητα των δύο LED να ρυθμίζεται με το ποτενσιόμετρο ως εξής: όταν η φωτεινότητα του ενός LED αυξάνει, η φωτεινότητα του άλλου να μειώνεται.

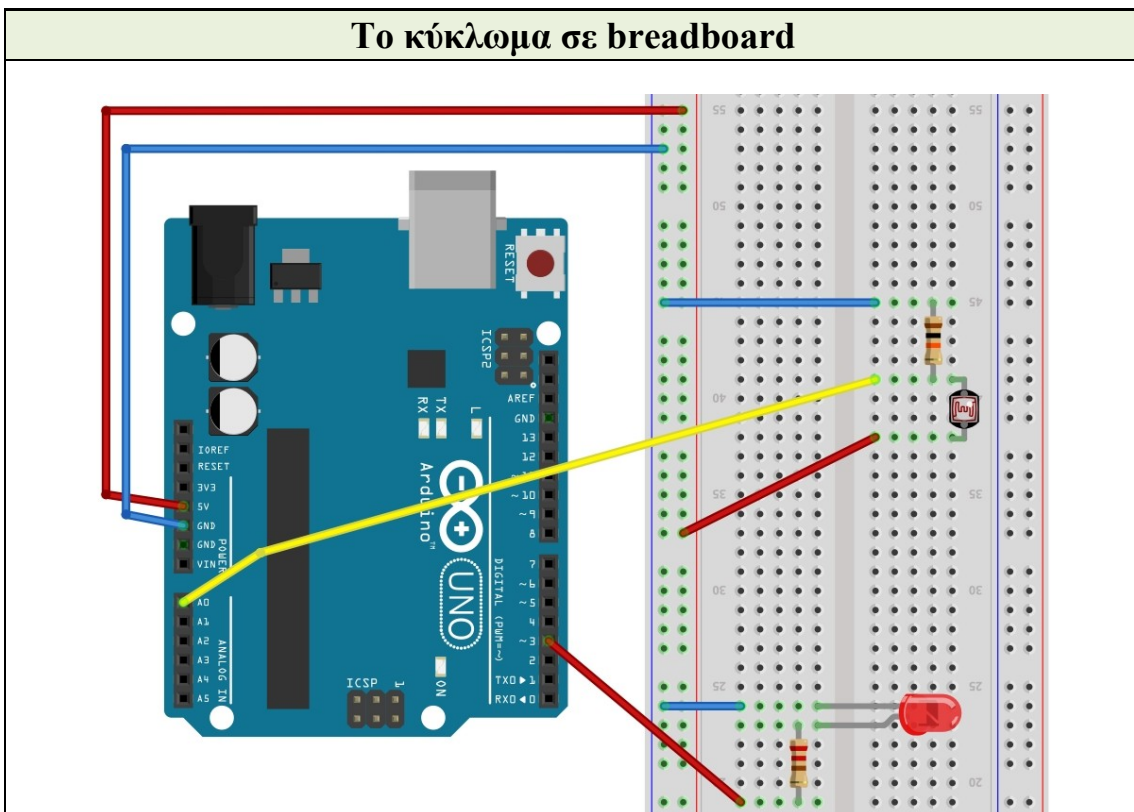
## Φύλλο εργασίας 4 - Αυτόματο φωτάκι νυκτός

Σε αυτήν την δραστηριότητα θα κατασκευάσουμε ένα αυτόματο φωτάκι νυκτός. Η διάταξη που θα δημιουργήσουμε θα αποτελείται από ένα LED και μια φωτοευαίσθητη αντίσταση. Η φωτοευαίσθητη αντίσταση είναι μια μεταβλητή αντίσταση η τιμή της οποίας εξαρτάται από την ένταση του φωτός που προσπίπτει σε αυτή. Όσο αυξάνει η ένταση του προσπίπτοντος φωτός η τιμή της φωτοευαίσθητης αντίστασης μειώνεται. Στο αυτόματο φωτάκι νυκτός που θα κατασκευάσουμε, το φωτάκι LED θα ανάβει όταν η τιμή της φωτοευαίσθητης αντίστασης πέσει κάτω από κάποια τιμή (κατώφλι) την οποία θα βρούμε κάνοντας δοκιμαστικές μετρήσεις.

### Υλικά που θα χρειαστούμε:

- ✓ 1 Arduino UNO
- ✓ 1 Breadboard
- ✓ 1 φωτοευαίσθητη αντίσταση
- ✓ 1 αντίσταση 10 kΩ 
- ✓ 1 LED
- ✓ 1 αντίσταση 220 Ω 

### Το κύκλωμα σε breadboard

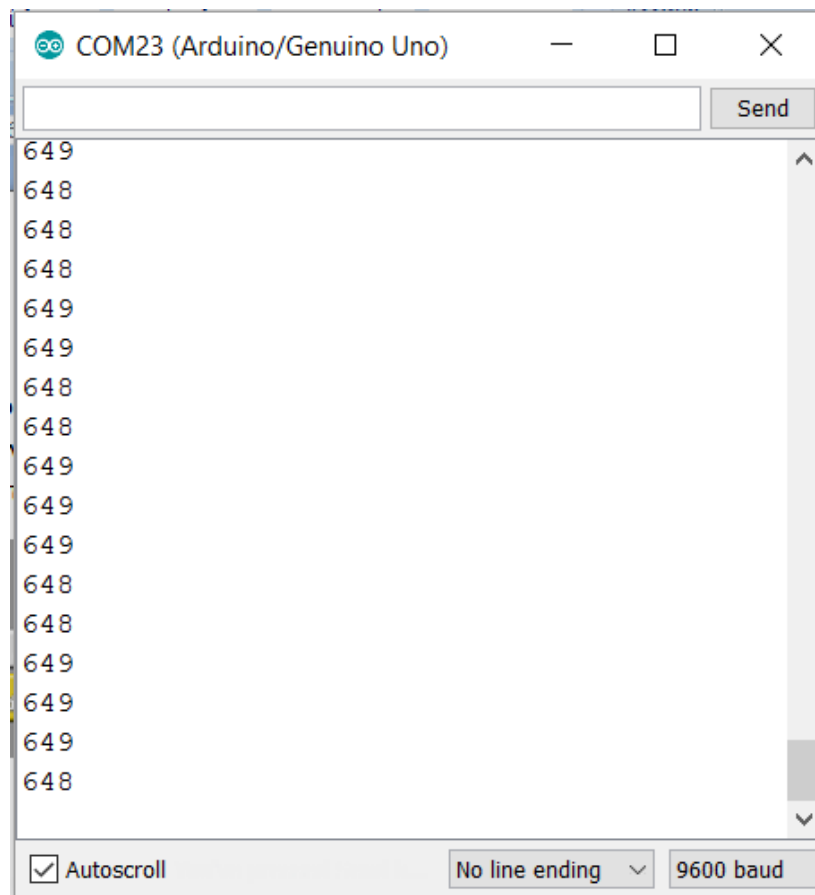




Πριν προχωρήσουμε με την ανάπτυξη του προγράμματος τρέχουμε το παρακάτω πρόγραμμα το οποίο εμφανίζει στην σειριακή οθόνη (Serial Monitor) τις τιμές που επιστρέφει η φωτοευαίσθητη αντίσταση.



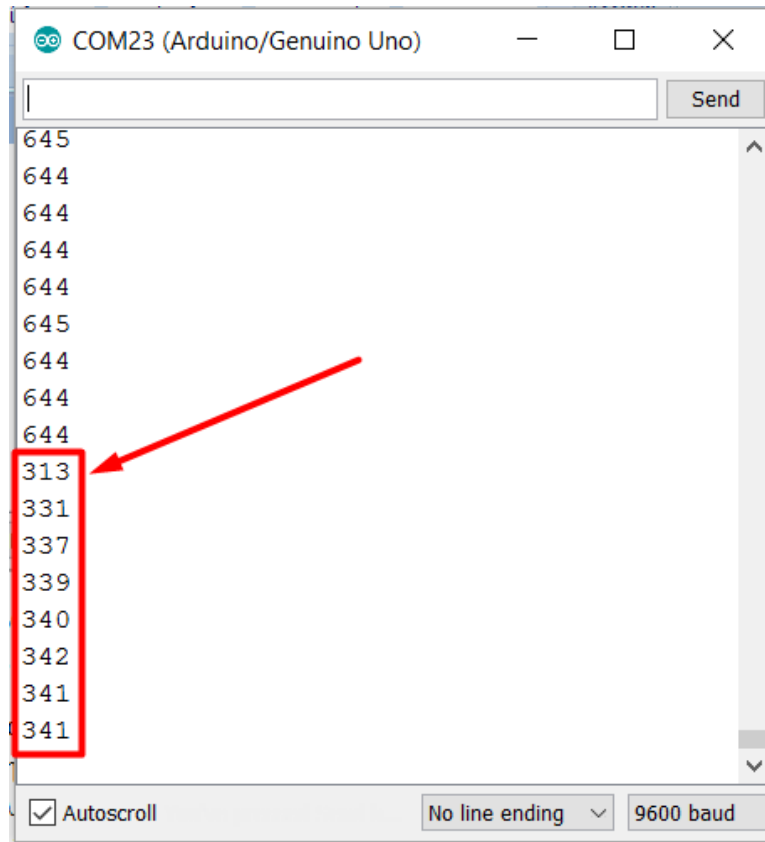
Η σειριακή οθόνη ενεργοποιείται κάνοντας κλικ στο κουμπί "**Serial Monitor**" του Ardublock.



Αν παρατηρήσουμε την σειριακή οθόνη θα διαπιστώσουμε ότι η φωτοευαίσθητη αντίσταση επιστρέφει τις τιμές 648 ή 649.

Αν ακουμπήσουμε το δάκτυλό μας πάνω στην φωτοευαίσθητη αντίσταση ώστε να προσομοιώσουμε την κατάσταση της νύκτας, όπου η ένταση του φωτός που προσπίπτει σε αυτή είναι πάρα πολύ μικρή, τότε θα δούμε ότι στην σειριακή οθόνη η τιμή που επιστρέφει η φωτοευαίσθητη αντίσταση είναι περίπου στην μέση της αρχικής τιμής.

Στο κόκκινο πλαίσιο βλέπουμε τις τιμές που επιστρέφει η φωτοευαίσθητη αντίσταση όταν ακουμπήσουμε το δάκτυλό μας πάνω της.

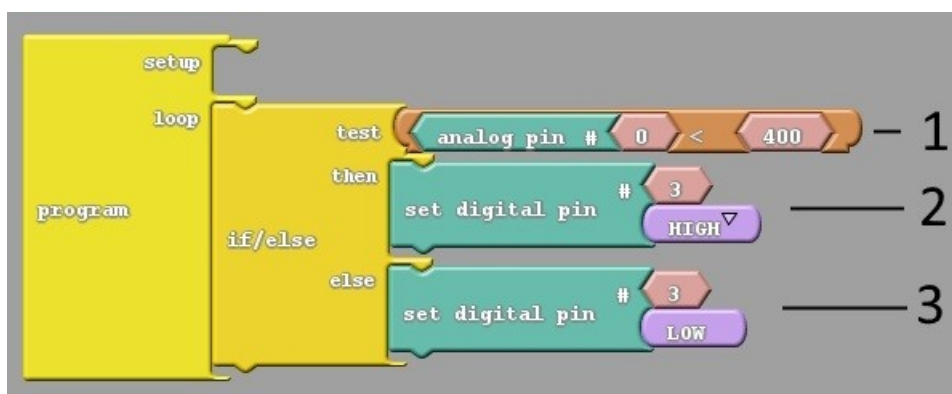


Από τις προηγούμενες δοκιμαστικές μετρήσεις επιλέγουμε την τιμή (κατώφλι) κάτω από την οποία πρέπει να ανάψει το LED. Επιλέγουμε ως τιμή κατωφλίου την τιμή 400. Οπότε, όταν η φωτοευαίσθητη αντίσταση επιστρέφει μια τιμή μικρότερη του 400 θα ανάψει το LED, σε διαφορετική περίπτωση το LED θα είναι σβηστό.

*Οι παραπάνω τιμές εξαρτώνται από τις συνθήκες φωτισμού του χώρου σας και από το κύκλωμα που χρησιμοποιείτε κάθε φορά.*

Ακολουθήστε τα παρακάτω βήματα για ανάπτυξη του προγράμματος που θα προσομοιώνει τη λειτουργία του αυτόματου φωτιστικού νυκτός:

- 1 Ανοίξτε το προγραμματιστικό περιβάλλον του Arduino.
- 2 Από το μενού **Tools** επίλεξε **Ardublock** για να ξεκινήσετε το προγραμματιστικό περιβάλλον του Ardublock.
- 3 Δημιουργήστε το παρακάτω πρόγραμμα στο περιβάλλον του Ardublock.



Εικόνα 4 – program4.adp

\* Οι αριθμοί δίπλα σε κάθε εντολή χρησιμοποιούνται στην ανάλυση του προγράμματος που ακολουθεί στην επόμενη παράγραφο.

### Ανάλυση προγράμματος (program4.adp)

\* Κάθε γραμμή αντιστοιχεί σε μια εντολή (με τον ίδιο αριθμό) στην Εικόνα 4.

1. Η εντολή **if ... then ... else ...** ελέγχει αν η τιμή που επιστρέφει η φωτοευαίσθητη αντίσταση είναι μικρότερη του 400. Αν είναι τότε εκτελείται η εντολή στη γραμμή 2, αλλιώς εκτελείται η εντολή η γραμμή 3.
2. Η εντολή **set digital pin( 3, HIGH )** στέλνει μια τάση 5 V στην ψηφιακή θύρα 3. Αυτό έχει σαν αποτέλεσμα να ανάψει το LED που χρησιμοποιήσαμε.
3. Η εντολή **set digital pin( 3, LOW )** στέλνει μια τάση 0 V στην ψηφιακή θύρα 3. Αυτό έχει σαν αποτέλεσμα να σβήσει το LED που χρησιμοποιήσαμε.

## Δραστηριότητες

1. Προσθέστε άλλο ένα LED στο παραπάνω κύκλωμα και αλλάξτε τη λειτουργία του αυτόματου φωτιστικού νυκτός ως εξής: αν η τιμή που επιστρέφει η φωτοευαίσθητη αντίσταση είναι μικρότερη του 400 ( ή της τιμής που επιλέξατε ως κατώφλι) τότε θα ανάβει το πρώτο LED, αλλιώς θα ανάβει το δεύτερο LED.

