**ΦΥΛΛΟ ΕΡΓΑΣΙΑΣ**

**Σύνδεση και προγραμματισμός σερβοκινητήρα TowerPro SG90**

**Δραστηριότητα 1η**

Απαντήστε στα παρακάτω ερωτήματα με την ομάδα σας αφού πρώτα κάνετε αναζήτηση στο διαδίκτυο:

1. **Ποια η λειτουργία ενός κινητήρα servo και σε ποιες εφαρμογές τον εντοπίζουμε;**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

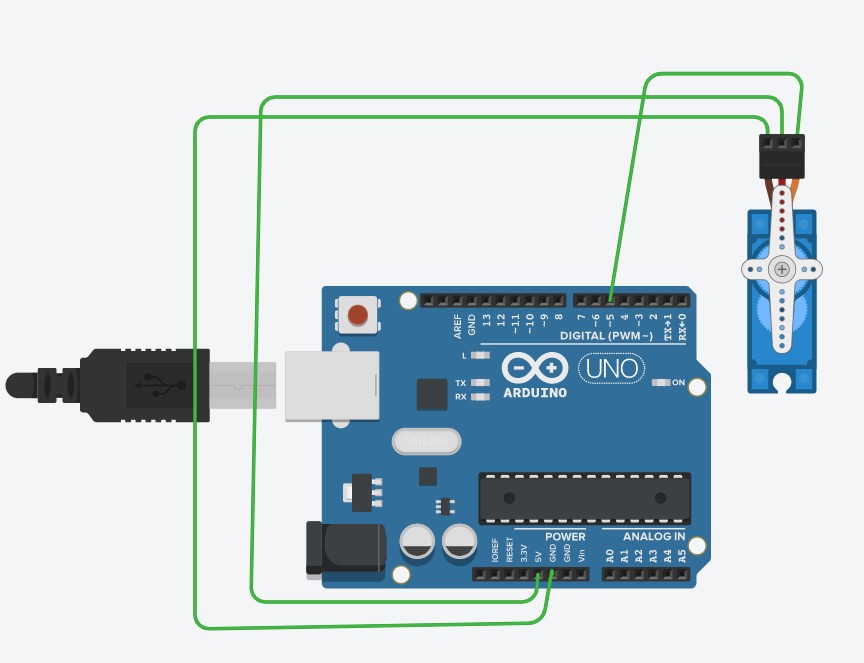
**β. Βρείτε τον τρόπο συνδεσμολογίας του σέρβο TowerPro SG90 που θα συνδεθεί στο Arduino και αναγνωρίστε τα χρώματα των καλωδίων:**

|  |  |  |
| --- | --- | --- |
| **α/α** | **Χρώμα Καλωδίου** | **Χρήση** |
| **1** |  |  |
| **2** |  |  |
| **3** |  |  |

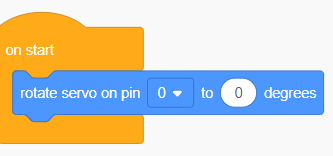
****

**Δραστηριότητα 2η**

**α.** Να σχεδιάσετε το παρακάτω κύκλωμα με το σέρβο TowerPro SG90 στο Tinkercad αλλά και στο κιτ Arduino – προσοχή, αλλάξτε και τα χρώματα των καλωδίων, να είναι ίδια με του σερβοκινητήρα.

****

**β.** Προγραμματίστε το servo σύμφωνα με το παρακάτω πρόγραμμα και κάντε τις απαραίτητες συμπληρώσεις (βρείτε σε ποιο pin είναι συνδεδεμένο το servo) και αλλάξτε τις μοίρες σύμφωνα με τον παρακάτω πίνακα του γ’ μέρους της δραστηριότητας.

****

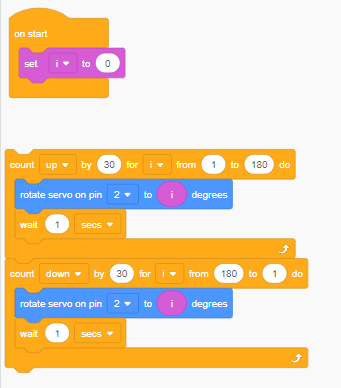
**γ.** Αλλάξτε τις μοίρες περιστροφής σε 00, 900, 180ο, 270ο, 360ο, τι παρατηρείτε σε κάθε περίπτωση; Ποια τα όρια λειτουργίας του servo;

|  |  |
| --- | --- |
| **Μοίρες** | **Mοίρες Περιστροφής** |
| 0 |  |
| 90 |  |
| 180 |  |
| 270 |  |
| 360 |  |

