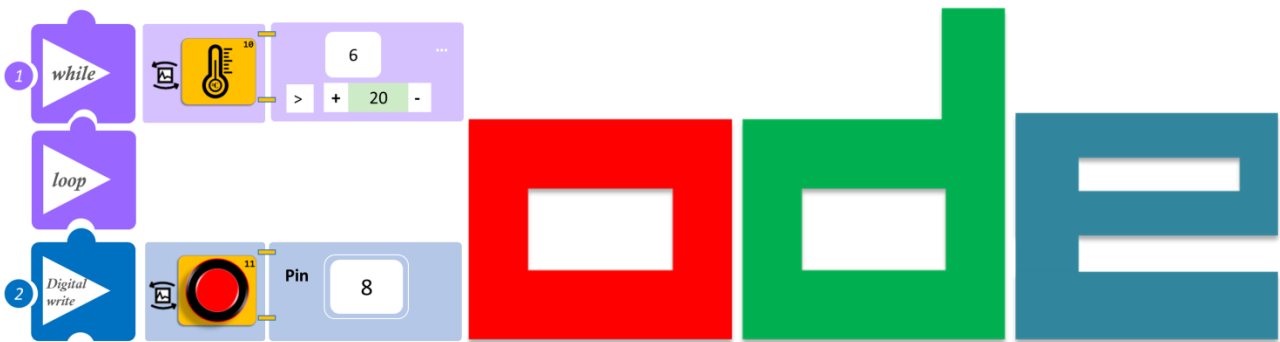


# S2

## ΕΝΟΤΗΤΑ 2 Προγραμματισμός - Οδηγός Χρήσης της Εφαρμογής



The screenshot shows the ARD:icon software interface. The top bar displays 'ARD:icon POLYTECH S.A. ©2023'. Below the bar are several tabs: Παραγωγή, Φόρτωση, Αποθήκευση, Επαναφορά, Πίνακας Ελέγχου, Τερματικό, and Προσομοίωση. The main area is divided into two sections. On the left is a hardware diagram showing an ARD:icon board connected to various components: three digital pins (D8, D7, D5) with red, green, and black LEDs, a digital pin (D9) with a green LED, and four analog pins (A0, A1, A2, A3). On the right is a code editor with the following code:

```
1 void setup () {
2   pinMode( 9 , OUTPUT );
3   pinMode( 8 , OUTPUT );
4   pinMode( 7 , OUTPUT );
5   pinMode( 6 , OUTPUT );
6   Serial.begin(9600);
7 }
8
9 //variables
10 void loop () {
11
12   //if (analogRead(1) > 10) {
13
14   #include <stdio.h>
15   #include <util/delay.h>
16   #include <stdint.h>
17   //startmain
18   int uart_putchar(char c, FILE*
19   stream);
20   void uart_init(unsigned long int
21   baud);
22   long analogReadStable(int
23   sensorpin);
24   void serialPrint(int reader);
25   int main()
26 }
```

SMART:BLOX

## Περιεχόμενα

<b>ΕΝΟΤΗΤΑ 2 Προγραμματισμός - Οδηγός Χρήσης της Εφαρμογής.....</b>	<b>1</b>
<b>1.1 Εισαγωγή στο ARD:icon.....</b>	<b>3</b>
<b>1.1.1 Εγκατάσταση της εφαρμογής ARD:icon.....</b>	<b>3</b>
<b>1.1.2 Εφαρμογή ARD:icon.....</b>	<b>7</b>
<b>1.2 Σύνδεση Ελεγκτή και Περιφερειακών μονάδων.....</b>	<b>11</b>
<b>1.2.1 Σωστή συναρμολόγηση των περιφερειακών συσκευών με τα τουβλάκια .....</b>	<b>12</b>
<b>1.3 Εντολές Προγραμματισμού .....</b>	<b>16</b>
<b>1.3.1. Μενού Επιλογής Εντολών Λειτουργίας.....</b>	<b>17</b>
<b>1.3.2. Ανάλυση Εντολών Λειτουργίας .....</b>	<b>20</b>
<b>1.3.3. Μενού Επιλογής Βασικών Μαθηματικών Εντολών .....</b>	<b>27</b>
<b>1.4 Γρήγορος Οδηγός Δραστηριοτήτων .....</b>	<b>30</b>
<b>1.4.1 Παραγωγή Κώδικα Κυκλώματος .....</b>	<b>32</b>
<b>1.4.2 Φόρτωση κώδικα στον ελεγκτή ARD:icon .....</b>	<b>33</b>
<b>1.4.3 Πίνακας Ελέγχου Εφαρμογής ARD:icon.....</b>	<b>34</b>
<b>1.4.4 Προσομοίωση Εφαρμογής ARD:icon .....</b>	<b>35</b>
<b>1.5 Αποθήκευση και Επαναφορά Ρουτίνας.....</b>	<b>40</b>

## 1.1 Εισαγωγή στο ARD:icon

Το πρόγραμμα ARD:icon, αποτελεί μια εύχρηστη εφαρμογή προγραμματισμού για ελεγκτές Arduino.

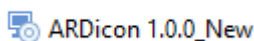
- Διαθέτει ένα διαδραστικό περιβάλλον κωδικοποίησης, που βασίζεται σε εικονίδια και μπλοκ εντολών, που κάνει τον προγραμματισμό εύκολο σαν παιχνίδι έτσι, ώστε ακόμα και τα μικρά παιδιά να μπορέσουν να κατανοήσουν εύκολα τη λογική των υπολογιστών.
- Διαθέτει βιβλιοθήκη συσκευών εισόδου και εξόδου, που βασίζονται σε εικονίδια (έγχρωμες λυχνίες LED, αισθητήρες, κινητήρες, πολλοί τύποι ηλεκτρονικών διακοπών, ροοστάτες κ.λπ.)
- Το σχήμα συνάρτησης-λογικών μπλοκ προσφέρει έναν μοναδικό, ευφυή και απλοποιημένο τρόπο εκπαίδευσης στον προγραμματισμό και την επεξεργασία διαφόρων εισόδων, την εφαρμογή θεμελιωδών εννοιών προγραμματισμού, όπως επανάληψη, δηλώσεις υπό όρους και μεταβλητές, προκειμένου να δημιουργηθούν τα επιθυμητά αποτελέσματα.

Το ARD:icon, είναι μια πλατφόρμα προγραμματισμού, που χρησιμοποιεί πλακίδια προγραμματισμού και εικονίδια δομικών στοιχείων, για να δημιουργήσει μια εύκολη εμπειρία “plug and play” για παιδιά.

### 1.1.1 Εγκατάσταση της εφαρμογής ARD:icon

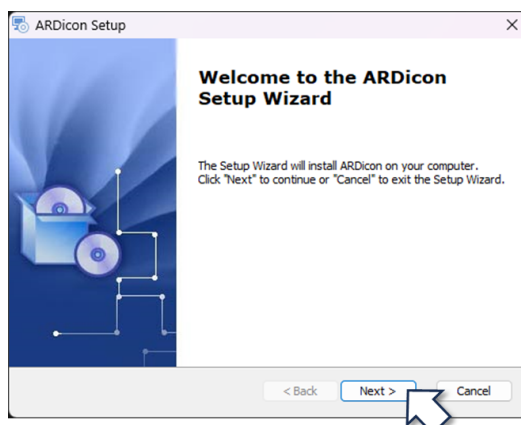
Για να ξεκινήσετε να προγραμματίζετε τις εργασίες με την εφαρμογή ARD:icon, το μόνο που έχετε να κάνετε είναι απλά να εγκαταστήσετε την εφαρμογή στον υπολογιστή σας και να τρέξετε την εφαρμογή.

Για να ξεκινήσει η εγκατάσταση της εφαρμογής, κάνετε διπλό κλικ πάνω στο εικονίδιο

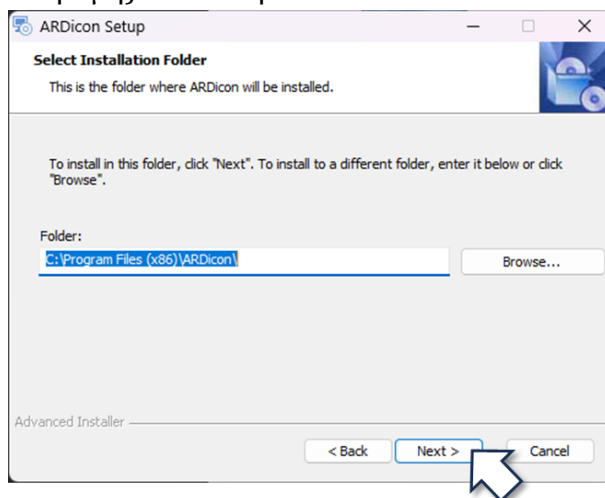


**Σημείωση:** Σε περίπτωση που έχετε ενεργοποιημένο το anti-virus, μπορεί να αναγνωρίσει το λογισμικό ARD:icon ως απειλή. Σε αυτή την περίπτωση δημιουργήστε μία εξαίρεση στο anti-virus ή απενεργοποιήστε το προσωρινά.

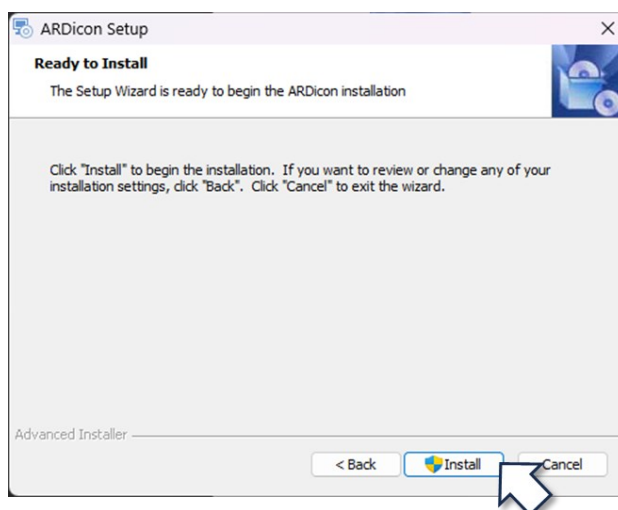
Στο αναδυόμενο παράθυρο, επιλέξτε “Next”



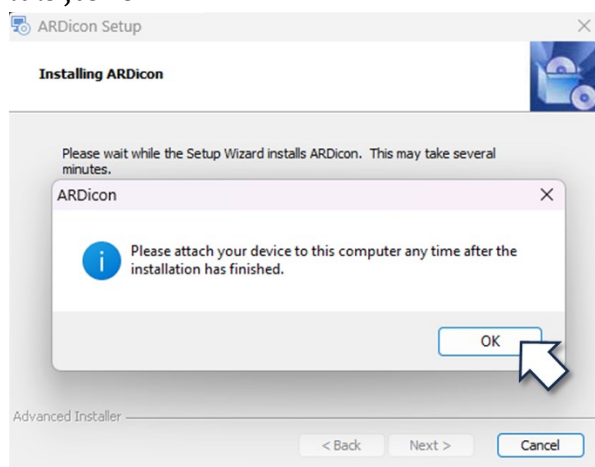
Επιλέξτε έναν φάκελο προορισμού, όπου θέλετε να γίνει η εγκατάσταση της εφαρμογής (ή αφήστε την υπάρχουσα διαδρομή) και πατήστε “Next”.