2. Προσθέστε άλλο ένα LED στο παραπάνω κύκλωμα (ώστε συνολικά να έχουμε 3 LED) και ακολουθήστε τα εξής βήματα:

- κάντε μερικές δοκιμαστικές μετρήσεις και βρείτε το διάστημα των τιμών που επιστρέφει η φωτοευαίσθητη αντίσταση, π.χ. [300, 900]
- χωρίστε το διάστημα τιμών που βρήκατε σε τρία ίσα διαστήματα, π.χ. [300, 500], (500, 700], (700, 900]
- τροποποιήστε τον αρχικό κώδικα στο Ardublock ώστε να λειτουργεί ως εξής:
  - αν η τιμή που επιστρέφει η φωτοευαίσθητη αντίσταση ανήκει στο διάστημα (700, 900] τότε να ανάβει ένα LED,
  - αλλιώς αν η τιμή που επιστρέφει η φωτοευαίσθητη αντίσταση ανήκει στο διάστημα (500, 700] τότε να ανάβουν δύο LED,
  - αλλιώς αν η τιμή που επιστρέφει η φωτοευαίσθητη αντίσταση ανήκει στο διάστημα [300, 500] τότε να ανάβουν τρία LED.

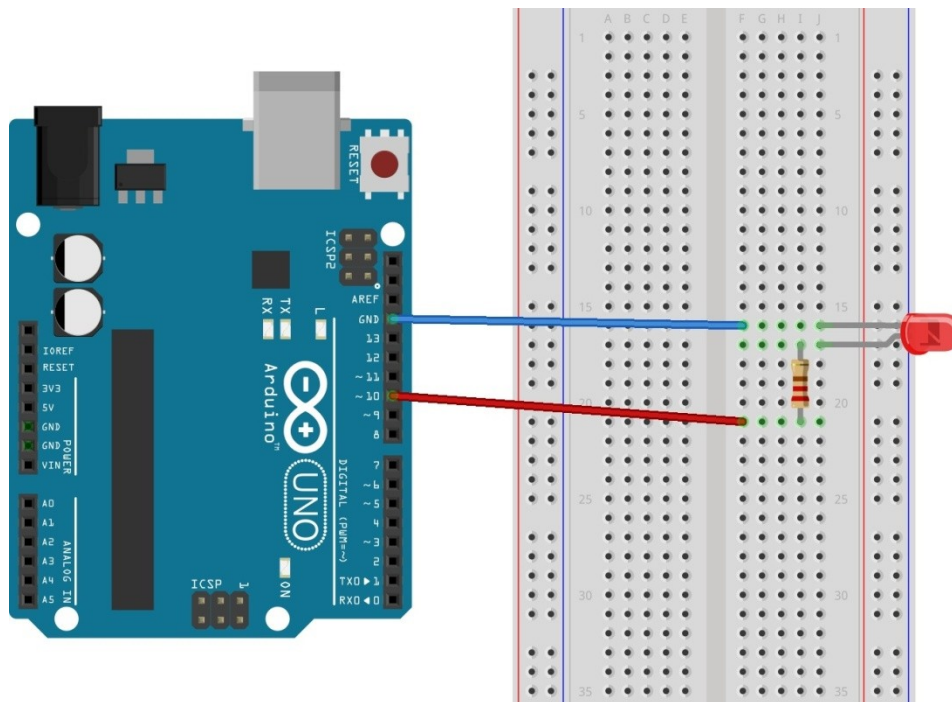
## Φύλλο εργασίας 5 - Χριστουγεννιάτικα φωτάκια (σταδιακή αύξηση και μείωση φωτεινότητας ενός LED)

Στην δραστηριότητα αυτή θα χρησιμοποιήσουμε ένα LED το οποίο θα ανάβει σταδιακά και όταν θα φτάσει στη μέγιστη τιμή του θα αρχίσει να σβήνει σταδιακά. Η διαδικασία αυτή θα επαναλαμβάνεται συνεχώς.

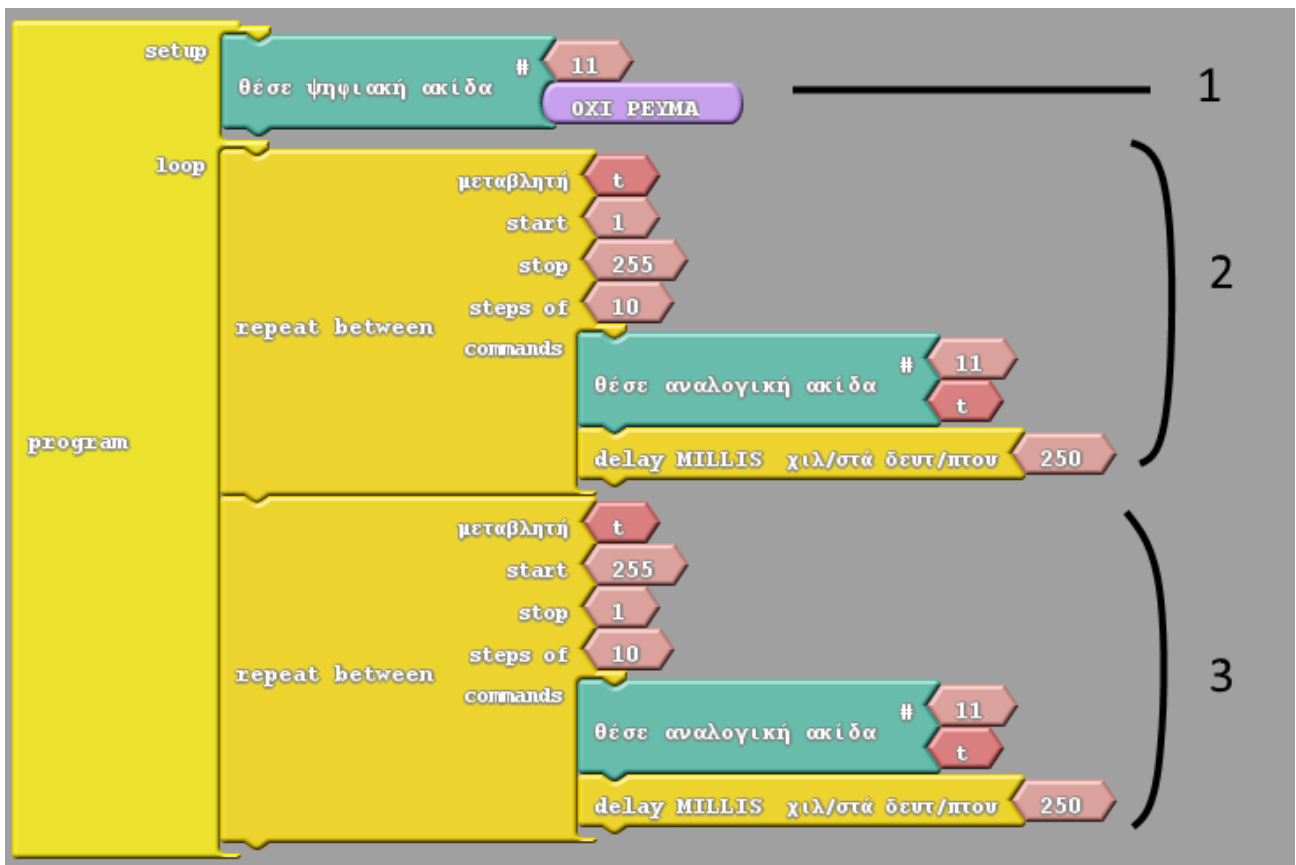
### Υλικά που θα χρειαστούμε:

- ✓ 1 Arduino UNO
- ✓ 1 Breadboard
- ✓ 1 LED Κόκκινο
- ✓ 1 αντίσταση 220 Ohm

### Το κύκλωμα σε breadboard



- 1 Ανοίξτε το προγραμματιστικό περιβάλλον του Arduino.
- 2 Από το μενού **Tools** επίλεξε **Ardublock** για να ξεκινήσετε το προγραμματιστικό περιβάλλον του Ardublock.
- 3 Δημιουργήστε το παρακάτω πρόγραμμα στο περιβάλλον του Ardublock.



Εικόνα 5 – program5.adp

\* Οι αριθμοί δίπλα σε κάθε εντολή χρησιμοποιούνται στην ανάλυση του προγράμματος που ακολουθεί στην επόμενη παράγραφο.

## Ανάλυση προγράμματος (program5.adp)

\* Κάθε γραμμή αντιστοιχεί σε μια εντολή (με τον ίδιο αριθμό) στην Εικόνα 5.

1. Η εντολή **θέσε ψηφιακή ακίδα(11, ΟΧΙ ΡΕΥΜΑ)** στέλνει μια τάση 0 V στην ψηφιακή θύρα 11. Αυτό έχει σαν αποτέλεσμα να σβήσει το κόκκινο LED. Η εντολή αυτή βρίσκεται μέσα την **setup**. Η **setup** είναι μια συνάρτηση την οποία εκτελεί ο μικροελεγκτής μια φορά (όταν ο μικροελεγκτής τροφοδοτηθεί με ρεύμα την πρώτη φορά ή όταν πατηθεί το πλήκτρο reset).
2. Η εντολή **repeat between (μεταβλητή: t, start: 1, stop: 255, steps of: 10)** εκτελεί τις εντολές που βρίσκονται στην περιοχή commands πολλές φορές.

Η εντολή repeat λειτουργεί ως εξής:

**Βήμα 1:** Αρχικά στην μεταβλητή t καταχωρείται η τιμή 1 (*παράμετρος start*).

**Βήμα 2:** Στη συνέχεια εξετάζεται η τιμή της συνθήκης  $t \leq 255$  (*παράμετρος stop*). Αν η συνθήκης είναι αληθής τότε εκτελούνται οι εντολές στην περιοχή commands, αλλιώς αν η συνθήκης είναι ψευδής τότε **δεν** εκτελείται καμία εντολή και η repeat **τερματίζει** την εκτέλεσή της.

**Βήμα 3:** Αυξάνεται η τιμή της μεταβλητής t κατά 10 (*παράμετρος steps of*).

Τα βήμα τα 2 και 3 επαναλαμβάνονται συνεχώς μέχρι η συνθήκη  $t \leq 255$  να γίνει ψευδής. Δηλαδή, όσο  $t \leq 255$  θα εκτελούνται οι εντολές στην περιοχή commands.

Η εντολή **θέσε αναλογική ακίδα (11, t)** στέλνει μια τάση που αντιστοιχεί στην τιμή της μεταβλητής t στην ψηφιακή θύρα 11.

Η εντολή **delay MILLIS χιλ/στα δευ/πτου (250)** σταματά την εκτέλεση του προγράμματος για χρονικό διάστημα 250 ms ( 1 sec = 1000 ms ).

Η εντολή repeat στέλνει τις εξής τάσεις στην ψηφιακή θύρα 11:  
1, 11, 21, 31, 41, ..., 241, 251 με αποτέλεσμα (λόγω της εντολής delay που ακολουθεί) το LED να αυξάνει τη φωτεινότητα που εκπέμπει σταδιακά.

*Η εντολή **θέσε αναλογική ακίδα** λειτουργεί μόνο στις **PWM** ψηφιακές θύρες οι οποίες στις περισσότερες πλακέτες είναι οι 3, 5, 6, 9, 10, 11. Δηλαδή στις θύρες αυτές μπορούμε να παράγουμε ενδιάμεσες τάσεις στο διάστημα [0, 5] Volt στέλνοντας τιμές από 0 έως 255.*

- 3.** Η εντολή **repeat between** (μεταβλητή: **t**, **start**: 255, **stop**: 1, **steps of**: 10) εκτελεί τις εντολές που βρίσκονται στην περιοχή **commands** πολλές φορές.

Η εντολή **repeat** λειτουργεί ως εξής:

**Βήμα 1:** Αρχικά στην μεταβλητή **t** καταχωρείται η τιμή 255 (παράμετρος **start**).

**Βήμα 2:** Στη συνέχεια εξετάζεται η τιμή της συνθήκης  $t \geq 1$  (παράμετρος **stop**). Αν η συνθήκη είναι αληθής τότε εκτελούνται οι εντολές στην περιοχή **commands**, αλλιώς αν η συνθήκη είναι ψευδής τότε **δεν** εκτελείται καμία εντολή και η **repeat** **τερματίζει** την εκτέλεσή της .

→ Παρατηρήστε ότι επειδή  $start > stop$  έχει αλλάξει η συνθήκη, αντί  $t \leq 1$  έχει γίνει  $t \geq 1$ ).

**Βήμα 3:** Μειώνεται η τιμή της μεταβλητής **t** κατά 10 (παράμετρος **steps of**).

→ Επειδή  $start > stop$  η μεταβλητή μειώνεται αντί να αυξάνεται.

Τα βήμα τα 2 και 3 επαναλαμβάνονται συνεχώς μέχρι η συνθήκη  $t \geq 1$  να γίνει ψευδής. Δηλαδή, όσο  $t \geq 1$  θα εκτελούνται οι εντολές στην περιοχή **commands**.

Η εντολή **θέσε αναλογική ακίδα (11, t)** στέλνει μια τάση που αντιστοιχεί στην τιμή της μεταβλητής **t** στην ψηφιακή θύρα 11.

Η εντολή **delay MILLIS χιλ/στα δευ/πτου (250)** σταματά την εκτέλεση του προγράμματος για χρονικό διάστημα 250 ms ( 1 sec = 1000 ms ).

Η εντολή **repeat** στέλνει τις εξής τάσεις στην ψηφιακή θύρα 11: 255, 245, 235, ..., 25, 15, 5 με αποτέλεσμα (λόγω της εντολής **delay** που ακολουθεί) το LED να μειώνει τη φωτεινότητα που εκπέμπει σταδιακά.

## Δραστηριότητες

- 1.** Προσπαθήστε να προσθέσετε ένα πράσινο LED στο παραπάνω κύκλωμα το οποίο θα έχει αντίστροφη λειτουργία με το υπάρχον κόκκινο LED. Συγκεκριμένα, όταν η φωτεινότητα του κόκκινου LED θα αυξάνει, η φωτεινότητα του πράσινου LED θα μειώνεται και το αντίστροφο.