**Όρια λειτουργίας Servo :** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

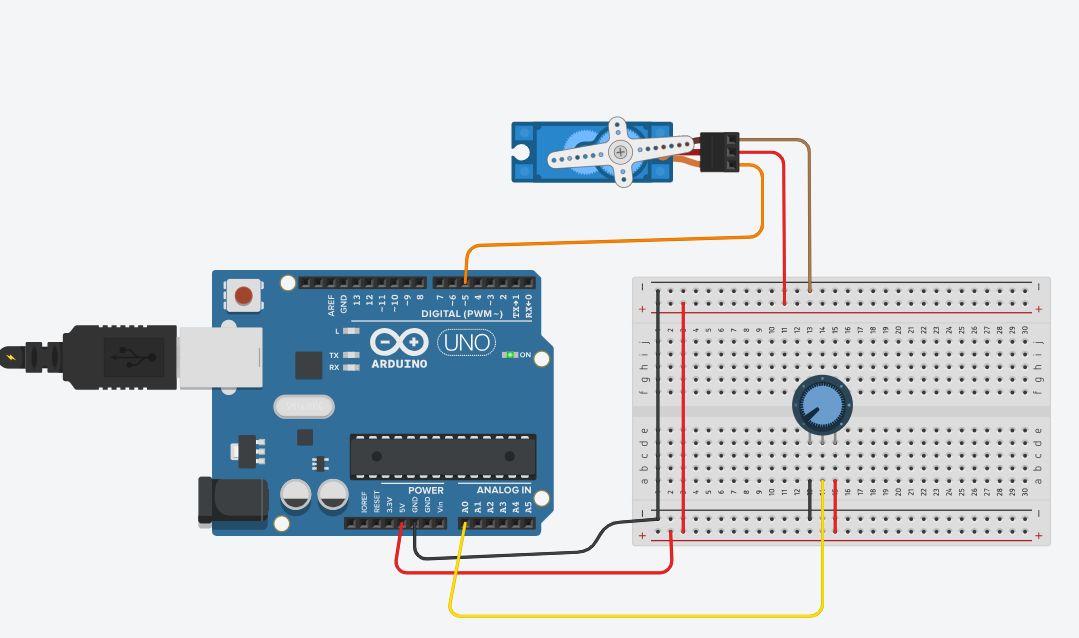
**δ.** Μετατρέψτε τα blocks σε κώδικα και αφού συναρμολογήσετε το servo στο Arduino και φορτώστε το πρόγραμμα να ελέγξετε την λειτουργία του σύμφωνα με το πίνακα που συμπληρώσατε παραπάνω.

**Δραστηριότητα 2η – Αυτόματη περιστροφή servo**

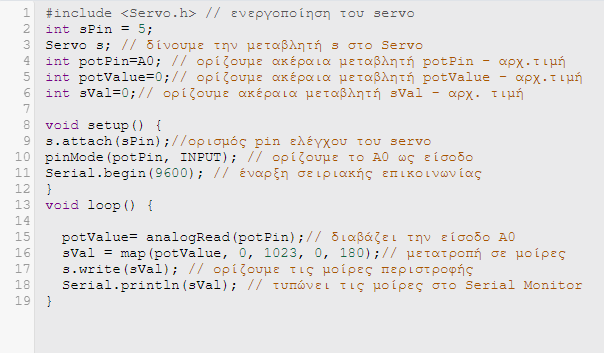
Σκοπός είναι να προγραμματίσουμε το servo ώστε ο άξονάς του να μετακινείται ανά 5 μοίρες, να φτάνει στις 180 και με τον ίδιο τρόπο να γυρίζει στην αρχική του θέση. Δημιουργήστε το παρακάτω πρόγραμμα, τρέξτε το στην προσομοίωση και κάντε τις απαραίτητες αλλαγές (5μοίρες ανά 1 sec).

**Δραστηριότητα 3η – Περιστροφή του servo με ποτενσιόμετρο**

**α.** Φτιάξτε το παρακάτω κύκλωμα

****

**β. Προγραμματίστε το κύκλωμα:**

****

**Eντολή sVal = map (potValue, 0 , 1023, 0, 180)**

H εντολή **map** είναι πολύ χρήσιμη όταν θέλουμε να κάνουμε μετατροπή των τιμών που έρχονται σε μια είσοδο του Arduino από έναν αισθητήρα ή aπό ένα ποτενσιόμετρο και να τις μετατρέψουμε σε τιμές συγκεκριμένων μεγεθών, ανάλογα με το πρόβλημα που έχουμε.

Συγκεκριμένα εδώ, ζητάμε οι τιμές που δίνει το ποτενσιόμετρο (μέσω της μεταβλητής potValue) οι οποίες όπως έχουμε δει είναι μεταξύ 0-1023 να τις μετατρέψουμε σε μοίρες για να ελέγχουμε την κίνηση του σερβοκινητήρα (0-180ο). και για να γνωρίζουμε κάθε φορά πόσες μοίρες έχει περιστραφεί ο άξονας του σερβοκινητήρα.

Το αποτέλεσμα της εντολής, αποθηκεύεται στην μεταβλητή **sVal**, η οποία μας δίνει την πληροφορία των μοιρών περιστροφής.

Για να δούμε τον αριθμό των μοιρών εκτελείται η εντολή:

Serial.println(sVal)

γ. Συμπληρώστε το παρακάτω πίνακα:

|  |  |
| --- | --- |
| **ΘΕΣΗ ΠΟΤΕΝΣΙΟΜΕΤΡΟΥ** | **ΜΟΙΡΕΣ** |
| **1** |  |
| **2** |  |
| **3** |  |
| **4** |  |
| **5** |  |
| **6** |  |
| **7** |  |