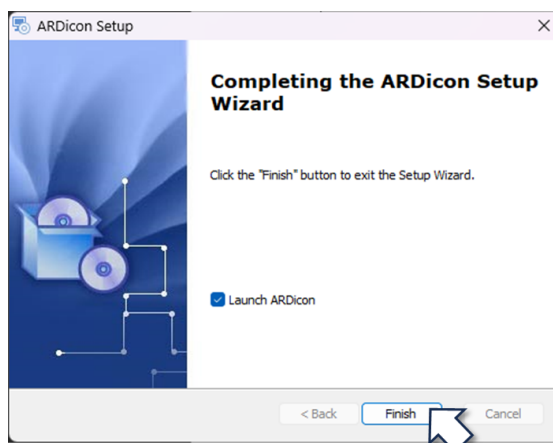
Κάνοντας αριστερό κλικ στο κουμπί “Install” θα ξεκινήσει η διαδικασία της εγκατάστασης.



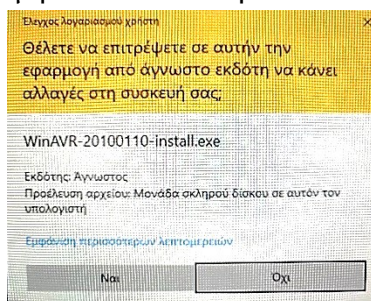
Στο επόμενο παράθυρο επιλέξτε “ok”.



Μόλις ολοκληρωθεί η εγκατάσταση, πατήστε “Finish”.

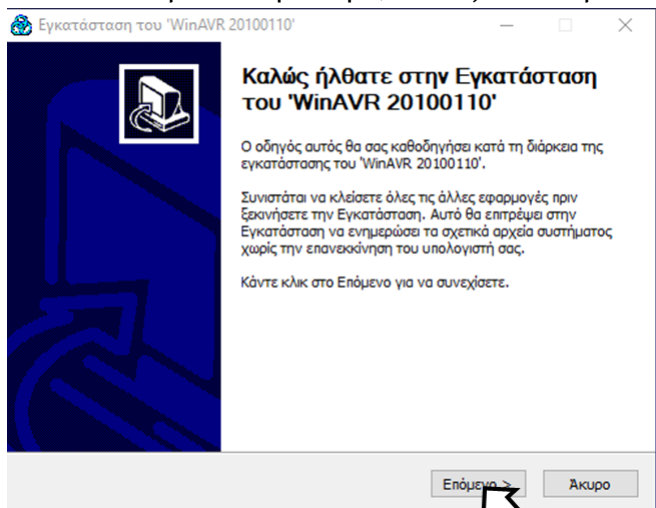


Όταν η εγκατάσταση του λογισμικού ολοκληρωθεί, στην επιφάνεια του υπολογιστή θα εμφανιστεί το παρακάτω παράθυρο.

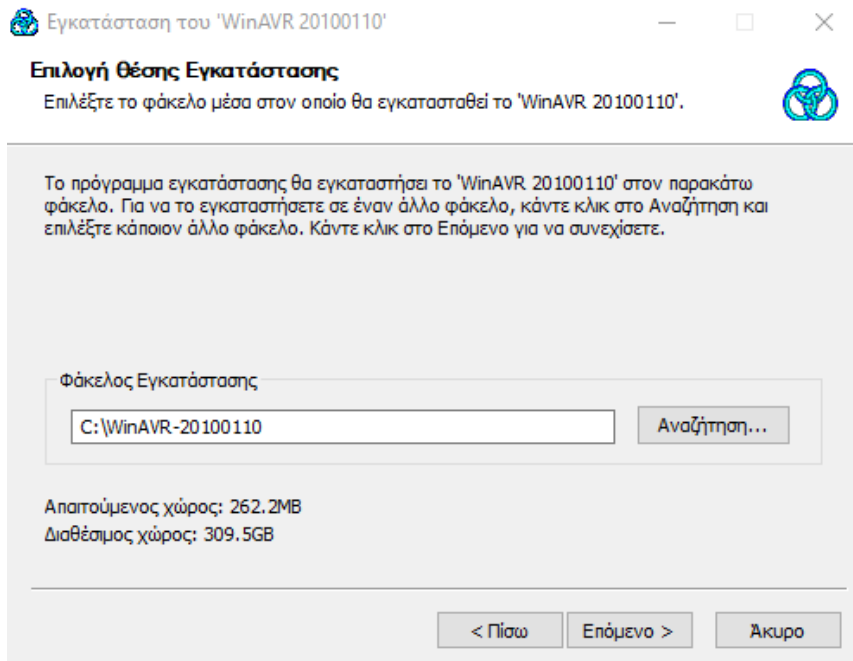


Επιλέξτε «**ΝΑΙ**» και προχωρήστε στην εγκατάσταση του λογισμικού WinAVR.

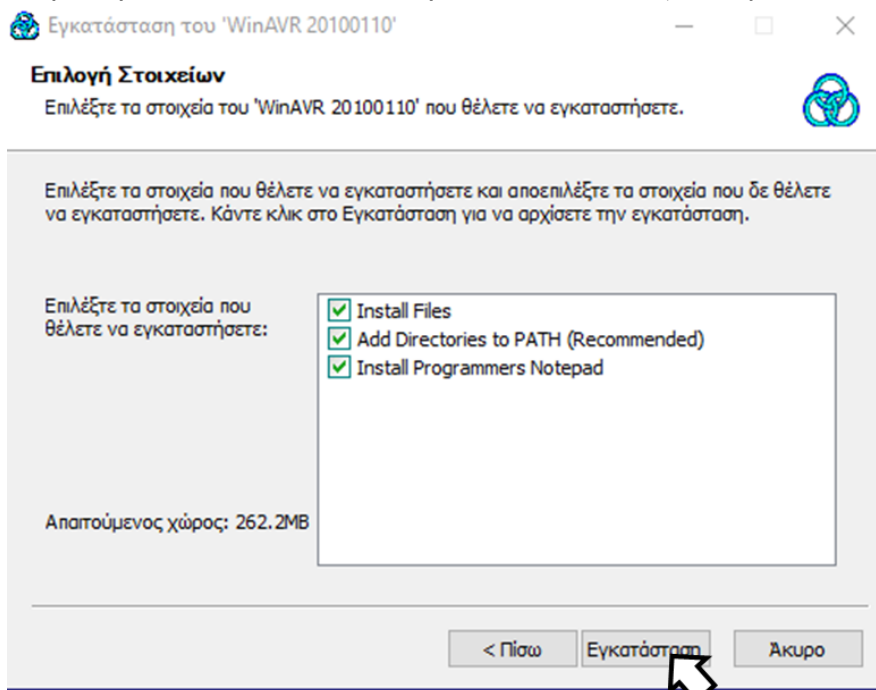
Στο αναδυόμενο παράθυρο, επιλέξτε “Επόμενο”.



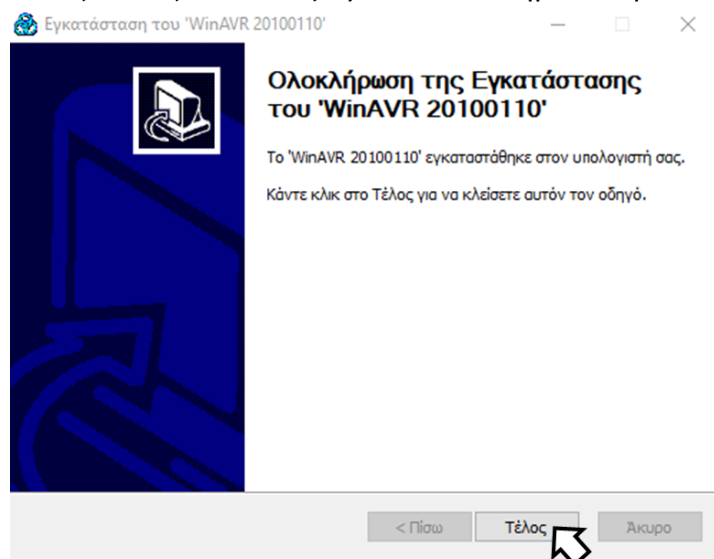
Στο επόμενο παράθυρο επιλέξτε έναν φάκελο προορισμού, όπου θέλετε να γίνει η εγκατάσταση της εφαρμογής επιλέγοντας αναζήτηση (ή αφήστε την υπάρχουσα διαδρομή) και πατήστε **“Επόμενο”**.



Στο επόμενο παράθυρο επιλέξτε τα στοιχεία που θέλετε να εγκαταστήσετε όπως φαίνεται στην παρακάτω εικόνα και στην συνέχεια επιλέξτε **«Εγκατάσταση»**.




Τέλος επιλέξτε «Τέλος» για να ολοκληρωθεί η διαδικασία.