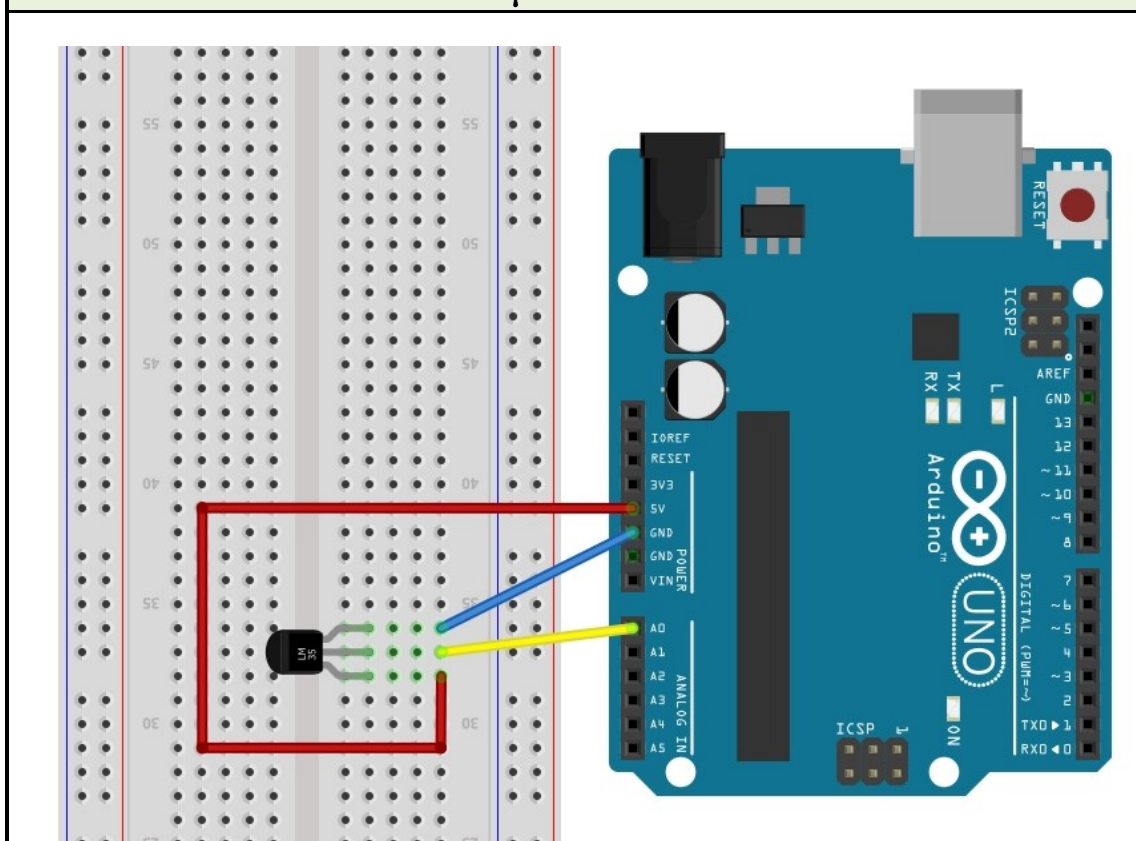
## Φύλλο εργασίας 6 - Θερμόμετρο εξωτερικού χώρου

Σε αυτήν την δραστηριότητα θα κατασκευάσουμε ένα θερμόμετρο εξωτερικού χώρου. Θα χρησιμοποιήσουμε τον αισθητήρα θερμοκρασίας LM35 και για την ένδειξη της θερμοκρασίας θα αξιοποιήσουμε τη σειριακή οθόνη (Serial Monitor). Όσο ο μικροελεγκτής είναι συνδεδεμένος με USB με τον υπολογιστή, μπορεί να στείλει δεδομένα στην σειριακή οθόνη (Serial Monitor). Η σειριακή οθόνη ενεργοποιείται κάνοντας κλικ στο κουμπί "Serial Monitor" του Ardublock.

### Υλικά που θα χρειαστούμε:

- ✓ 1 Arduino UNO
- ✓ 1 Breadboard
- ✓ 1 αισθητήρα θερμοκρασίας LM35

### Το κύκλωμα σε breadboard



- 1 Ανοίξτε το προγραμματιστικό περιβάλλον του Arduino.
- 2 Από το μενού **Tools** επίλεξε **Ardublock** για να ξεκινήσετε το προγραμματιστικό περιβάλλον του Ardublock.
- 3 Δημιουργήστε το παρακάτω πρόγραμμα στο περιβάλλον του Ardublock.



Εικόνα 6 - program6.adp

\* Οι αριθμοί δίπλα σε κάθε εντολή χρησιμοποιούνται στην ανάλυση του προγράμματος που ακολουθεί στην επόμενη παράγραφο.

### Ανάλυση προγράμματος (program6.adp)

\* Κάθε γραμμή αντιστοιχεί σε μια εντολή (με τον ίδιο αριθμό) στην Εικόνα 6.

1. Η εντολή **set decimal number variable** ( vin, **analog pin # 0** x 5.0 ) θέτει στην μεταβλητή **vin** το αποτέλεσμα του πολλαπλασιασμού της τιμής που επιστρέφει ο αισθητήρας θερμοκρασίας (**analog pin # 0**) με τον αριθμό 5.

*Η εντολή **analog pin #0** επιστρέφει την τιμή του ακροδέκτη εισόδου A0. Στην συγκεκριμένη εφαρμογή στον ακροδέκτη αυτό (A0) είναι συνδεδεμένη η έξοδος του αισθητήρα θερμοκρασίας LM35.*

2. Η εντολή **set decimal number variable** ( vin,  $vin / 1024$  ) θέτει στην μεταβλητή **vin** το αποτέλεσμα της διαίρεσης  $vin / 1024$ .
3. Η εντολή **set decimal number variable** ( vin,  $vin \times 1000.0$  ) θέτει στην μεταβλητή **vin** το αποτέλεσμα του πολλαπλασιασμού  $vin \times 1000$ .

- 4.** Η εντολή **set decimal number variable** ( vin, vin / 10 ) θέτει στην μεταβλητή **vin** το αποτέλεσμα της διαίρεσης vin / 10.
- 5.** Η εντολή **serial println( glue ( vin ) )** στέλνει στην σειριακή οθόνη (Serial Monitor) την τιμή της μεταβλητής vin.
- 6.** Με την εντολή **delay MILLIS milliseconds(1000)** σταματά η εκτέλεση του προγράμματος για χρονικό διάστημα 1000 ms ( 1 sec = 1000 ms ).

*Όλες οι παραπάνω αριθμητικές πράξεις έχουν ως σκοπό την μετατροπή της τιμής που επιστρέφει ο αισθητήρας θερμοκρασίας στην κλίμακα των βαθμών Κελσίου. Οι αναλογικές τιμές που επιστρέφει ο αισθητήρας θερμοκρασίας αντιστοιχίζονται σε ψηφιακές διακριτές τιμές στον μικροελεγκτή Arduino. Επειδή ο μικροελεγκτής διαθέτει ακρίβεια 10 bits για βρούμε το βήμα μεταβολής των διακριτών τιμών πρέπει να κάνουμε την πράξη  $5/2^{10} = 5/1024 = 0.0048828125$ . Ο αριθμός 5 είναι η τάση 5V στην οποία αντιστοιχίζεται η μέγιστη τιμή που επιστρέφει ο αισθητήρας θερμοκρασίας.*

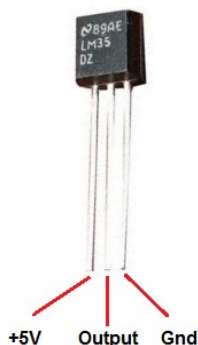
*Επομένως η τάση εισόδου δίνεται από την σχέση:  $V = (\text{analog pin \#0}) * 5/1024$   
Η εντολή **analog pin #0** επιστρέφει την τιμή του ακροδέκτη εισόδου A0.*

*Για τον αισθητήρα LM35, ισχύει ότι για κάθε έναν  $^{\circ}\text{C}$  αντιστοιχούν 10 mV. Συνεπώς για να μετατρέψουμε την παραπάνω τάση εισόδου σε βαθμούς Κελσίου πρέπει να την μετατρέψουμε σε mV (πολλαπλασιάζοντας με το 1000) και στη συνέχεια να διαιρέσουμε με τον αριθμό 10 mV.*

*Οπότε η σχέση που μετατρέπει την τιμή που επιστρέφει ο αισθητήρας θερμοκρασίας LM35 σε βαθμού Κελσίου είναι η εξής:*

$$C = (\text{analog pin \#0}) * \frac{5}{1024} * \frac{1000}{10}$$

### Αισθητήρας θερμοκρασίας LM35





Στην συγκεκριμένη εφαρμογή χρησιμοποιήσαμε την εντολή **set decimal number variable** αντί για την εντολή **set integer variable** που έχουμε μάθει μέχρι τώρα.

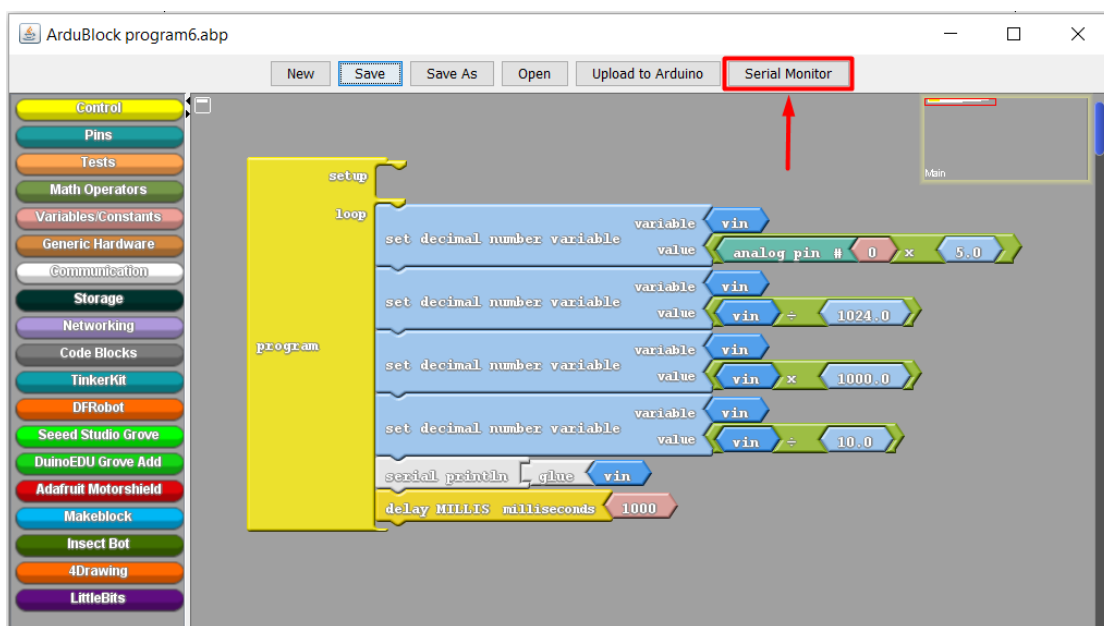
Με την εντολή **set integer variable** μπορούμε να διαχειριστούμε μόνο ακέραιους αριθμούς.

Με την εντολή **set decimal number variable** μπορούμε να διαχειριστούμε πραγματικούς αριθμούς κινητής υποδιαστολής.

Στην συγκεκριμένη εφαρμογή χρησιμοποιήσαμε την εντολή **set decimal number variable** διότι οι τιμές της θερμοκρασίας είναι πραγματικοί αριθμοί.



Η σειριακή οθόνη ενεργοποιείται κάνοντας κλικ στο κουμπί "**Serial Monitor**" του Ardublock.

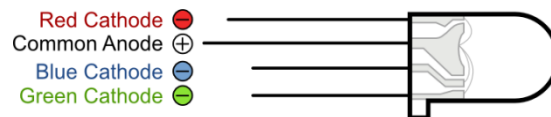


## Δραστηριότητες

1. Προσθέστε 3 LED στο παραπάνω κύκλωμα τα οποία θα ανάβουν ως εξής: όταν η θερμοκρασία που επιστρέφει ο αισθητήρας LM35 ανήκει στο διάστημα  $[0, 15]$  βαθμοί Κελσίου θα ανάβει το μόνο το πρώτο LED. Όταν η θερμοκρασία είναι στο διάστημα  $(15, 30]$  θα ανάβει το πρώτο και το δεύτερο LED και όταν είναι μεγαλύτερη των 30 βαθμών Κελσίου θα ανάβουν όλα τα LED.

## Φύλλο εργασίας 7 - Δημιουργώ τα δικά μου χρώματα με το RGB LED

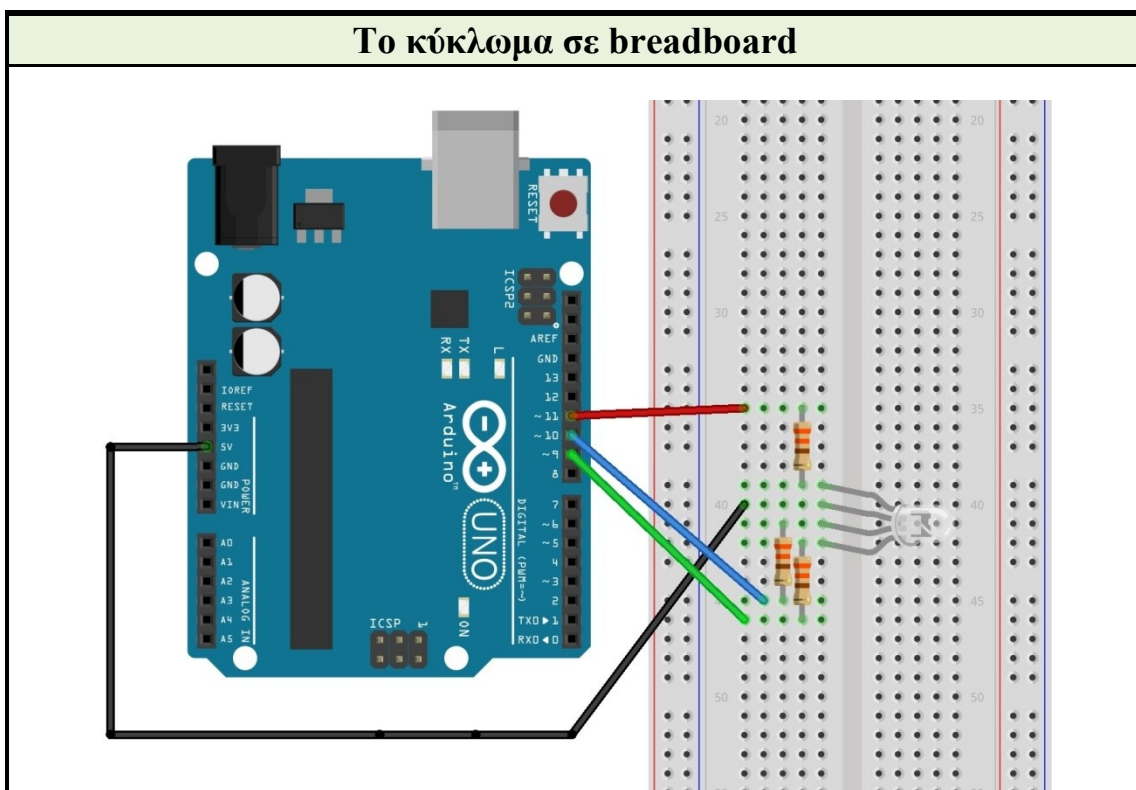
Στην δραστηριότητα αυτή θα δουλέψουμε με το RGB LED για να παράγουμε μια μεγάλη ποικιλία χρωμάτων. Το RGB LED είναι στην ουσία τρία διαφορετικά LED (κόκκινο, πράσινο, μπλε) τα οποία έχουν ενσωματωθεί σε ένα LED. Ένα RGB LED (Red, Green, Blue) διαθέτει 4 ακροδέκτες, έναν για κάθε χρώμα και έναν κοινής ανόδου (ή καθόδου αν έχουμε LED κοινής καθόδου). Τα RGB LED παράγουν κόκκινο, πράσινο, μπλε χρώμα και συνδυασμούς αυτών των χρωμάτων. Το RGB LED που θα χρησιμοποιήσουμε έχει τους παρακάτω ακροδέκτες (pins):



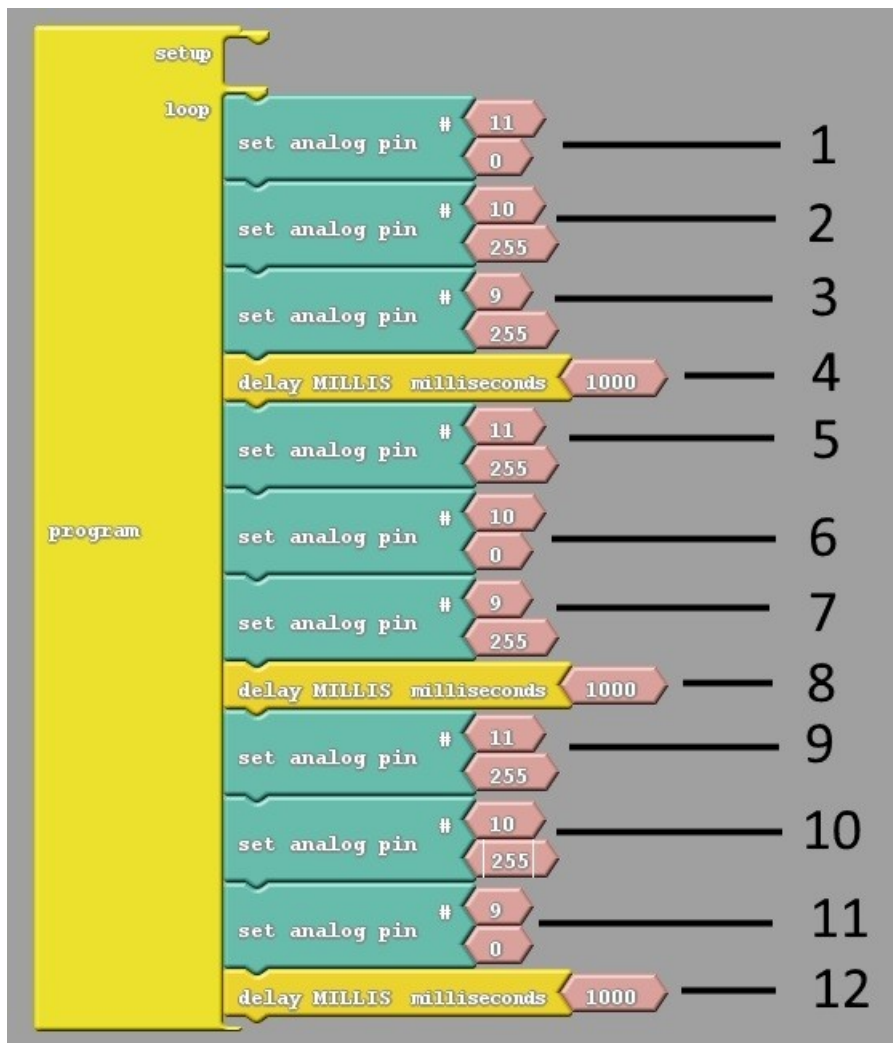
### Υλικά που θα χρειαστούμε:

- ✓ 1 Arduino UNO
- ✓ 1 Breadboard
- ✓ 1 RGB LED (LED Diffused 5mm RGB - Common Anode)
- ✓ 3 αντιστάσεις 330 Ohm

### Το κύκλωμα σε breadboard



- 1 Ανοίξτε το προγραμματιστικό περιβάλλον του Arduino.
- 2 Από το μενού **Tools** επίλεξε **Ardublock** για να ξεκινήσετε το προγραμματιστικό περιβάλλον του Ardublock.
- 3 Δημιουργήστε το παρακάτω πρόγραμμα στο περιβάλλον του Ardublock.



Εικόνα 7 – program7.adp

\* Οι αριθμοί δίπλα σε κάθε εντολή χρησιμοποιούνται στην ανάλυση του προγράμματος που ακολουθεί στην επόμενη παράγραφο.

### Ανάλυση προγράμματος (program7.adp)

\* Κάθε γραμμή αντιστοιχεί σε μια εντολή (με τον ίδιο αριθμό) στην Εικόνα 7.

1. Η εντολή **set analog pin(11, 0)** στέλνει την τάση 0 την ψηφιακή θύρα 11. Αυτό έχει σαν αποτέλεσμα το κόκκινο χρώμα να πάρει την μέγιστη τιμή του. **Επειδή το RGB LED που χρησιμοποιούμε είναι κοινής ανόδου (common anode)** οι τιμές που στέλνουμε πρέπει να είναι **αντιστρόφως ανάλογες** με την φωτεινότητα του κάθε χρώματος που θέλουμε να πετύχουμε. Για παράδειγμα, η τιμή 0 στην θύρα 11 θα έχει σαν αποτέλεσμα το κόκκινο χρώμα να πάρει την μέγιστη τιμή του. Αντίθετα, η τιμή 255 θα έχει σαν αποτέλεσμα να μην εκπέμπεται καθόλου κόκκινο χρώμα.

*Η set analog pin λειτουργεί μόνο στις PWM ψηφιακές θύρες οι οποίες στις περισσότερες πλακέτες είναι οι 3, 5, 6, 9, 10, 11. Δηλαδή στις θύρες αυτές μπορούμε να παράγουμε ενδιάμεσες τάσεις στο διάστημα [0, 5] Volt στέλνοντας τιμές από 0 έως 255.*

2. Η εντολή **set analog pin(10, 255)** στέλνει την τάση 5 V στην ψηφιακή θύρα 10. Αυτό έχει σαν αποτέλεσμα το μπλε χρώμα να πάρει την ελάχιστη τιμή του, δηλαδή να μην εκπέμπεται καθόλου μπλε χρώμα.
3. Η εντολή **set analog pin(9, 255)** στέλνει την τάση 5 V στην ψηφιακή θύρα 9. Αυτό έχει σαν αποτέλεσμα το πράσινο χρώμα να πάρει την ελάχιστη τιμή του, δηλαδή να μην εκπέμπεται καθόλου πράσινο χρώμα.

Το αποτέλεσμα της εκτέλεσης των εντολών στις γραμμές 1, 2 και 3 είναι το RGB LED να εκπέμπει το **κόκκινο χρώμα**.

4. Με την εντολή **delay MILLIS milliseconds(1000)** σταματά η εκτέλεση του προγράμματος για χρονικό διάστημα 1000 ms ( 1 sec = 1000 ms ).
5. Η εντολή **set analog pin(11, 255)** στέλνει την τάση 5 V την ψηφιακή θύρα 11. Αυτό έχει σαν αποτέλεσμα να μην εκπέμπεται καθόλου κόκκινο χρώμα.
6. Η εντολή **set analog pin(10, 0)** στέλνει την τάση 0 V στην ψηφιακή θύρα 10. Αυτό έχει σαν αποτέλεσμα το μπλε χρώμα να πάρει την μέγιστη τιμή του.
7. Η εντολή **set analog pin(9, 255)** στέλνει την τάση 5 V στην ψηφιακή θύρα 9. Αυτό έχει σαν αποτέλεσμα το πράσινο χρώμα να πάρει την ελάχιστη τιμή του, δηλαδή να μην εκπέμπεται καθόλου πράσινο χρώμα.

Το αποτέλεσμα της εκτέλεσης των εντολών στις γραμμές 5, 6 και 7 είναι το RGB LED να εκπέμπει το **μπλε χρώμα**.

8. Με την εντολή **delay MILLIS milliseconds(1000)** σταματά η εκτέλεση του προγράμματος για χρονικό διάστημα 1000 ms ( 1 sec = 1000 ms ).
9. Η εντολή **set analog pin(11, 255)** στέλνει την τάση 5 V την ψηφιακή θύρα 11. Αυτό έχει σαν αποτέλεσμα να μην εκπέμπεται καθόλου κόκκινο χρώμα.



**10.** Η εντολή **set analog pin(10, 255)** στέλνει την τάση 5 V στην ψηφιακή θύρα 10. Αυτό έχει σαν αποτέλεσμα το μπλε χρώμα να πάρει την ελάχιστη τιμή του, δηλαδή να μην εκπέμπεται καθόλου μπλε χρώμα.

**11.** Η εντολή **set analog pin(9, 0)** στέλνει την τάση 0 V στην ψηφιακή θύρα 9. Αυτό έχει σαν αποτέλεσμα το πράσινο χρώμα να πάρει την μέγιστη τιμή του.

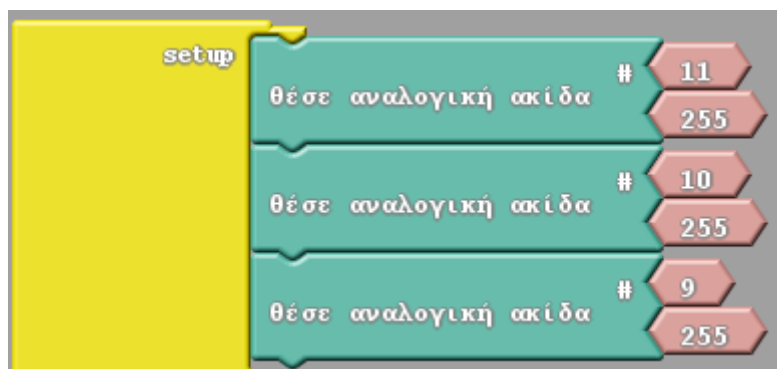
Το αποτέλεσμα της εκτέλεσης των εντολών στις γραμμές 9, 10 και 11 είναι το RGB LED να εκπέμπει το **πράσινο χρώμα**.

**12.** Με την εντολή **delay MILLIS milliseconds(1000)** σταματά η εκτέλεση του προγράμματος για χρονικό διάστημα 1000 ms ( 1 sec = 1000 ms ).