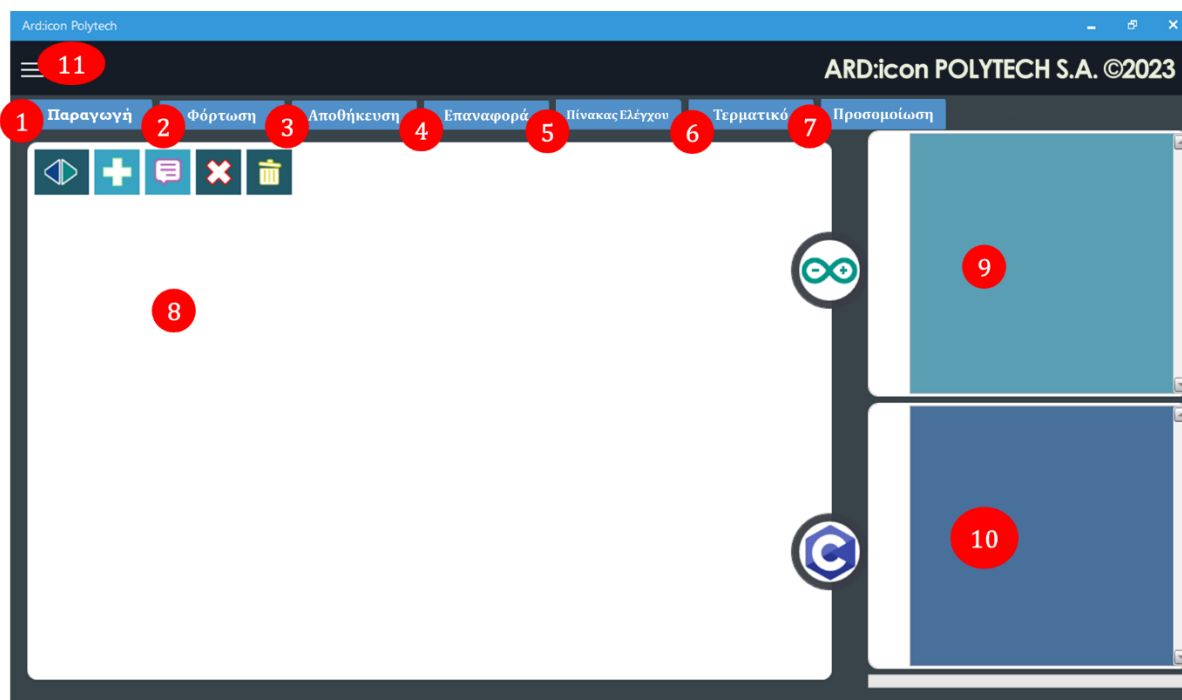
Αφού τελειώσετε και την διαδικασία εγκατάστασης του winAVR (κλείστε την σελίδα του winAVR που ανοίγει στο διαδίκτυο) τότε στην επιφάνεια του υπολογιστή, αυτόματα θα δημιουργηθεί το εικονίδιο συντόμευσης της εφαρμογής . Με διπλό κλικ πάνω στο εικονίδιο, μπορείτε να ανοίξετε την εφαρμογή "ARD:icon".

### 1.1.2 Εφαρμογή ARD:icon

Ανοίξτε την εφαρμογή, επιλέγοντας το εικονίδιο της εφαρμογής  . Στην επόμενη οθόνη, επιλέξτε Start.



Η αρχική οθόνη της εφαρμογής θα εμφανιστεί στην οθόνη του υπολογιστή σας.



## Μενού Επιλογών

### 1. Παραγωγή

Με το κουμπί Παραγωγή, μπορείτε να παράγετε τον κώδικα που προγραμματίσατε σε Κώδικα Arduino και κώδικα C++.

### 2. Φόρτωση

Με το κουμπί Φόρτωση, μπορείτε να φορτώσετε τον κώδικα στη συσκευή Arduino.

### 3. Αποθήκευση

Με το κουμπί Αποθήκευση, μπορείτε να αποθηκεύσετε τον κώδικα (ρουτίνα) που προγραμματίσατε, προκειμένου να τον χρησιμοποιήσετε σε κάποια άλλη εργασία.

### 4. Επαναφορά

Με το κουμπί Επαναφορά, μπορείτε να επαναφέρετε κάποια από τις ρουτίνες που έχετε αποθηκεύσει.

### 5. Πίνακας Ελέγχου

Με το κουμπί Πίνακας Ελέγχου, μπορείτε να εισέλθετε στην οθόνη ελέγχου των Συσκευών που έχετε προγραμματίσει.

### 6. Τερματικό

Με το κουμπί Τερματικό, μπορείτε να ανοίξετε την οθόνη του τερματικού, όπου τυπώνονται οι τιμές των μεταβλητών που εσείς προγραμματίζετε (εντολή "[serialprint](#)"). Στην ουσία είναι ένας εικονικός εκτυπωτής.

### 7. Προσομοίωση


Με το κουμπί Προσομοίωση, μπορείτε να ανοίξετε την οθόνη προσομοίωσης του κώδικα που προγραμματίσατε.




## 8. Οθόνη προγραμματισμού


Σε αυτή την οθόνη ξεκινάτε να προγραμματίζετε τις εργασίες σας. Όπως βλέπετε πάνω στην περιοχή προγραμματισμού, υπάρχει μια σειρά από κουμπιά. Ας δούμε τη λειτουργία καθενός από αυτά ξεχωριστά:



A. Το πλήκτρο  ορίζει το επίπεδο του κώδικα, που χρησιμοποιείται για ένθετες συναρτήσεις και υπορουτίνες. Διαθέτει 3 επίπεδα. Πιέστε το κουμπί, για να μεταφερθείτε στην γραμμή 2 και 3 αλλά και για να επιστρέψετε στην γραμμή 1. Πάντα ξεκινάμε μία καινούργια ρουτίνα ή υπορουτίνα από το επίπεδο (γραμμή) 1.

B. Το πλήκτρο  σας εισάγει στο περιβάλλον προγραμματισμού, για το επίπεδο που έχετε επιλέξει.

Γ. Το πλήκτρο  σας δίνει τη δυνατότητα να προσθέσετε κάποιο σχόλιο

Δ. Το πλήκτρο  διαγράφει τις τελευταίες εντολές που έχετε επιλέξει


E. Το πλήκτρο  διαγράφει όλες τις εντολές που έχετε επιλέξει

9. Οθόνη εμφάνισης κώδικα, που δημιουργείται αυτόματα από το Arduino/**Arduino Code**

10. Οθόνη εμφάνισης κώδικα C, που δημιουργείται αυτόματα/ C code

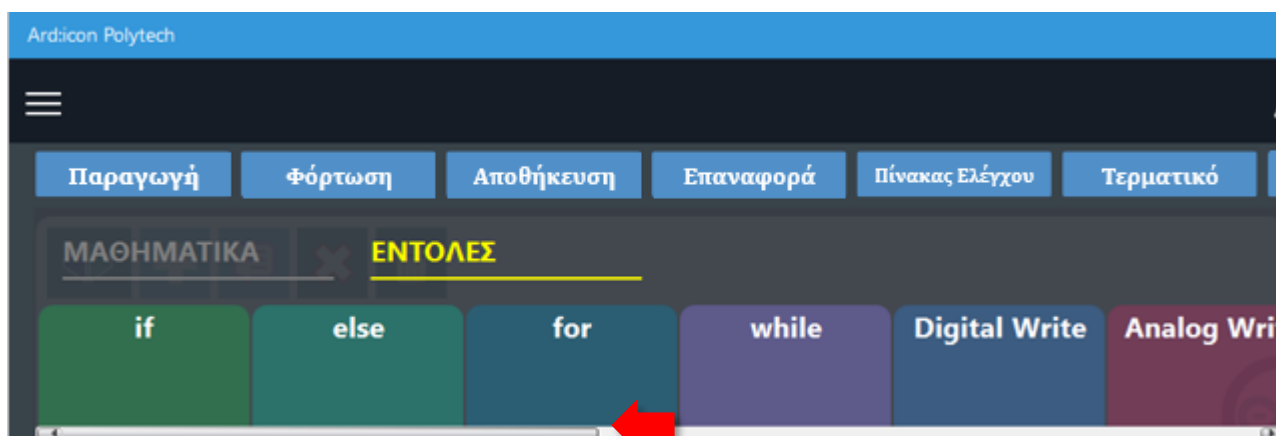
11. Μενού επιπλέον επιλογών. Πατώντας πάνω στο εικονίδιο, θα δείτε τους συνδέσμους, όπου μπορείτε να κατεβάσετε τα εγχειρίδια λειτουργίας των λογισμικών, καθώς και τον Γρήγορο Οδηγό Δραστηριοτήτων.

Για να ξεκινήσετε να προγραμματίζετε ένα έργο, πρώτα επιλέγετε το επίπεδο του κώδικα

χρησιμοποιώντας το κουμπί  στην περιοχή προγραμματισμού. Πάντα ξεκινάτε μια νέα άσκηση από το επίπεδο ένα.

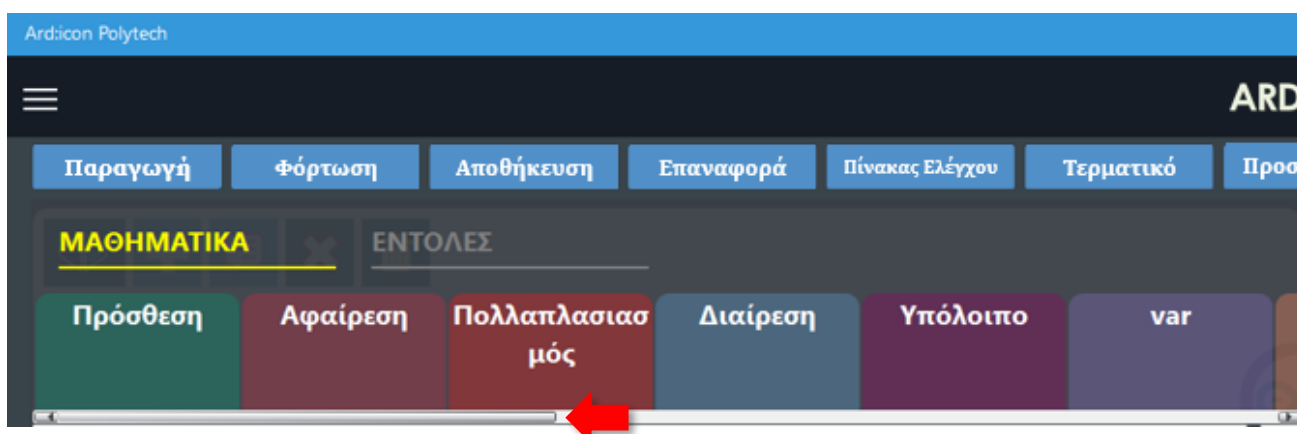
Στη συνέχεια, πατώντας το κουμπί , εισέρχεστε στο περιβάλλον των εντολών. Εμφανίζονται οι εξής επιλογές.