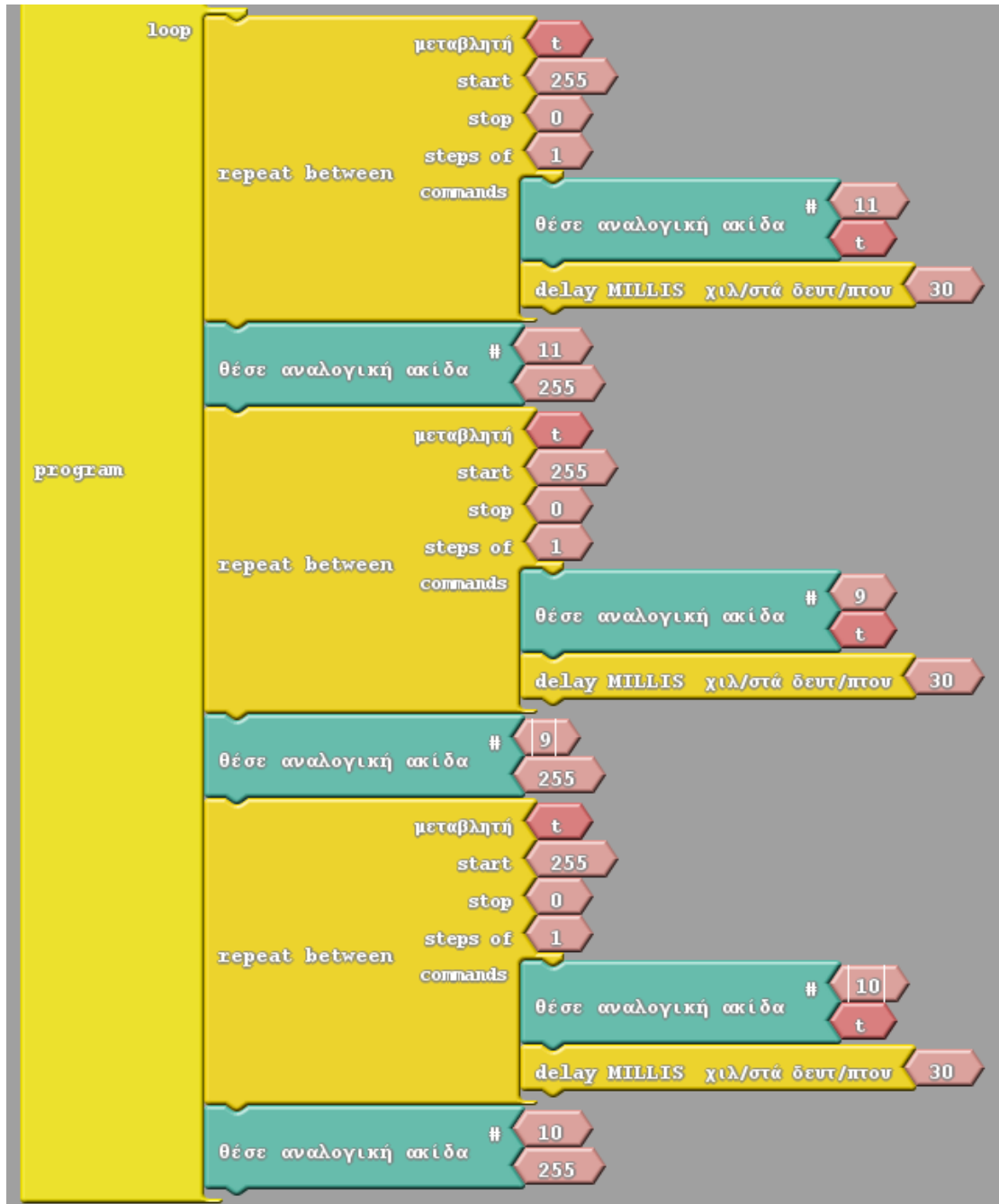


Μια άλλη ενδιαφέρουσα εφαρμογή είναι η σταδιακή αύξηση της φωτεινότητας μόνο ενός σημείου σύνδεσης κάθε φορά. Δηλαδή, να ανεβάζουμε σταδιακά την τάση (από 0V στα 5V) στο σημείο σύνδεσης που αντιστοιχεί στο κόκκινο χρώμα (θύρα 11 στο παραπάνω κύκλωμα), στη συνέχεια να κάνουμε το ίδιο με το πράσινο χρώμα (θύρα 9 στο παραπάνω κύκλωμα) και τέλος να επαναλάβουμε το ίδιο με το μπλε χρώμα (θύρα 10 στο παραπάνω κύκλωμα). Το αποτέλεσμα θα είναι εντυπωσιακό.

Στη setup, στέλνουμε σε όλες τις θύρες μια τάση 0 Volt για να μην εκπέμπει καμιά ακτινοβολία το RGB led.

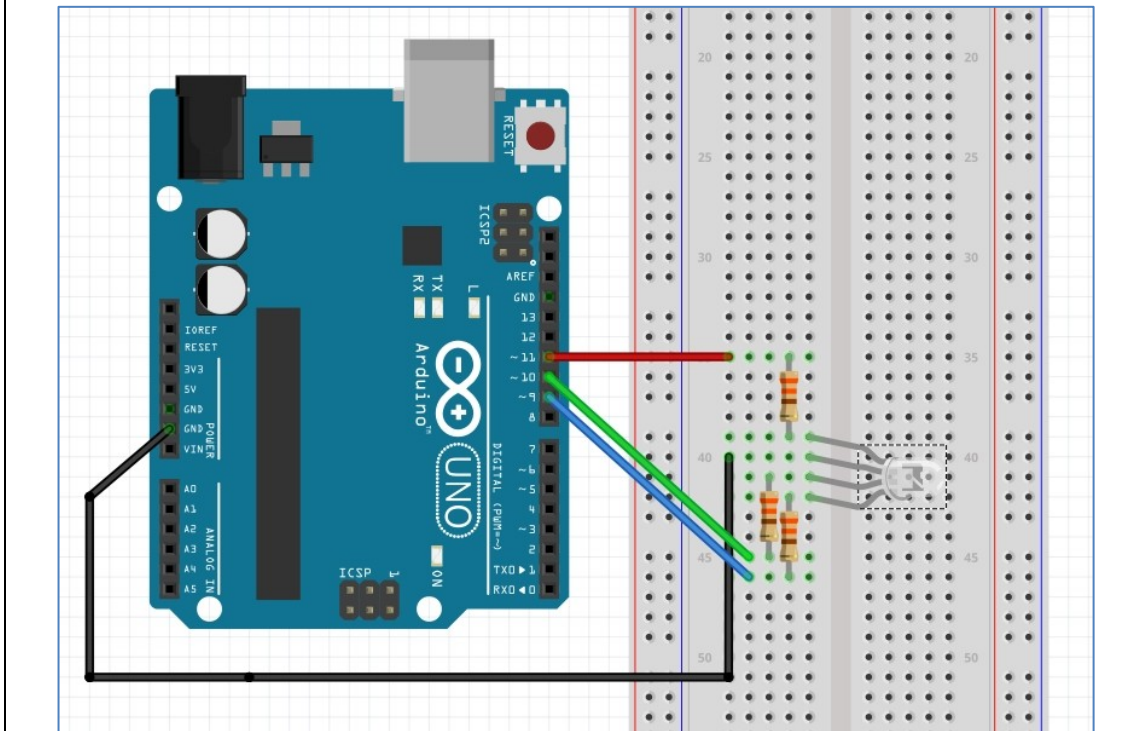


Στη συνάρτηση loop, με την πρώτη εντολή **repeat between** (μεταβλητή: t, **start: 255, stop: 0, steps of: 1**) αυξάνουμε σταδιακά την τάση από 0 Volt στα 5 Volt (εντολή **θέσε αναλογική ακίδα**) στη θύρα 11 και το RGB led ακτινοβολεί κόκκινο φως με τη φωτεινότητα του να αυξάνει σταδιακά. Αμέσως μετά, με την εντολή **θέσε αναλογική ακίδα (11, 255)** στέλνουμε τάση 0 Volt στη θύρα 11 και το RGB led δεν εκπέμπει καμιά ακτινοβολία. Η δεύτερη εντολή repeat κάνει ακριβώς τα ίδια πράγματα στη θύρα 9 με αποτέλεσμα το RGB led να ακτινοβολεί **πράσινο** φως με τη φωτεινότητα του να αυξάνει σταδιακά. Τέλος, με την τρίτη εντολή repeat το RGB led ακτινοβολεί **μπλε** φως με τη φωτεινότητα του να αυξάνει σταδιακά.





Αν χρησιμοποιήσετε ένα RGB LED κοινής καθόδου τότε τον ακροδέκτη της καθόδου (CATHODE) πρέπει να τον συνδέσετε στην γείωση (GND) (αντί για την τάση 5 V που συνδέσαμε στο RGB LED κοινής ανόδου παραπάνω).




## Δραστηριότητες

1. Τροποποιήστε το παραπάνω πρόγραμμα (Εικόνα 4) ώστε εκτός από τα βασικά χρώματα (κόκκινο, μπλε, πράσινο) που εναλλάσσονται ανά ένα δευτερόλεπτο να προστεθεί και το άσπρο χρώμα και να εναλλάσσονται με την εξής σειρά: κόκκινο - μπλε - πράσινο - άσπρο.
2. Τι χρώμα θα εκπέμψει το RGB LED όταν στις θύρες 11, 10 και 9 στείλουμε τις τιμές 0, 255, 0 αντίστοιχα ;

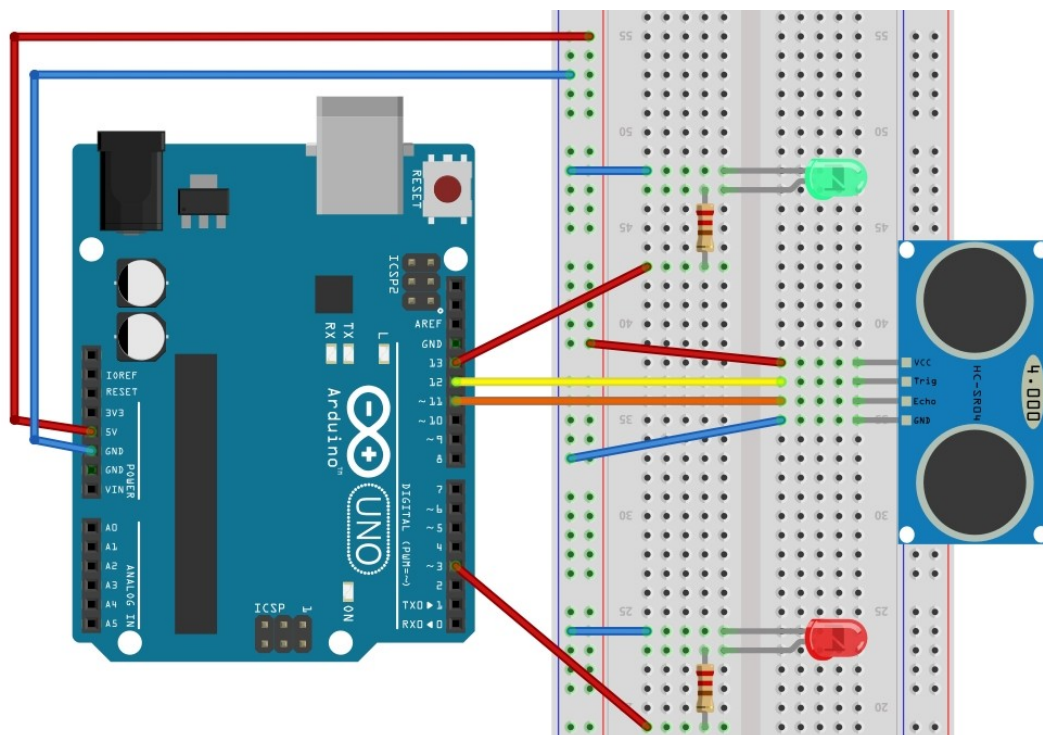
## Φύλλο εργασίας 8 - Ανιχνευτής απόστασης

Σε αυτήν τη δραστηριότητα θα κατασκευάσουμε έναν ανιχνευτή απόστασης. Θα χρησιμοποιήσουμε τον αισθητήρα υπερήχων (ή απόστασης) HC-SR04 για τον υπολογισμό της απόστασης. Στη συγκεκριμένη εφαρμογή, όταν ο αισθητήρας εντοπίσει αντικείμενο σε απόσταση μικρότερη ή ίση των 10 εκατοστών θα ανάβει ένα κόκκινο LED, διαφορετικά θα ανάβει ένα πράσινο LED.

### Υλικά που θα χρειαστούμε:

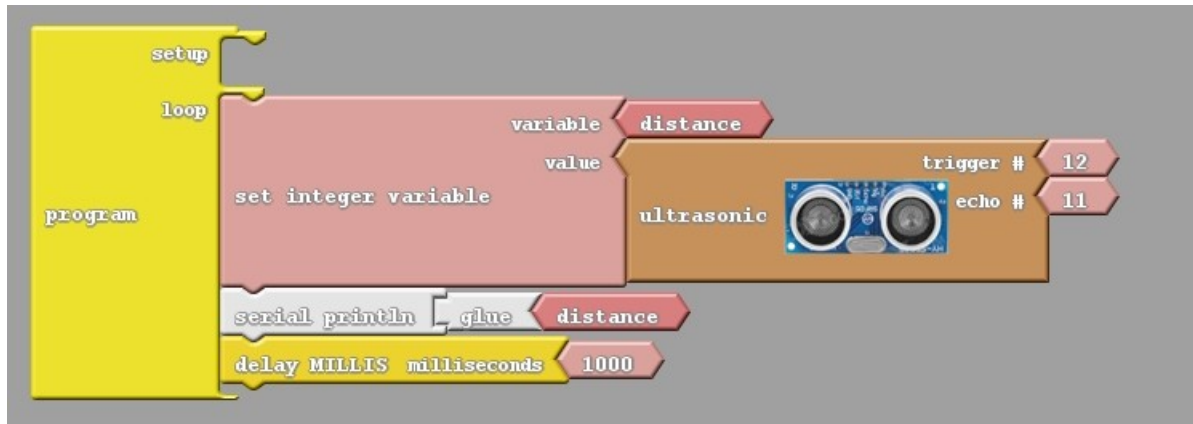
- ✓ 1 Arduino UNO
- ✓ 1 Breadboard
- ✓ 1 αισθητήρα απόστασης (HC-SR04 Ultrasonic Sensor)
- ✓ 1 κόκκινο LED
- ✓ 1 πράσινο LED
- ✓ 2 αντιστάσεις 220 Ω 

### Το κύκλωμα σε breadboard

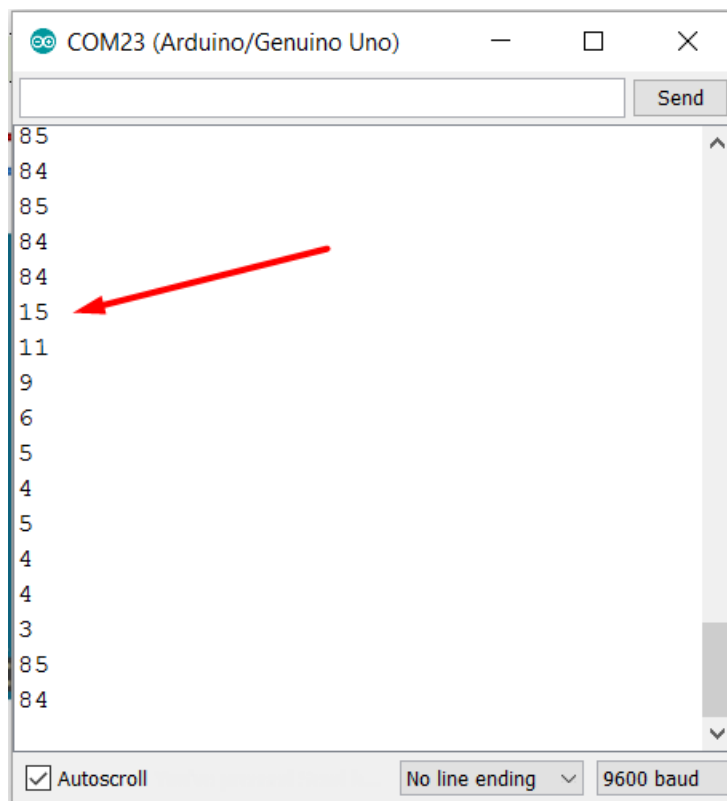


Ο αισθητήρας διαθέτει 4 ακροδέκτες. Ο ακροδέκτης **VCC** συνδέεται στη τροφοδοσία και ο **GND** στη γείωση. Ο ακροδέκτης **Trig** συνδέεται στη θύρα 12 και ο **Echo** στη θύρα 11.