## A. Μενού επιλογών Εντολών Λειτουργίας



Μπορείτε να μεταβείτε στις υπόλοιπες εντολές, μετακινώντας την μπάρα στο κάτω μέρος

## B. Μενού επιλογών βασικών μαθηματικών εντολών



Μπορείτε να μεταβείτε στις υπόλοιπες εντολές, μετακινώντας την μπάρα στο κάτω μέρος

## 1.2 Σύνδεση Ελεγκτή και Περιφερειακών μονάδων

Για να ξεκινήσετε να προγραμματίζετε με την εφαρμογή ARD:icon θα χρειαστείτε τον ελεγκτή ARD:icon τον οποίο πρέπει να συνδέσετε με τον υπολογιστή σας χρησιμοποιώντας το καλώδιο USB.



Συνδέστε την άκρη USB Type A στον υπολογιστή σας και στη συνέχεια την άκρη

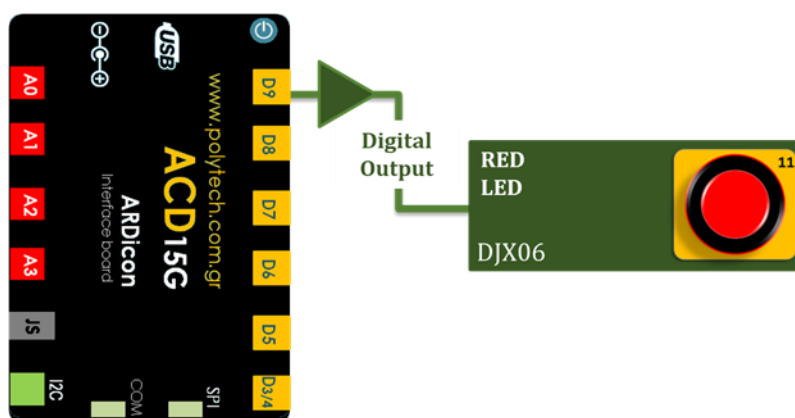


USB Type B στη θύρα USB του ελεγκτή ARD:icon.

Για να συνδέσετε τις περιφερειακές συσκευές (αισθητήρες, ενεργοποιητές και εξόδους) χρησιμοποιήστε τα καλώδια UTP.

Συνδέστε το ένα άκρο του καλωδίου στη συσκευή και το άλλο άκρο στην ανάλογη θύρα του ελεγκτή σύμφωνα με τις οδηγίες που θα βρείτε σε κάθε δραστηριότητα. Όπως το παρακάτω παράδειγμα:

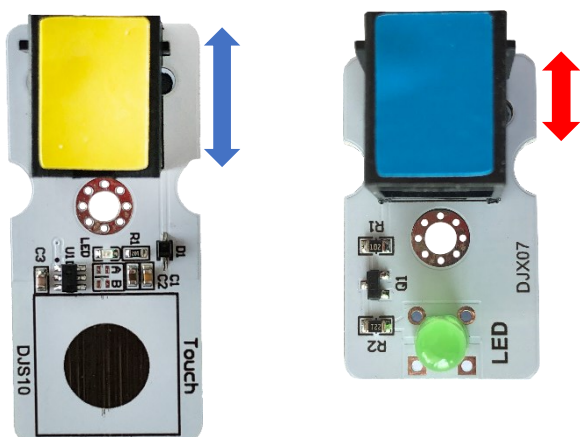
- Κόκκινο LED (DJX06) στη θύρα D9



### 1.2.1 Σωστή συναρμολόγηση των περιφερειακών συσκευών με τα τουβλάκια

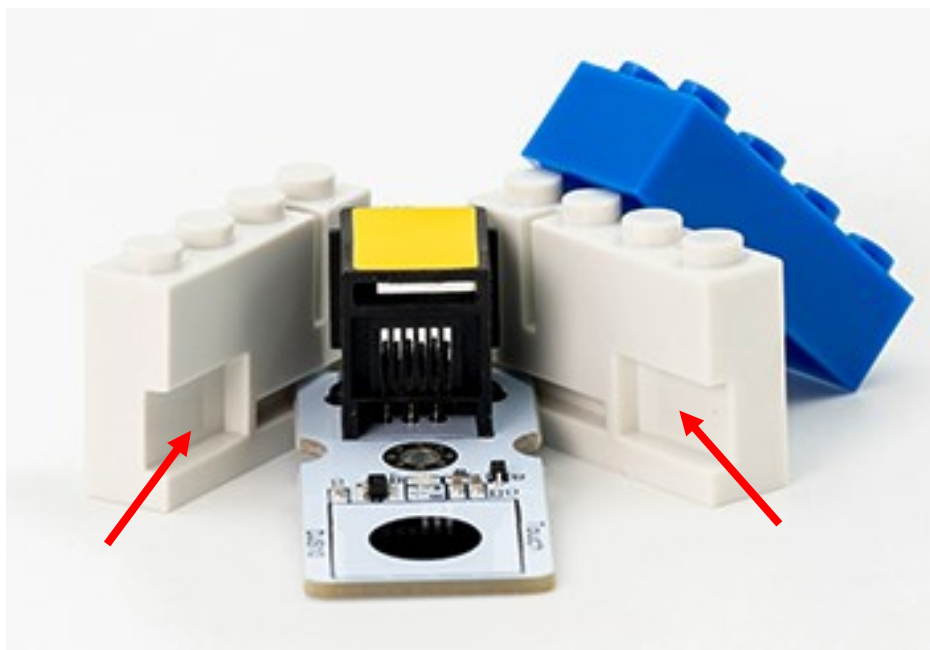
Τα τουβλάκια που συνοδεύουν τις περιφερειακές μονάδες εισόδου και εξόδου μπορούν να συναρμολογηθούν με δύο διαφορετικούς τρόπους ανάλογα με το μέγεθος της πλακέτας στη πλευρά σύνδεσης του καλωδίου RJ11.

Πιο συγκεκριμένα όπως μπορείτε να δείτε στις παρακάτω εικόνες, οι πλακέτες του αισθητήρα αφής και του πράσινου LED διαφέρουν σε μέγεθος. Στον αισθητήρα αφής η πλευρά της σύνδεσης του καλωδίου (**μπλε βέλος**) είναι μεγαλύτερη από αυτή του LED (**κόκκινο βέλος**).

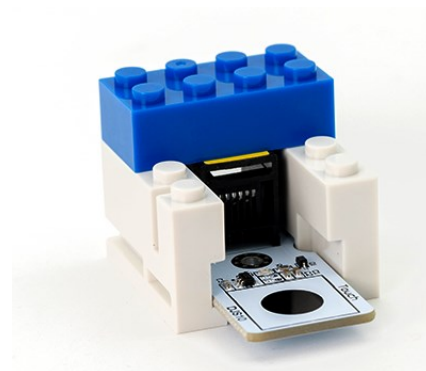


#### A. Συσκευή με μέγεθος όπως ο αισθητήρας αφής.

Αρχικά, τοποθετείστε τη συσκευή στα πλαϊνά τουβλάκια. Οι εγκοπές στα τουβλάκια πρέπει να είναι όπως δείχνει η παρακάτω εικόνα, για να μπορέσει να τοποθετηθεί σωστά η συσκευή.

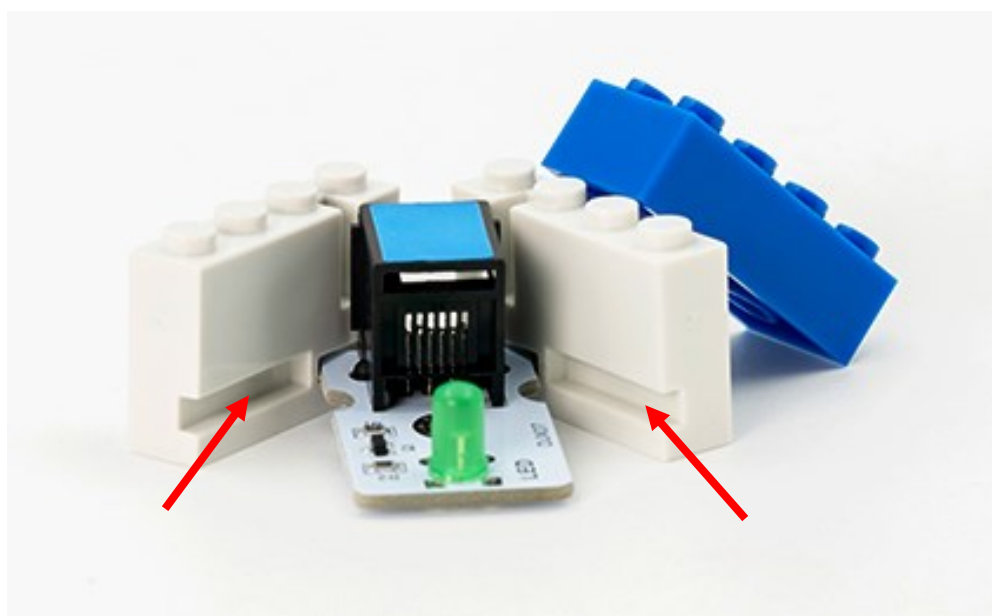


Τέλος, τοποθετείστε το μπλε τουβλάκι στο πάνω μέρος.

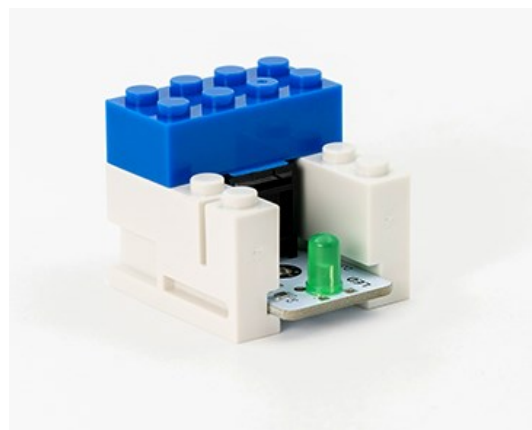


### **Β. Συσκευή με μέγεθος όπως το πράσινο LED.**

Αρχικά, τοποθετείστε τη συσκευή στα πλαϊνά τουβλάκια. Οι εγκοπές στα τουβλάκια πρέπει να είναι όπως δείχνει η παρακάτω εικόνα, για να μπορέσει να τοποθετηθεί σωστά η συσκευή.

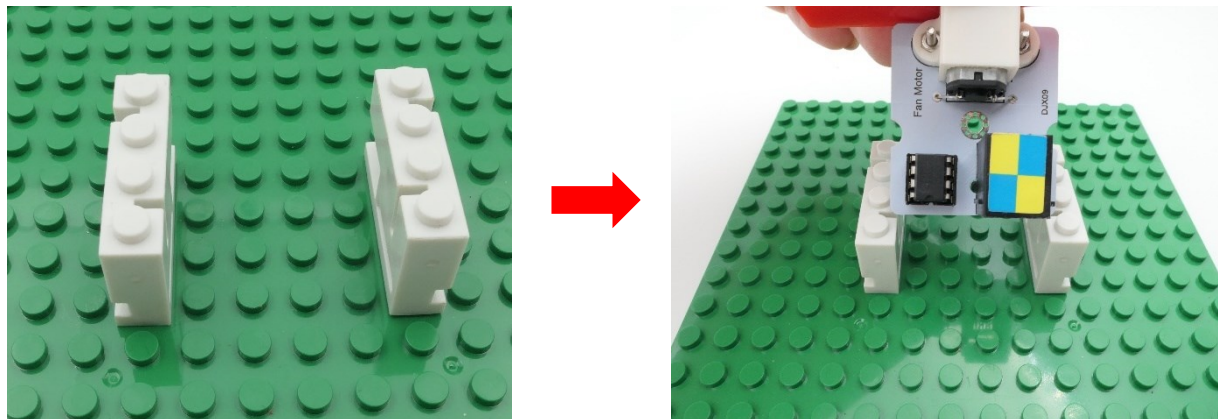


Τέλος, τοποθετείστε το μπλε τουβλάκι στο πάνω μέρος.

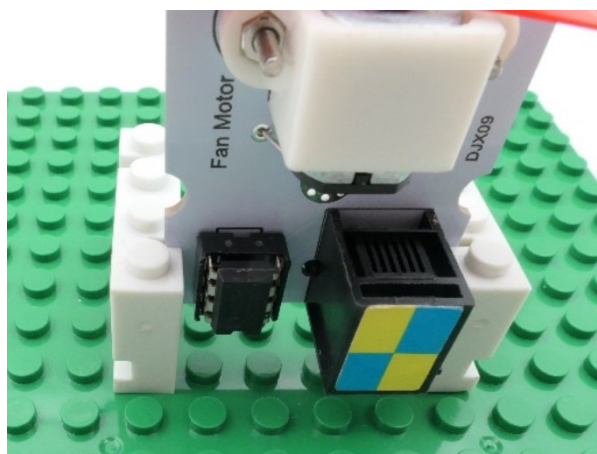


Κάποιες από τις συσκευές έχουν αρκετά μεγαλύτερη πλακέτα σε πλάτος και δεν μπορούν να στερεωθούν με αυτόν το τρόπο στα τουβλάκια. Δείτε παρακάτω στις εικόνες τον τρόπο που μπορείτε να στερεώσετε αυτές τις συσκευές στα τουβλάκια.

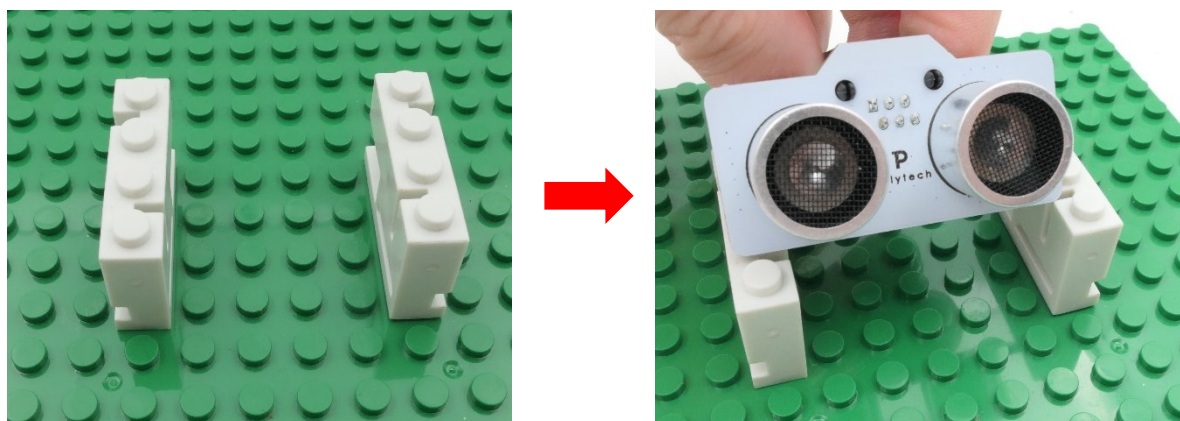
### 1. Μονάδα Step Motor με Προπέλα – DJX09



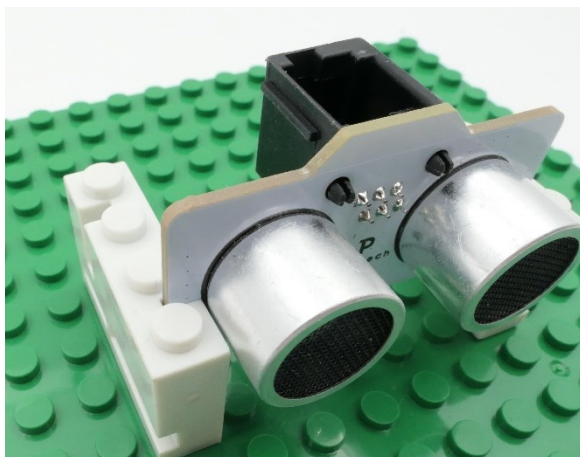
Στερεώστε τη συσκευή στις εγκοπές με το μικρότερο άνοιγμα.



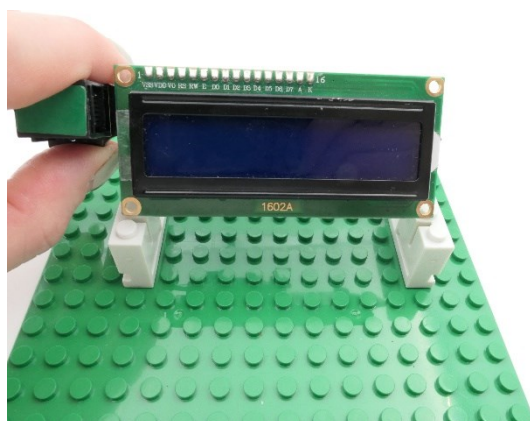
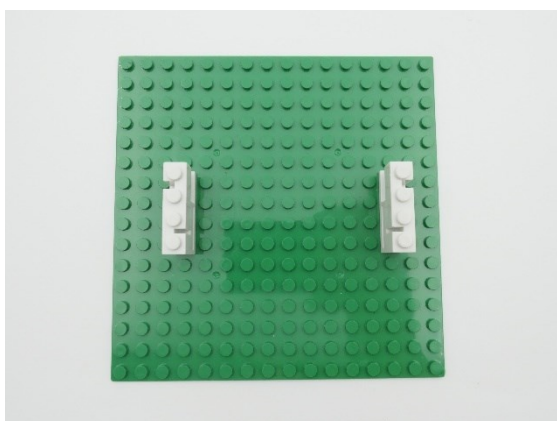
### 2. Μονάδα Αισθητήρα Απόστασης Υπερήχων – DJS22



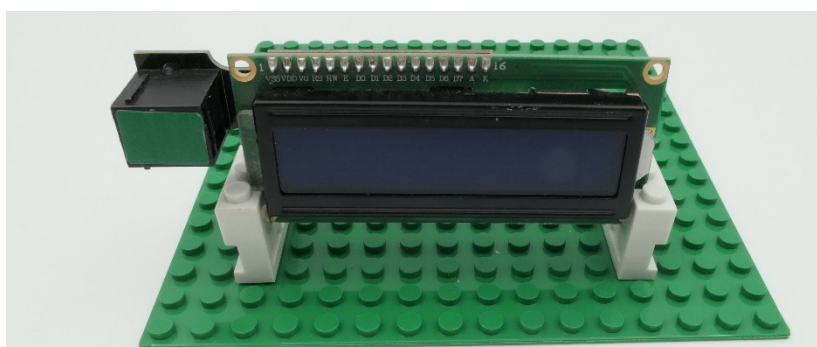
Στερεώστε τη συσκευή στις εγκοπές με το μικρότερο άνοιγμα.



### 3. Μονάδα Οθόνης LCD - AJX04



Στερεώστε τη συσκευή στις εγκοπές με το μικρότερο άνοιγμα.




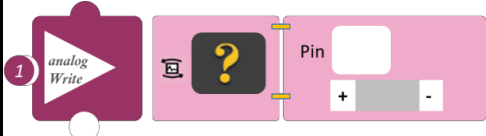



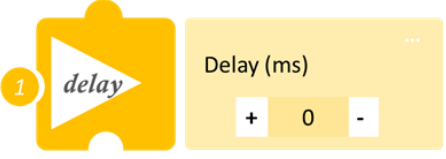

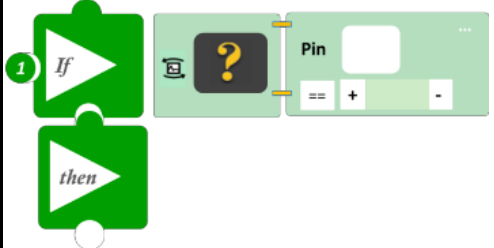
### 1.3 Εντολές Προγραμματισμού

Το μενού επιλογών της εφαρμογής αποτελείται από το:


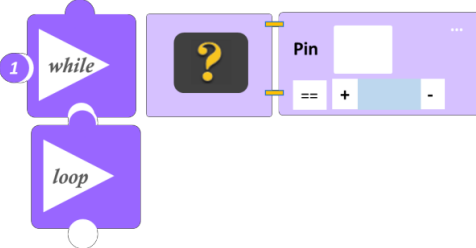
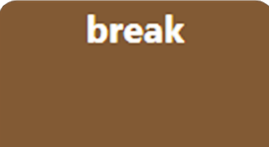

1. Μενού εντολών λειτουργίας. Την ανάλυση των εντολών του συγκεκριμένου μενού μπορείτε να τη δείτε στις παρακάτω ενότητες.
2. Μενού μαθηματικών εντολών. Την ανάλυση των εντολών του συγκεκριμένου μενού μπορείτε να τη δείτε στις παρακάτω ενότητες.
3. Μενού Εντολών των Ρομπότ R2 και R4. Την ανάλυση των εντολών του συγκεκριμένου μενού στα εγχειρίδια λειτουργίας των ρομπότ.