Πριν προχωρήσουμε με την ανάπτυξη του προγράμματος τρέχουμε το παρακάτω πρόγραμμα το οποίο εμφανίζει στην σειριακή οθόνη (Serial Monitor) τις τιμές που επιστρέφει ο αισθητήρας απόστασης.



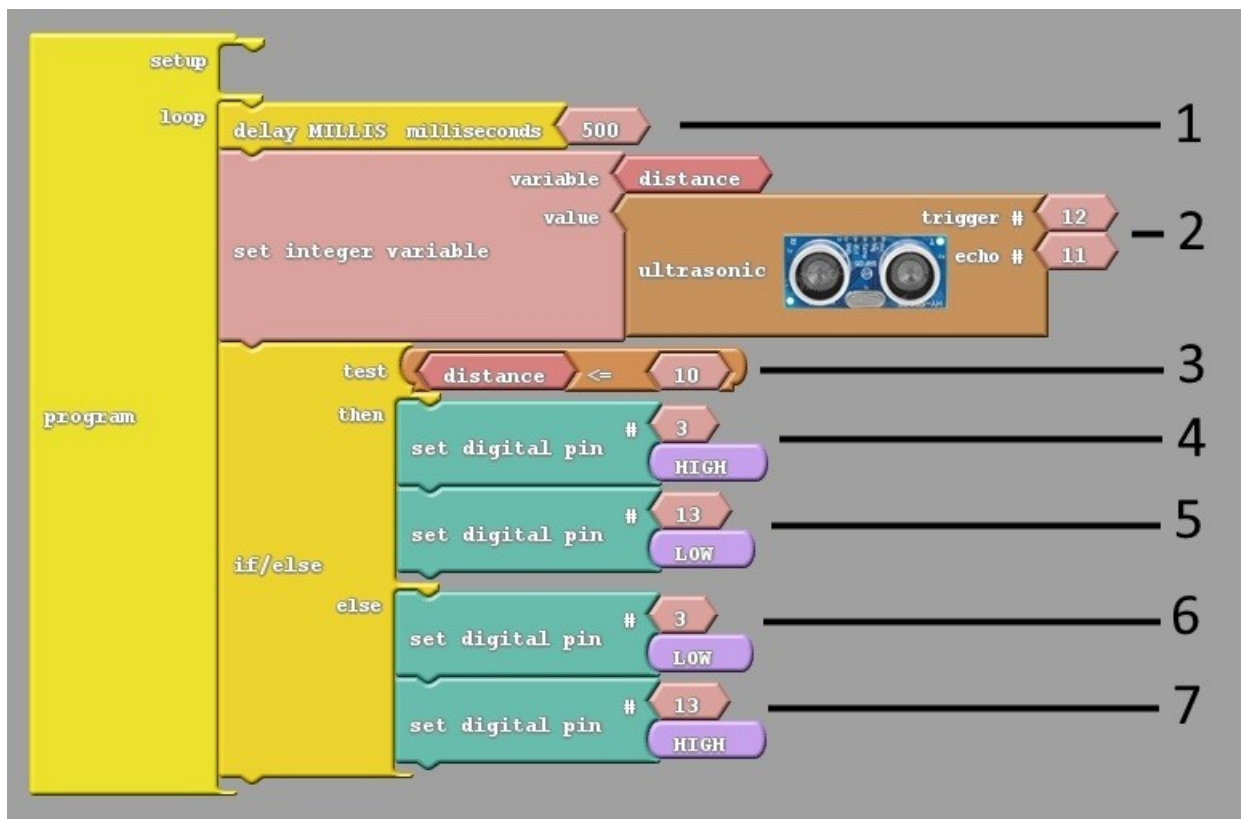
Η σειριακή οθόνη ενεργοποιείται κάνοντας κλικ στο κουμπί "Serial Monitor" του Ardublock.



Αν βάλουμε το χέρι μας μπροστά από τον αισθητήρα απόστασης και το μετακινούμε προς το μέρος του, θα παρατηρήσουμε η τιμή που επιστρέφει μειώνεται. Όταν το χέρι μας βρεθεί σχεδόν μπροστά του, τότε οι τιμές που επιστρέφει είναι μικρότερες του αριθμού 4. Ο αισθητήρας που χρησιμοποιούμε (HC-SR04) μπορεί να μετράει αποστάσεις από 2 εκατοστά μέχρι 400 εκατοστά.

Ακολουθήστε τα παρακάτω βήματα για την ανάπτυξη προγράμματος που θα ανιχνεύει την απόσταση των αντικειμένων που βρίσκονται στο οπτικό πεδίο του αισθητήρα και ανάλογα με την απόσταση του αντικειμένου θα ανάβει ένα κόκκινο ή πράσινο LED :

- 1 Ανοίξτε το προγραμματιστικό περιβάλλον του Arduino.
- 2 Από το μενού **Tools** επίλεξε **Ardublock** για να ξεκινήσετε το προγραμματιστικό περιβάλλον του Ardublock.
- 3 Δημιουργήστε το παρακάτω πρόγραμμα στο περιβάλλον του Ardublock.



**Εικόνα 8** - program8.adp

\* Οι αριθμοί δίπλα σε κάθε εντολή χρησιμοποιούνται στην ανάλυση του προγράμματος που ακολουθεί στην επόμενη παράγραφο.

### Ανάλυση προγράμματος (program8.adp)

\* Κάθε γραμμή αντιστοιχεί σε μια εντολή (με τον ίδιο αριθμό) στην Εικόνα 8.

1. Με την εντολή **delay MILLIS milliseconds(500)** σταματά η εκτέλεση του προγράμματος για χρονικό διάστημα 500 ms ( 1 sec = 1000 ms ).
2. Η εντολή **set integer variable** θέτει στην μεταβλητή **distance** μια τιμή (value). Στη συγκεκριμένη περίπτωση η τιμή αυτή είναι η τιμή που επιστρέφει το μπλοκ **ultrasonic** (στο οποίο περνάμε ως παραμέτρους τις θύρες που έχουμε συνδέσει τον ακροδέκτη **Trig** και τον ακροδέκτη **Echo**).
3. Η εντολή **if ... then ... else ...** ελέγχει αν η τιμή της μεταβλητής **distance** είναι μικρότερη ή ίση του 10. Αν είναι τότε εκτελούνται οι εντολές στις γραμμές 4 και 5, αλλιώς εκτελούνται οι εντολές στις γραμμές 6 και 7.
4. Η εντολή **set digital pin( 3, HIGH )** στέλνει μια τάση 5 V στην ψηφιακή θύρα 3. Αυτό έχει σαν αποτέλεσμα να ανάψει το κόκκινο LED.
5. Η εντολή **set digital pin( 13, LOW )** στέλνει μια τάση 0 V στην ψηφιακή θύρα 13. Αυτό έχει σαν αποτέλεσμα να σβήσει το πράσινο LED.
6. Η εντολή **set digital pin( 3, LOW )** στέλνει μια τάση 0 V στην ψηφιακή θύρα 3. Αυτό έχει σαν αποτέλεσμα να σβήσει το κόκκινο LED.
7. Η εντολή **set digital pin( 13, HIGH )** στέλνει μια τάση 5 V στην ψηφιακή θύρα 13. Αυτό έχει σαν αποτέλεσμα να ανάψει το πράσινο LED.

### Δραστηριότητες

1. Ας θεωρήσουμε ότι το κάθε LED αντιστοιχεί σε ένα bit. Επίσης, ας θεωρήσουμε ότι όταν το LED είναι σβηστό έχει την τιμή 0 και όταν είναι αναμμένο έχει την τιμή 1. Όλοι οι δυνατοί συνδυασμοί των δύο LED φαίνονται στον παρακάτω πίνακα:

Κόκκινο LED	Πράσινο LED	Κωδικοποίηση
Αναμμένο	Αναμμένο	11
Αναμμένο	Σβηστό	10
Σβηστό	Αναμμένο	01
Σβηστό	Σβηστό	00

Τροποποιήστε το παραπάνω πρόγραμμα ώστε τα LED να ανάβουν σύμφωνα με τον παρακάτω πίνακα:

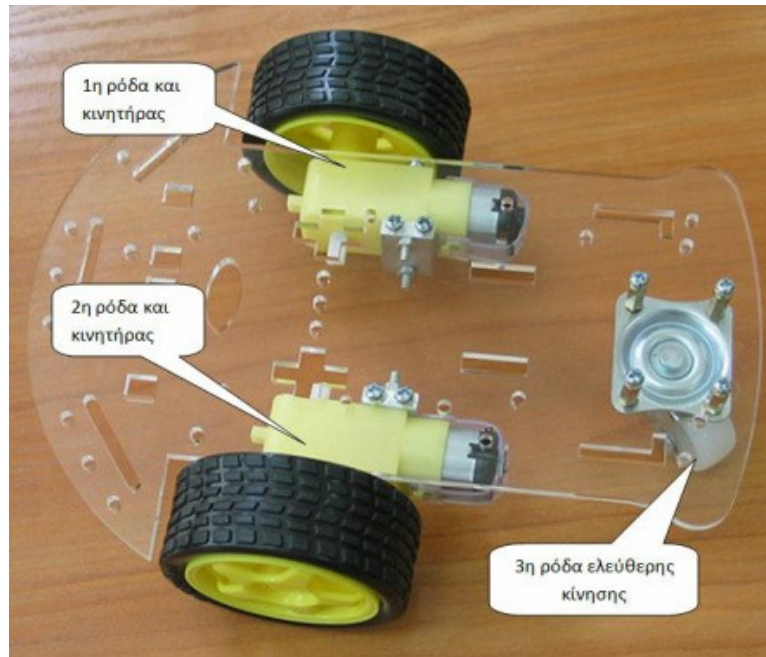
Κωδικοποίηση	Απόσταση (τιμή μεταβλητής distance)
11	απόσταση $\leq 10$ εκ.
10	$10 < \text{απόσταση} \leq 40$ εκ.
01	$40 < \text{απόσταση} \leq 60$ εκ.
00	$60 < \text{απόσταση}$



## Φύλλο εργασίας 9 - Αυτόνομο ρομποτικό όχημα αποφυγής εμποδίων

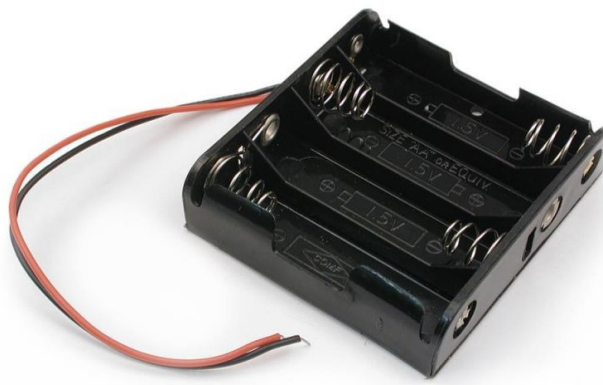
Σε αυτήν τη δραστηριότητα θα κατασκευάσουμε ένα αυτόνομο ρομποτικό όχημα αποφυγής εμποδίων. Εκτός από τον μικροελεγκτή Arduino, το breadboard, έναν αισθητήρα απόστασης HC-SR04 και τα υπόλοιπα στοιχεία του κυκλώματος, για την δημιουργία του ρομποτικού οχήματος θα χρησιμοποιήσουμε τα παρακάτω μέρη:

- ένα σασί με 2 μοτέρ (πάνω στα οποία συνδέονται 2 ρόδες) και μια ρόδα ελεύθερης κίνησης



Εικόνα 9.1

- μια θήκη για 4 AA μπαταρίες (η οποία θα τοποθετηθεί πάνω στο σασί)



Εικόνα 9.2

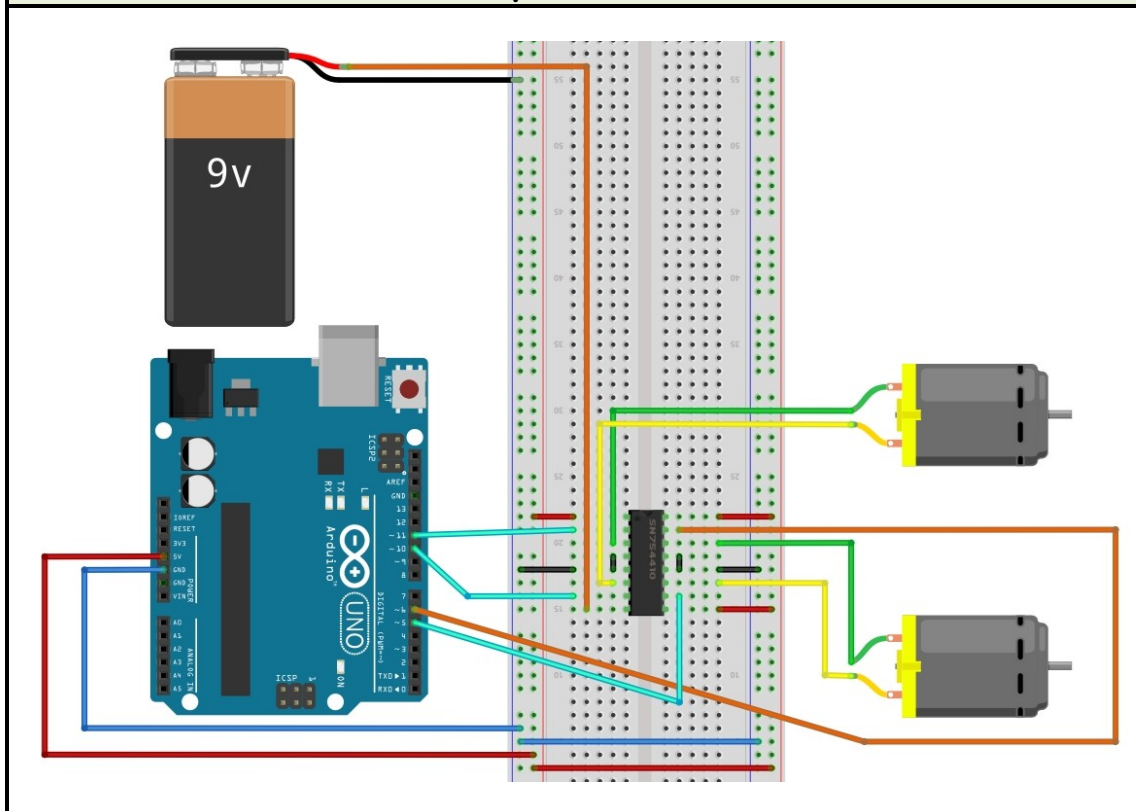
Το ρομποτικό όχημα θα κινείται συνεχώς προς τα εμπρός έως ότου ο αισθητήρας απόστασης εντοπίσει κάποιο εμπόδιο σε απόσταση μικρότερη ή ίση των 10