## 1.3.1. Μενού Επιλογής Εντολών Λειτουργίας

Εικονίδιο Επιλογής Εντολής	Περιγραφή	Πλακίδιο Εντολής
	<b>Analog Read:</b> Αυτή η εντολή είναι ενσωματωμένη στις εντολές "if", "while" και "variable".	
	<b>Analog Write:</b> Αυτή η εντολή προγραμματίζει τις παραμέτρους της <b>αναλογικής συσκευής εξόδου</b> που συνδέετε στον ελεγκτή ARD:icon.	 <p>Δείτε το σχετικό βίντεο <a href="#">2 ENTOLH Analog Write</a> με τις οδηγίες προγραμματισμού της εντολής</p>
	<b>Digital Read:</b> Αυτή η εντολή είναι ενσωματωμένη στις εντολές "if", "while" και "variable".	
	<b>Digital Write:</b> Αυτή η εντολή προγραμματίζει τις παραμέτρους της <b>ψηφιακής συσκευής εξόδου</b> που συνδέετε στον ελεγκτή ARD:icon.	 <p>Δείτε το σχετικό βίντεο <a href="#">1 ENTOLH Digital Write</a> με τις οδηγίες προγραμματισμού της εντολής</p>
	<b>delay:</b> Επιλέγουμε την παύση του προγράμματος για συγκεκριμένο αριθμό χιλιοστών του δευτερολέπτου.	 <p>Δείτε το σχετικό βίντεο <a href="#">3 ENTOLH Delay</a> με τις οδηγίες προγραμματισμού της εντολής</p>
	<b>if..then:</b> Επιλέγουμε τη συνθήκη if. Αυτή η εντολή προγραμματίζει τις παραμέτρους της συνθήκης. Εάν η συνθήκη είναι αληθής, τότε θα εκτελεστεί η εντολή.	 <p>Δείτε τα 2 σχετικά βίντεο, <a href="#">4 ENTOLH if then digital</a> και <a href="#">5 ENTOLH if then Analog</a> με τις οδηγίες προγραμματισμού της εντολής</p>

	<p><b>else:</b> Επιλέγουμε την συνθήκη else. Συνδέεται ΠΑΝΤΑ με την δήλωση if. Αν η δήλωση if είναι ψευδής, τότε θα εκτελεστεί η εντολή else.</p> <p>Η εντολή else πάντα συνοδεύεται από τουλάχιστον μία εντολή προγραμματισμού αναλογικής (<b>analog write</b>) ή ψηφιακής εξόδου(<b>digital write</b>)</p>	 <p>Δείτε το σχετικό βίντεο <a href="#">6 ENTOLH else</a> με τις οδηγίες προγραμματισμού της εντολής</p>
	<p><b>Serial print:</b> Επιλέγουμε τις μεταβλητές που θέλουμε να τυπώσει ο ελεγκτής. Μπορεί να είναι οι τιμές κάποιας αναλογικής ή ψηφιακής συσκευής εισόδου ή κάποια συγκεκριμένη φράση ή αριθμός.</p>	 <p>Δείτε τα 2 σχετικά βίντεο, <a href="#">7 ENTOLH Serial Print</a> και <a href="#">M2A ENTOLH Serial Print</a> με τις οδηγίες προγραμματισμού της εντολής</p>
	<p><b>for:</b> Επιλέγουμε τη συνθήκη βρόχου “for”, για να προγραμματίσουμε πόσες φορές κάτι πρόκειται να επαναληφθεί για μια δεδομένη συνθήκη if. Η επανάληψη θα εκτελείται, έως ότου η δεδομένη συνθήκη σταματήσει να ισχύει.</p> <p>Η εντολή “for” πάντα συνοδεύεται τουλάχιστον από μία εντολή Προγραμματισμού αναλογικής (<b>analog write</b>) ή ψηφιακής εξόδου(<b>digital write</b>).</p>	 <p>Δείτε το σχετικό βίντεο <a href="#">10 ENTOLH for</a> με τις οδηγίες προγραμματισμού της εντολής</p>
	<p><b>LCD Οθόνη:</b> Επιλέγουμε τις μεταβλητές που θέλουμε να τυπώνονται στη σειρά 1 και 2 της οθόνης LCD.</p>	 <p>Δείτε το σχετικό βίντεο <a href="#">12 ENTOLH LCD Screen</a> με τις οδηγίες προγραμματισμού της εντολής</p>
	<p><b>RGB LED:</b> Επιλέγουμε το χρώμα που θέλουμε να ανάβει το κάθε LED της συσκευής RGB LED.</p>	 <p>Δείτε το σχετικό βίντεο <a href="#">11 ENTOLH RGB LED</a> με τις οδηγίες προγραμματισμού της εντολής</p>

Εικονίδιο Επιλογής Εντολής	Περιγραφή	Πλακίδιο Εντολής
	<p><b>while:</b> Επιλέγουμε τη συνθήκη βρόχου “while”, για να προγραμματίσουμε πόσες φορές κάτι πρόκειται να επαναληφθεί, έως ότου μια δεδομένη συνθήκη σταματήσει να ισχύει και ενώ εκτελείται μια άλλη συνθήκη.</p> <p>Η σύνταξη της εντολής “while” βασίζεται σε μια συνθήκη, παρόμοια με την εντολή «if». Η διαφορά τους είναι ότι η «if» εκτελεί μόνο μια φορά το τμήμα Κώδικα που την αφορά, ενώ η «while» για όσο είναι αληθής η συνθήκη της. Πάντα συνοδεύεται τουλάχιστον από μία εντολή προγραμματισμού αναλογικής (<a href="#">analog write</a>) ή ψηφιακής εξόδου(<a href="#">digital write</a>).</p>	 <p>Δείτε το σχετικό βίντεο <a href="#">13 ENTOLH while</a> με τις οδηγίες προγραμματισμού της εντολής</p>
	<p><b>break:</b> Επιλέγουμε την εντολή break για να τερματίσουμε την if, το βρόχο της while, και τέλος και της for.</p>	

## 1.2.2 Ανάλυση Εντολών Λειτουργίας

### 1. Η εντολή "digital write":

αναφέρεται σε έναν όρο, που χρησιμοποιείται σε προγραμματιστικά περιβάλλοντα, για τον έλεγχο των ψηφιακών ακροδεκτών/θυρών ενός μικροελεγκτή. Συγκεκριμένα, αναφέρεται σε μια εντολή που γράφει (ή θέτει) μια τιμή σε έναν ψηφιακό ακροδέκτη/θύρα.

Οι ψηφιακοί ακροδέκτες μπορούν να έχουν δύο πιθανές καταστάσεις: "HIGH" (υψηλό) και "LOW" (χαμηλό). Η εντολή "digital write", επιτρέπει στον προγραμματιστή να ορίσει την κατάσταση του ψηφιακού ακροδέκτη σε ένα από αυτά τα δύο σήματα.

Όταν προγραμματίζουμε μια ψηφιακή συσκευή, χρησιμοποιώντας την εντολή "digital write", συνήθως αναφερόμαστε στον έλεγχο ενός ψηφιακού ακροδέκτη για την ενεργοποίηση ή απενεργοποίηση μιας συσκευής ή ενός περιφερειακού, όπως ένα LED ή ένας κινητήρας.

### 2. Η εντολή "analog write":

χρησιμοποιείται συνήθως σε πλατφόρμες προγραμματισμού, όπως η Arduino, για να ελέγξουμε την ένταση ενός αναλογικού ακροδέκτη/θύρα. Αν και η ονομασία "analog write" μπορεί να είναι παραπλανητική, δεν πρόκειται για πραγματική αναλογική έξοδο, αλλά για μια μορφή παλμοταχύτητας (pulse-width modulation - PWM).

Η τεχνική της PWM σας επιτρέπει να προσομοιώσετε μια αναλογική έξοδο, ρυθμίζοντας την αναλογία μεταξύ της περιόδου, που η έξοδος είναι σε υψηλή κατάσταση (HIGH) και της περιόδου, που είναι σε χαμηλή κατάσταση (LOW). Αυτή η αλλαγή της αναλογίας επιτρέπει τον έλεγχο της φωτεινότητας LED, της ταχύτητας κινητήρων ή άλλων περιφερειακών, που ανταποκρίνονται σε αναλογικές τιμές.

### 3. Η εντολή "if" (ή "if statement"):

αναφέρεται σε έναν έλεγχο υπό συνθήκη, σε ένα πρόγραμμα. Η εντολή "if", χρησιμοποιείται για να εκτελέσει μια συγκεκριμένη ενέργεια ή ένα κομμάτι κώδικα, αν η συνθήκη που δίνεται είναι αληθής (true).

Η γενική μορφή της εντολής "if" είναι η εξής:

```
if (συνθήκη) {  
    // Κώδικας που εκτελείται, αν η συνθήκη είναι αληθής  
}
```

Η "συνθήκη", είναι μια λογική έκφραση, που αξιολογείται ως αληθής ή ψευδής. Αν η συνθήκη είναι αληθής, τότε ο κώδικας μέσα στις αγκύλες {} εκτελείται. Αν η συνθήκη είναι ψευδής, τότε ο κώδικας μέσα στις αγκύλες παραλείπεται και η εκτέλεση συνεχίζεται μετά το τμήμα "if".

### 4. Η εντολή "else":

χρησιμοποιείται σε συνδυασμό με την εντολή "if", σε ένα πρόγραμμα για την εκτέλεση ενός διαφορετικού κομματιού κώδικα, όταν η συνθήκη της εντολής "if" είναι ψευδής (false).