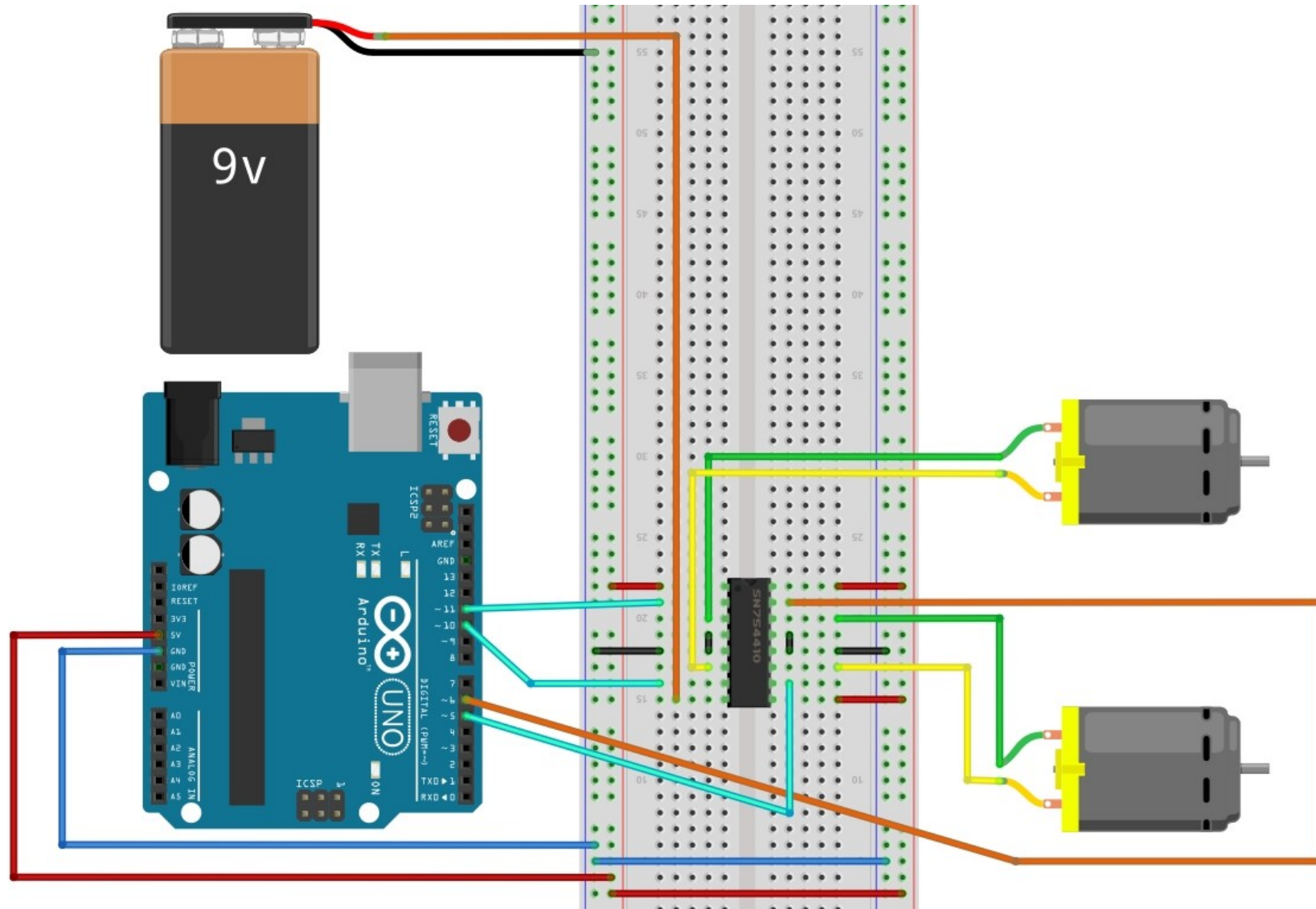
εκατοστών. Όταν ο αισθητήρας απόστασης εντοπίσει κάποιο εμπόδιο, τότε το ρομπότ στρίβει αριστερά ή δεξιά (η επιλογή γίνεται με τυχαίο τρόπο) και στη συνέχεια κινείται προς τα εμπρός. Η διαδικασία αυτή θα επαναλαμβάνεται συνεχώς.

**Υλικά που θα χρειαστούμε:**

- ✓ 1 Arduino UNO
- ✓ 1 Breadboard
- ✓ 1 αισθητήρα απόστασης (HC-SR04 Ultrasonic Sensor)
- ✓ 1 σασί με δύο μοτέρ με ρόδες και μία ρόδα ελεύθερης κίνησης
- ✓ 1 θήκη για 4 AA μπαταρίες (για την τροφοδοσία των μοτέρ)
- ✓ 1 ολοκληρωμένο κύκλωμα SN754410 (που θα χρησιμοποιηθεί για την οδήγηση των 2 μοτέρ)

**Το κύκλωμα σε breadboard**





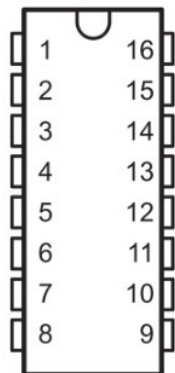
Εικόνα 9.3

Οι ακροδέκτες 2 και 7 του ολοκληρωμένου κυκλώματος SN754410 ελέγχουν την φορά του πρώτου μοτέρ που έχουμε συνδέσει σύμφωνα με τον παρακάτω πίνακα:

Ακροδέκτης 2	Ακροδέκτης 7	Φορά περιστροφής
LOW	HIGH	δεξιά
HIGH	LOW	αριστερά
LOW	LOW	δεν περιστρέφεται
HIGH	HIGH	δεν περιστρέφεται

Οι ακροδέκτες 10 και 15 του ολοκληρωμένου κυκλώματος SN754410 ελέγχουν την φορά του δεύτερου μοτέρ που έχουμε συνδέσει σύμφωνα με τον παρακάτω πίνακα:

Ακροδέκτης 10	Ακροδέκτης 15	Φορά περιστροφής
LOW	HIGH	δεξιά
HIGH	LOW	αριστερά
LOW	LOW	δεν περιστρέφεται
HIGH	HIGH	δεν περιστρέφεται

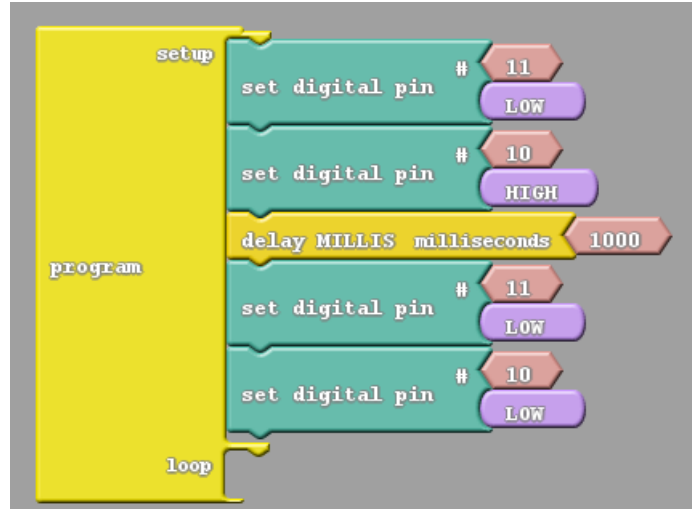


**Εικόνα 9.4.** Ολοκληρωμένο κύκλωμα SN754410

Αν προσέξετε την **Εικόνα 9.3** θα διαπιστώσετε ότι τον ακροδέκτη 2 του ολοκληρωμένου κυκλώματος τον έχουμε συνδέσει στην θύρα 11 του Arduino και τον ακροδέκτη 7 στην θύρα 10. Επίσης τον ακροδέκτη 10 τον έχουμε συνδέσει στη θύρα 5 και τον 15 στην θύρα 6 αντίστοιχα.

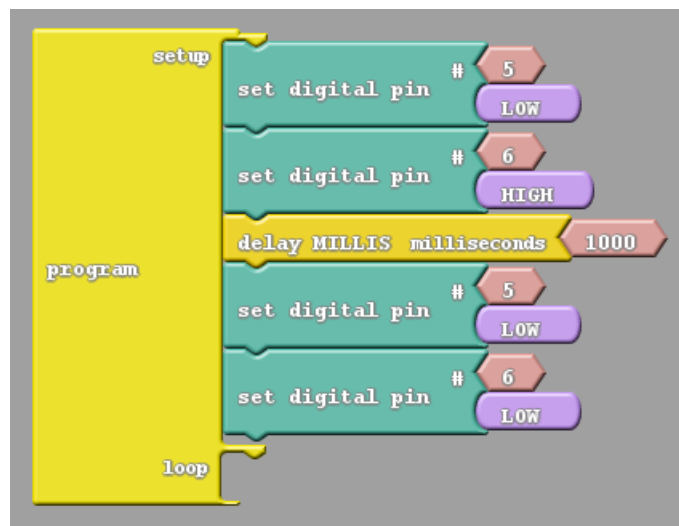
Ακροδέκτης (SN754410)	Θύρα (στον Arduino)
2	11
7	10
10	5
15	6

Για να διαπιστώσουμε ότι έχουμε κάνει σωστά όλες τις συνδέσεις δημιουργούμε και φορτώνουμε το παρακάτω πρόγραμμα στον μικροελεγκτή. Σύμφωνα με τους παραπάνω πίνακες το **πρώτο μοτέρ** θα πρέπει να περιστρέφεται δεξιόστροφα για ένα δευτερόλεπτο.



Εικόνα 9.5 - program9-motor1.adp

Εκτελούμε το παρακάτω πρόγραμμα για να ελέγξουμε ότι δουλεύει και το **δεύτερο μοτέρ** κανονικά (περιστρέφεται δεξιόστροφα για ένα δευτερόλεπτο).

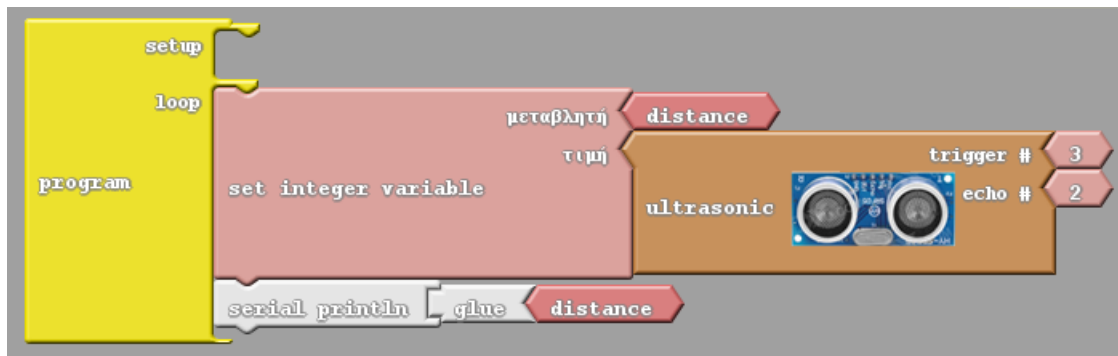


Εικόνα 9.6 - program9-motor2.adp

Στο σημείο αυτό συνδέουμε τον αισθητήρα απόστασης στο κύκλωμά μας ως εξής:

Ακροδέκτης αισθητήρα απόστασης	Θύρα στον μικροελεγκτή Arduino
VCC	Τροφοδοσία
GND	Γείωση
Trig	3
Echo	2

Πριν προχωρήσουμε με την ανάπτυξη του τελικού προγράμματος τρέχουμε το παρακάτω πρόγραμμα το οποίο εμφανίζει στην σειριακή οθόνη (Serial Monitor) τις τιμές που επιστρέφει ο αισθητήρας απόστασης (Η σειριακή οθόνη ενεργοποιείται κάνοντας κλικ στο κουμπί "**Serial Monitor**" του Ardublock).