Αφού εκτελεστεί ο κώδικας, που βρίσκεται μέσα στο τμήμα "if", όταν η συνθήκη είναι αληθής, μπορείτε να προσθέσετε το τμήμα "else", για να καθορίσετε τι θα γίνει, αν η συνθήκη είναι ψευδής. Ο κώδικας μέσα στο τμήμα "else" θα εκτελεστεί, αν η συνθήκη της εντολής "if" δεν είναι αληθής.

Η γενική μορφή της εντολής "if-else" είναι η εξής:

```
if (συνθήκη) {  
    // Κώδικας που εκτελείται, αν η συνθήκη είναι αληθής  
} else {  
    // Κώδικας που εκτελείται, αν η συνθήκη είναι ψευδής  
}
```

Αν η συνθήκη της εντολής "if" είναι αληθής, τότε ο κώδικας μέσα στο πρώτο μπλοκ εκτελείται. Αν η συνθήκη είναι ψευδής, τότε ο κώδικας μέσα στο μπλοκ "else" εκτελείται αντί αυτού.

### 5. Η εντολή "for":

χρησιμοποιείται σε πολλές γλώσσες προγραμματισμού, για την επανάληψη ενός τμήματος κώδικα για έναν συγκεκριμένο αριθμό επαναλήψεων.

Η γενική μορφή της εντολής "for" είναι η εξής:

```
for (αρχικοποίηση; συνθήκη; ενημέρωση) {  
    // Κώδικας που εκτελείται κατά τη διάρκεια της επανάληψης  
}
```

Στην αρχή της εντολής "for", το τμήμα "αρχικοποίηση" χρησιμοποιείται για να αρχικοποιήσετε μια μεταβλητή ελέγχου ή να εκτελέσετε οποιοσδήποτε άλλες εντολές, που χρειάζονται πριν από την έναρξη της επανάληψης.

Η "συνθήκη", είναι μια λογική έκφραση που ελέγχεται σε κάθε επανάληψη. Αν η συνθήκη είναι αληθής, ο κώδικας μέσα στις αγκύλες εκτελείται. Αν η συνθήκη είναι ψευδής, η εκτέλεση της επανάληψης σταματά και η εκτέλεση του προγράμματος συνεχίζεται από το επόμενο σημείο μετά την εντολή "for".

Η "ενημέρωση", είναι μια εντολή, που εκτελείται μετά από κάθε επανάληψη και χρησιμοποιείται για να αλλάξετε την κατάσταση της μεταβλητής ελέγχου ή να εκτελέσετε άλλες ενέργειες, που απαιτούνται στο τέλος κάθε επανάληψης.

Ας δούμε ένα παράδειγμα: Έστω, ότι θέλουμε να εκτυπώσουμε τους αριθμούς από 1 έως 5. Μπορούμε να το πετύχουμε με την εξής χρήση της εντολής "for" σε γλώσσα όπως η C++:

```
for (int i = 1; i <= 5; i++) {  
    cout << i << endl;  
}
```

Σε αυτό το παράδειγμα, η μεταβλητή "i" αρχικοποιείται στην τιμή 1. Η συνθήκη ελέγχει αν η "i" είναι μικρότερη ή ίση με 5. Καθώς αυξάνουμε την "i" κατά 1, μετά από κάθε επανάληψη (i++), ο κώδικας μέσα στις αγκύλες εκτελείται και εκτυπώνει την τρέχουσα τιμή της "i". Η διαδικασία επαναλαμβάνεται, μέχρι η συνθήκη να γίνει ψευδής, δηλαδή όταν η "i" γίνει μεγαλύτερη από 5.

### 6. Η εντολή "while":

χρησιμοποιείται σε πολλές γλώσσες προγραμματισμού, για την επανάληψη ενός τμήματος κώδικα, μέχρις ότου μια συνθήκη γίνει ψευδής (false). Η συνθήκη ελέγχεται στην αρχή κάθε επανάληψης και αν είναι αληθής (true), ο κώδικας εντός του "while" εκτελείται. Η εκτέλεση συνεχίζεται, μέχρι η συνθήκη να γίνει ψευδής, οπότε η εκτέλεση συνεχίζεται από το επόμενο σημείο μετά το τμήμα "while".

Η γενική μορφή της εντολής "while" είναι η εξής:

```
while (συνθήκη) {  
    // Κώδικας που εκτελείται κατά τη διάρκεια της επανάληψης  
}
```

Ο κώδικας μέσα στο τμήμα "while" εκτελείται, μόνο όσο η συνθήκη είναι αληθής. Αν η συνθήκη είναι αρχικά ψευδής, ο κώδικας μέσα στο τμήμα "while" δεν εκτελείται καθόλου.

Ας δούμε ένα παράδειγμα: Έστω, ότι θέλουμε να εκτυπώσουμε τους αριθμούς από 1 έως 5 χρησιμοποιώντας την εντολή "while" σε γλώσσα όπως η C++:

```
int i = 1;  
while (i <= 5) {  
    cout << i << endl;  
    i++;  
}
```

Σε αυτό το παράδειγμα, ξεκινούμε με την αρχική τιμή της μεταβλητής "i" να είναι 1. Η συνθήκη ελέγχει αν η "i" είναι μικρότερη ή ίση του 5. Όσο η συνθήκη είναι αληθής, ο κώδικας μέσα στο "while" εκτελείται, εκτυπώνοντας την τρέχουσα τιμή της "i". Στο τέλος κάθε επανάληψης, αυξάνουμε την "i" κατά 1 (i++). Η διαδικασία συνεχίζεται, μέχρι η συνθήκη να γίνει ψευδής, δηλαδή όταν η "i" γίνει μεγαλύτερη από 5.

### 7. Η εντολή "**delay**":

χρησιμοποιείται συνήθως σε προγραμματισμό μικροελεγκτών ή αρχιτεκτονικών, που υποστηρίζουν εικονικό χρονισμό. Ο σκοπός της είναι να προκαλέσει μια παύση ή καθυστέρηση στην εκτέλεση του προγράμματος για ένα συγκεκριμένο χρονικό διάστημα.

Συνήθως, η εντολή "delay", παίρνει ως όρισμα έναν ακέραιο αριθμό, που αναπαριστά το χρονικό διάστημα σε मिलिदευτερόλεπτα (ms) ή μικροδευτερόλεπτα (μs). Κατά τη διάρκεια της καθυστέρησης, η εκτέλεση του προγράμματος παύει, και η συσκευή αναμένει το πέρας της περιόδου καθυστέρησης, προτού συνεχίσει την εκτέλεση του επόμενου κώδικα.

Η εντολή "delay" είναι χρήσιμη σε πολλές περιπτώσεις, όπως για την προσθήκη χρονικών καθυστερήσεων μεταξύ των ενεργειών, τον συγχρονισμό των ενεργειών ή απλά για να προσομοιώσετε χρονικές καθυστερήσεις.

Εδώ είναι ένα παράδειγμα χρήσης της εντολής "delay" στην Arduino C++:

```
void setup() {  
    // Αρχικές ρυθμίσεις  
}  
  
void loop() {  
    // Κώδικας επανάληψης  
  
    // Καθυστέρηση για 1 δευτερόλεπτο (1000 ms)  
    delay(1000);  
}
```

Σε αυτό το παράδειγμα, ο κώδικας μέσα στη συνάρτηση `loop()` θα επαναλαμβάνεται συνεχώς. Αμέσως μετά την εκτέλεση του κώδικα, η εντολή `delay(1000)` προκαλεί μια παύση για 1 δευτερόλεπτο, πριν συνεχίσει την επόμενη επανάληψη. Αυτό δημιουργεί ένα διάστημα ενός δευτερολέπτου μεταξύ των εκτελέσεων του κώδικα μέσα στον βρόχο.

### 8. Η εντολή "***Serial print***":

χρησιμοποιείται συνήθως σε γλώσσες προγραμματισμού, για την αποστολή δεδομένων από έναν μικροελεγκτή σε έναν υπολογιστή ή άλλη συσκευή, μέσω της σειριακής θύρας.

Η συνάρτηση `Serial.print()` παίρνει ως **αναφορά** μια τιμή ή μια μεταβλητή και την αποστέλλει στη σειριακή θύρα. Ανάλογα με τον τύπο της τιμής ή της μεταβλητής, η `Serial.print()` μπορεί να εκτυπώσει ακέραιους αριθμούς, δεκαδικούς αριθμούς, χαρακτήρες, συμβολοσειρές και άλλους τύπους δεδομένων.

Ας δούμε ένα παράδειγμα χρήσης της εντολής `Serial.print()` στην Arduino C++:

```
int sensorValue = 500;
float temperature = 25.5;

void setup() {
  Serial.begin(9600); // Αρχικοποίηση της σειριακής θύρας
}

void loop() {
  // Αποστολή δεδομένων στη σειριακή θύρα
  Serial.print("Sensor value: ");
  Serial.print(sensorValue);
  Serial.print(", Temperature: ");
  Serial.print(temperature);
  Serial.println(" °C");

  delay(1000);
}
```

Σε αυτό το παράδειγμα, οι τιμές του αισθητήρα (`sensorValue`) και η θερμοκρασία (`temperature`) εκτυπώνονται μέσω της σειριακής θύρας με τη χρήση της `Serial.print()`. Ο κώδικας του παραδείγματος εκτυπώνει ένα μήνυμα, που περιλαμβάνει τις τιμές των μεταβλητών, όπως `"Sensor value: 500, Temperature: 25.5 °C"`. Η εντολή `Serial.println()` προσθέτει έναν χαρακτήρα αλλαγής γραμμής (`'\n'`), μετά την εκτύπωση του μηνύματος.

Ανοίγοντας τον παράλληλο παρατηρητή (`serial monitor`) στο περιβάλλον ανάπτυξης της Arduino, μπορείτε να δείτε τα μηνύματα που αποστέλλονται μέσω της σειριακής θύρας.

### 9. Η εντολή "***digital Read***":

χρησιμοποιείται συνήθως σε γλώσσες προγραμματισμού, όπως η Arduino C++, για να διαβάσει την κατάσταση μιας ψηφιακής εισόδου. Μια ψηφιακή είσοδος μπορεί να είναι σε δύο καταστάσεις: `"HIGH"` (υψηλό) ή `"LOW"` (χαμηλό), αντιπροσωπευόμενες από τις τιμές 1 και 0, αντίστοιχα.