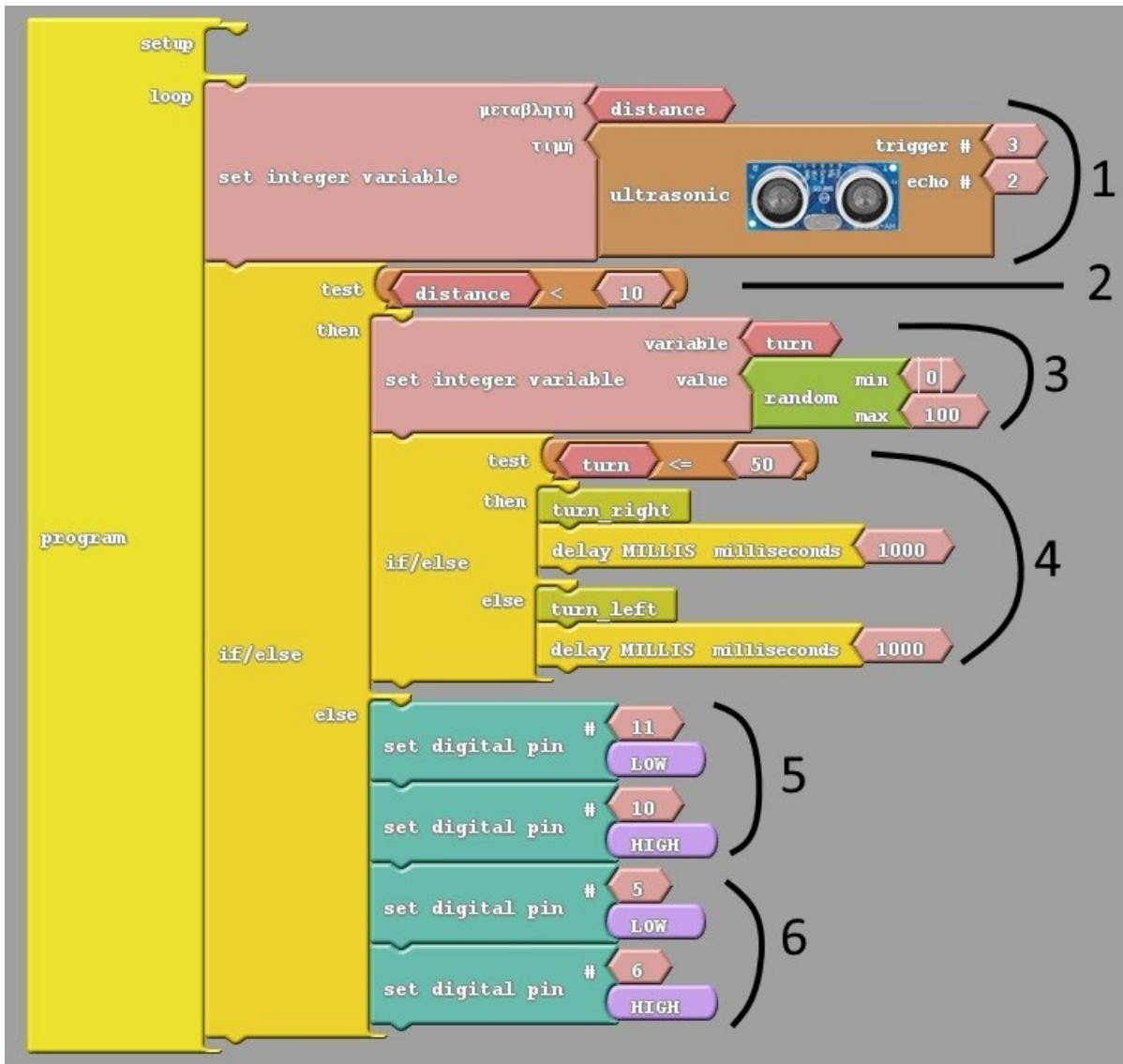


Εικόνα 9.7 - program9-monitor.adp

Ελέγχουμε τις τιμές που επιστρέφει ο αισθητήρας απόστασης στην σειριακή οθόνη και επιβεβαιώνουμε ότι δουλεύει σωστά.

Ακολουθήστε τα παρακάτω βήματα για την ανάπτυξη του τελικού προγράμματος που θα ελέγχει το ρομποτικό όχημα:

- 1** Ανοίξτε το προγραμματιστικό περιβάλλον του Arduino.
- 2** Από το μενού **Tools** επίλεξε **Ardublock** για να ξεκινήσετε το προγραμματιστικό περιβάλλον του Ardublock.
- 3** Δημιουργήστε το παρακάτω πρόγραμμα στο περιβάλλον του Ardublock.



Εικόνα 9.8 - program9.adp

\* Οι αριθμοί δίπλα σε κάθε εντολή χρησιμοποιούνται στην ανάλυση του προγράμματος που ακολουθεί στην επόμενη παράγραφο.

### Ανάλυση προγράμματος (program9.adp)

\* Κάθε γραμμή αντιστοιχεί σε μια εντολή (με τον ίδιο αριθμό) στην Εικόνα 9.8.

1. Η εντολή **set integer variable** θέτει στην μεταβλητή **distance** μια τιμή (value). Στη συγκεκριμένη περίπτωση η τιμή αυτή είναι η τιμή που επιστρέφει το μπλοκ **ultrasonic** (στο οποίο περνάμε ως παραμέτρους τις θύρες που έχουμε συνδέσει τον ακροδέκτη **Trig** και τον ακροδέκτη **Echo**).
2. Η εντολή **if ... then ... else ...** ελέγχει αν η τιμή της μεταβλητής **distance** είναι μικρότερη ή ίση του 10. Αν είναι τότε εκτελούνται οι εντολές στις γραμμές 3 και 4, αλλιώς εκτελούνται οι εντολές στις γραμμές 5 και 6.

3. Η εντολή **set integer variable** θέτει στην μεταβλητή **turn** μια τιμή (value). Στη συγκεκριμένη περίπτωση η τιμή αυτή είναι η τιμή που επιστρέφει η εντολή **random( 0, 100)**. Η εντολή **random** επιστρέφει τυχαία μια τιμή στο διάστημα [0, 100].
4. Η εντολή **if ... then ... else ...** ελέγχει αν η τιμή της μεταβλητής **turn** είναι μικρότερη ή ίση του 50. Αν είναι τότε εκτελείται η εντολή **turn\_right** (το ρομπότ στρίβει δεξιά) και στη συνέχεια με την εντολή **delay MILLIS milliseconds(1000)** σταματά η εκτέλεση του προγράμματος για χρονικό διάστημα 1000 ms ( 1 sec = 1000 ms ). Διαφορετικά ( $turn > 50$ ), εκτελείται η εντολή **turn\_left** (το ρομπότ στρίβει αριστερά) και στη συνέχεια με την εντολή **delay MILLIS milliseconds(1000)** σταματά η εκτέλεση του προγράμματος για χρονικό διάστημα 1000 ms ( 1 sec = 1000 ms ).

Οι εντολές **turn\_right** και **turn\_left** είναι εντολές που έχουμε δημιουργήσει εμείς (και θα αναλύσουμε στη συνέχεια) χρησιμοποιώντας το μπλοκ



που βρίσκεται στην ομάδα εντολών **Control**. Το περιεχόμενο των εντολών **turn\_right** και **turn\_left** μπορείτε να το δείτε στις επόμενες παραγράφους.

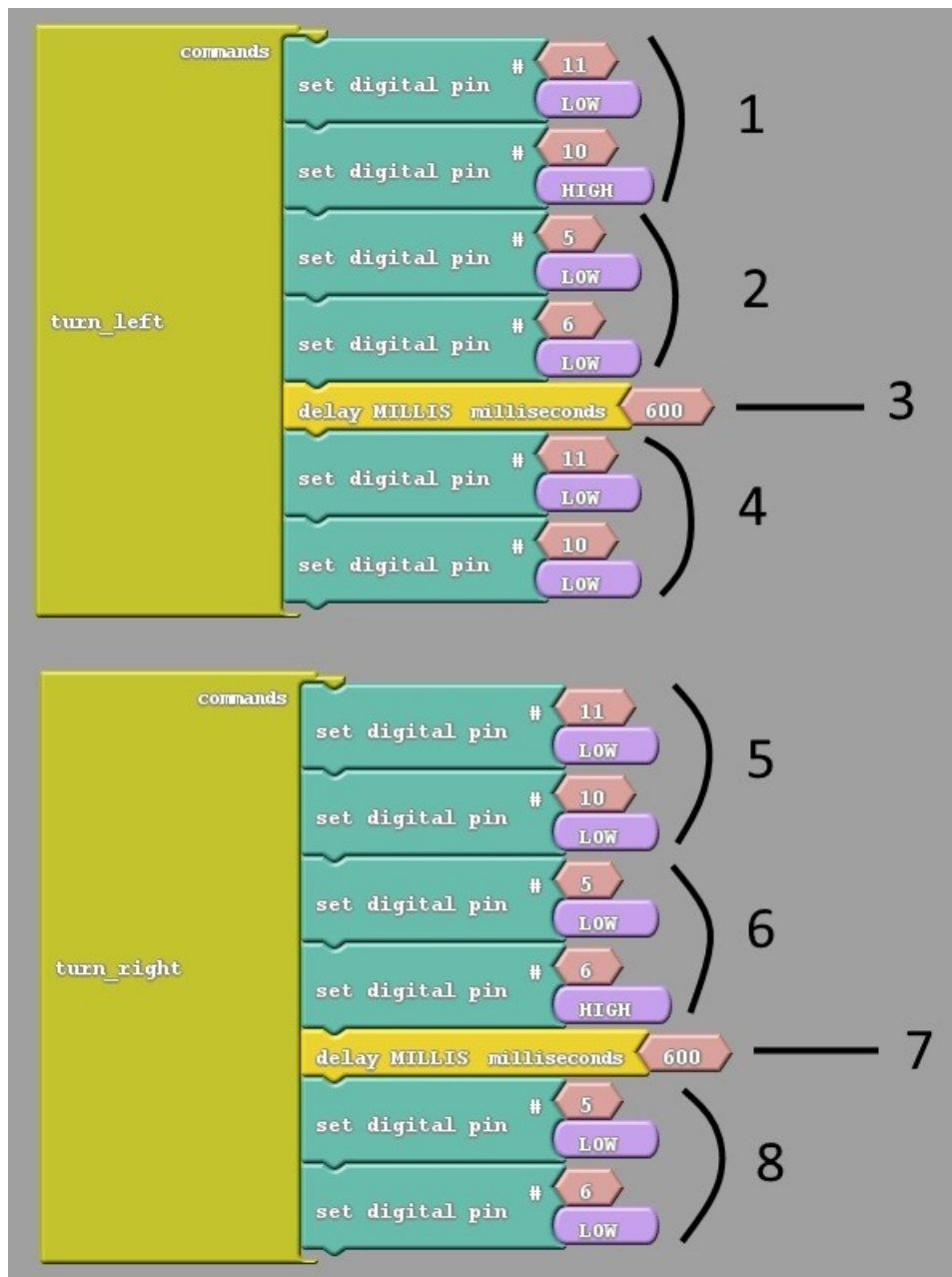
Για να δημιουργήσουμε μια νέα εντολή σέρνουμε το μπλοκ **subroutine** (που βρίσκεται στην ομάδα **Control**) στην περιοχή προγραμματισμού του Ardublock. Στη συνέχεια κάνουμε κλικ στην λέξη **subroutine** και πληκτρολογούμε το όνομα της νέας εντολής. Στην περιοχή **commands** εισάγουμε τις εντολές που επιθυμούμε να εκτελούνται κάθε φορά που θα γίνεται κλήση αυτής της εντολής.



5. Οι εντολές **set digital pin(11, LOW)** και **set digital pin(10, HIGH)** περιστρέφουν το πρώτο μοτέρ δεξιόστροφα.
6. Οι εντολές **set digital pin(5, LOW)** και **set digital pin(6, HIGH)** περιστρέφουν το δεύτερο μοτέρ δεξιόστροφα.



### Εντολές (διαδικασίες) `turn_left` και `turn_right`



Εικόνα 9.9 - program9.adp

\* Οι αριθμοί δίπλα σε κάθε εντολή χρησιμοποιούνται στην ανάλυση του προγράμματος που ακολουθεί στην επόμενη παράγραφο.

### Ανάλυση προγράμματος (program9.adp) Εντολή (διαδικασία) **turn\_left**

\* Κάθε γραμμή αντιστοιχεί σε μια εντολή (με τον ίδιο αριθμό) στην Εικόνα 9.9.

1. Οι εντολές **set digital pin(11, LOW)** και **set digital pin(10, HIGH)** περιστρέφουν το πρώτο μοτέρ δεξιόστροφα. Το μοτέρ περιστρέφεται συνεχώς μέχρι να εκτελεστούν άλλες εντολές που να σταματούν την περιστροφή του.
2. Οι εντολές **set digital pin(5, LOW)** και **set digital pin(6, LOW)** σταματούν την κίνηση του δεύτερου μοτέρ.
3. Με την εντολή **delay MILLIS milliseconds(600)** σταματά η εκτέλεση του προγράμματος για χρονικό διάστημα 600 ms ( 1 sec = 1000 ms ). Το πρώτο μοτέρ συνεχίζει να περιστρέφεται.
4. Οι εντολές **set digital pin(11, LOW)** και **set digital pin(10, LOW)** σταματούν την κίνηση του πρώτου μοτέρ.

Η εντολή **turn\_left** στρίβει το ρομποτικό όχημα προς τα αριστερά ακινητοποιώντας το ένα μοτέρ (που βρίσκεται στα αριστερά του οχήματος) και περιστρέφοντας το άλλο μοτέρ (που βρίσκεται στα δεξιά του οχήματος).

### Ανάλυση προγράμματος (program9.adp) Εντολή (διαδικασία) **turn\_right**

\* Κάθε γραμμή αντιστοιχεί σε μια εντολή (με τον ίδιο αριθμό) στην Εικόνα 9.9.

5. Οι εντολές **set digital pin(11, LOW)** και **set digital pin(10, LOW)** σταματούν την κίνηση του πρώτου μοτέρ.
6. Οι εντολές **set digital pin(5, LOW)** και **set digital pin(6, HIGH)** περιστρέφουν το δεύτερο μοτέρ δεξιόστροφα. Το μοτέρ περιστρέφεται συνεχώς μέχρι να εκτελεστούν άλλες εντολές που να σταματούν την περιστροφή του.
7. Με την εντολή **delay MILLIS milliseconds(600)** σταματά η εκτέλεση του προγράμματος για χρονικό διάστημα 600 ms ( 1 sec = 1000 ms ). Το δεύτερο μοτέρ συνεχίζει να περιστρέφεται.
8. Οι εντολές **set digital pin(5, LOW)** και **set digital pin(6, LOW)** σταματούν την κίνηση του δεύτερου μοτέρ.

Η εντολή **turn\_right** στρίβει το ρομποτικό όχημα προς τα δεξιά ακινητοποιώντας το ένα μοτέρ (που βρίσκεται στα δεξιά του οχήματος) και περιστρέφοντας το άλλο μοτέρ (που βρίσκεται στα αριστερά του οχήματος).

## Δραστηριότητες

1. Προσθέστε δύο LED στο ρομποτικό σας όχημα. Το ένα στην αριστερή πλευρά και το άλλο στην δεξιά πλευρά. Όταν το όχημα στρίβει προς τα δεξιά να ανάβει το δεξί LED και όταν στρίβει προς τα αριστερά να ανάβει το αριστερό LED.
2. Τροποποιήστε το πρόγραμμα της προηγούμενης δραστηριότητας ώστε τα LED να αναβοσβήνουν όταν το ρομποτικό όχημα στρίβει.