Η εντολή `digitalRead` παίρνει ως όρισμα τον αριθμό του ακροδέκτη (`pin`), στον οποίο είναι

συνδεδεμένη η ψηφιακή είσοδος και επιστρέφει την κατάσταση της εισόδου, ως αποτέλεσμα. Η επιστρεφόμενη τιμή είναι είτε "HIGH" (1), εάν η είσοδος είναι σε υψηλή κατάσταση, είτε "LOW" (0), εάν η είσοδος είναι σε χαμηλή κατάσταση.

Ας δούμε ένα παράδειγμα χρήσης της εντολής "digitalRead" στην Arduino C++:

```
int buttonPin = 2;
int ledPin = 13;

void setup() {
  pinMode(buttonPin, INPUT);
  pinMode(ledPin, OUTPUT);
}

void loop() {
  int buttonState = digitalRead(buttonPin);

  if (buttonState == HIGH) {
    digitalWrite(ledPin, HIGH);
  } else {
    digitalWrite(ledPin, LOW);
  }
}
```

Σε αυτό το παράδειγμα, υπάρχει ένας διακόπτης, που είναι συνδεδεμένος στον ακροδέκτη 2 (buttonPin), και ένα LED, που είναι συνδεδεμένο στον ακροδέκτη 13 (ledPin). Η εντολή "pinMode" ρυθμίζει τον ακροδέκτη του διακόπτη, ως είσοδο, και τον ακροδέκτη του LED, ως έξοδο.

Στη συνέχεια, η εντολή "digitalRead(buttonPin)" διαβάζει την κατάσταση του διακόπτη και αποθηκεύει την τιμή στη μεταβλητή "buttonState". Αν η κατάσταση του διακόπτη είναι "HIGH", τότε η εντολή "digitalWrite(ledPin, HIGH)" ενεργοποιεί το LED, ενώ αν η κατάσταση του διακόπτη είναι "LOW", τότε η εντολή "digitalWrite(ledPin, LOW)" απενεργοποιεί το LED.

Έτσι, το LED θα ανάβει, όταν ο διακόπτης είναι σε κατάσταση HIGH/ON και θα σβήσει, όταν ο διακόπτης είναι σε κατάσταση LOW/OFF.

### 10. Η εντολή "*analog Read*":

χρησιμοποιείται συνήθως σε γλώσσες προγραμματισμού όπως η Arduino C++ για την ανάγνωση τιμών από αναλογικές εισόδους. Μια αναλογική είσοδος παρέχει συνεχείς τιμές σήματος, σε αντίθεση με τις διακριτές τιμές "HIGH" και "LOW", που παρέχονται από τις ψηφιακές εισόδους.

Για να χρησιμοποιήσετε την εντολή "analogRead", θα πρέπει να συνδέσετε την αναλογική είσοδο σας σε έναν αναλογικό ακροδέκτη (pin) στον μικροελεγκτή (όπως η Arduino). Η εντολή "analogRead" διαβάζει την τάση που παρέχεται από την αναλογική είσοδο και επιστρέφει μια τιμή από το 0 έως το 1023, αντιπροσωπεύοντας την τάση ανάμεσα στη γείωση (0V) και την αναφορά τάσης (συνήθως 5V).



Ας δούμε ένα παράδειγμα χρήσης της εντολής "analogRead" στην Arduino C++:

```
int analogPin = A0;

void setup() {
  Serial.begin(9600);
}

void loop() {
  int sensorValue = analogRead(analogPin);
  float voltage = sensorValue * (5.0 / 1023.0);

  Serial.print("Sensor value: ");
  Serial.print(sensorValue);
  Serial.print(", Voltage: ");
  Serial.print(voltage);
  Serial.println(" V");
  delay(1000);
}
```

Σε αυτό το παράδειγμα, η αναλογική είσοδος είναι συνδεδεμένη στον αναλογικό ακροδέκτη A0. Η εντολή "Serial.begin(9600)" αρχικοποιεί τη σειριακή επικοινωνία με μια ταχύτητα 9600 baud.

Στη συνέχεια, η εντολή "analogRead(analogPin)" διαβάζει την τιμή από την αναλογική είσοδο και αποθηκεύει την τιμή στη μεταβλητή "sensorValue". Η μεταβλητή "sensorValue" αναπαριστά την αναλογική τιμή, σε μορφή ακέραιου αριθμού από 0 έως 1023.

Στη συνέχεια, η τάση υπολογίζεται με βάση την τιμή του "sensorValue" και αποθηκεύεται στη μεταβλητή "voltage". Τέλος, τα μηνύματα εκτυπώνονται μέσω της σειριακής θύρας, με τη χρήση της εντολής "Serial.print" και "Serial.println".

Συνήθως, η τιμή που διαβάζεται από μια αναλογική είσοδο μετατρέπεται σε κάποια φυσική μεγέθυνση (όπως θερμοκρασία ή φωτεινότητα), χρησιμοποιώντας κατάλληλους μετασχηματισμούς και μετρήσεις.

### 11. Η εντολή "**break**":

χρησιμοποιείται για να διακόπτει την εκτέλεση ενός βρόχου (loop) ή ενός switch statement (εναλλαγή). Η εντολή `break` είναι ένας τύπος ελέγχου ροής (flow control), που επιτρέπει στο πρόγραμμά σας να βγει από τον τρέχοντα βρόχο ή να τερματίσει την εκτέλεση ενός switch statement.

Συνοπτικά, χρησιμοποιείται με την εξής σύνταξη:

```
c
break;
```


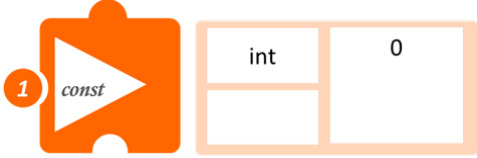





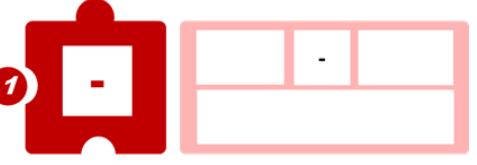
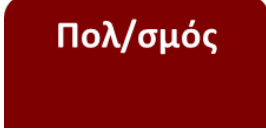
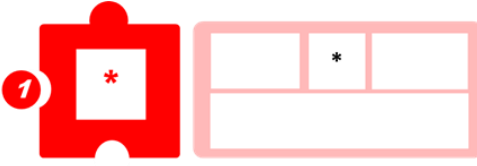
Ένα παράδειγμα χρήσης της εντολής `break` περιλαμβάνει:

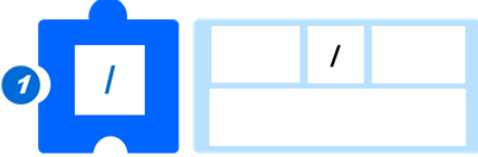
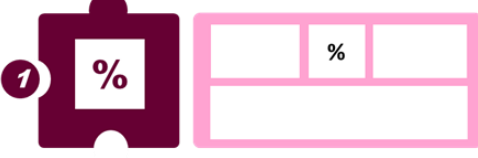

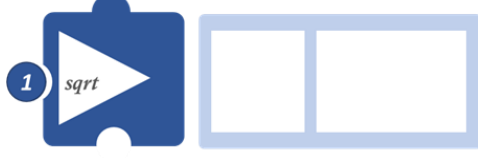
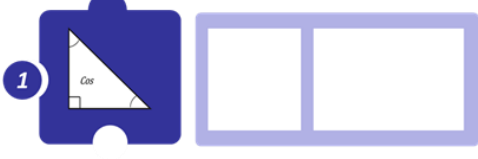

### **Βρόχοι (Loops):**


```
for (int i = 1; i <= 10; i++) {  
    if (i == 5) {  
        break; // Τερματίζει τον βρόχο όταν i γίνει 5  
    }  
    printf("%d\n", i);  
}
```

Σε αυτό το παράδειγμα, ο βρόχος `for` θα τρέξει μέχρι το `i` να γίνει 5, και τότε η εντολή `break` θα τερματίσει τον βρόχο.

## 1.3.3. Μενού Επιλογής Βασικών Μαθηματικών Εντολών

Εικονίδιο Επιλογής Εντολής	Περιγραφή	Πλακίδιο Εντολής
	<b><u>const (Σταθερά/constant):</u></b> Αυτή η εντολή προσδιορίζει μια μεταβλητή με σταθερή τιμή, η οποία δεν μπορεί να αλλάξει κατά τη διάρκεια εκτέλεσης του προγράμματος.	 Δείτε το σχετικό βίντεο <a href="#">M1 ENTOLH cons</a> με τις οδηγίες προγραμματισμού της εντολής
	<b><u>Var (Μεταβλητή):</u></b> Αυτή η εντολή προσδιορίζει μια μεταβλητή που μπορεί να πάρει διαφορετικές τιμές. Οι τιμές της μεταβλητής Var μπορούν να αλλάζουν και κατά τη ροή του προγράμματος.	 Δείτε το σχετικό βίντεο <a href="#">M2 ENTOLH var</a> με τις οδηγίες προγραμματισμού της εντολής
	<b><u>Πρόσθεση:</u></b> Αυτή η εντολή + (συν), καθορίζει δύο τελεστές για την παραγωγή ενός αθροίσματος.	 Δείτε το σχετικό βίντεο <a href="#">M3 ENTOLH Addition</a> με τις οδηγίες προγραμματισμού της εντολής
	<b><u>Αφαίρεση:</u></b> Αυτή η εντολή –(μείον), καθορίζει δύο τελεστές για τη παραγωγή της διαφοράς του δεύτερου από τον πρώτο.	 Δείτε το σχετικό βίντεο <a href="#">M4 ENTOLH Subtraction</a> με τις οδηγίες προγραμματισμού της εντολής
	<b><u>Πολλαπλασιασμός:</u></b> Αυτή η εντολή * (επί), καθορίζει δύο τελεστές για την παραγωγή του μεταξύ τους γινομένου.	 Δείτε το σχετικό βίντεο <a href="#">M5 ENTOLH Multiplication</a> με τις οδηγίες προγραμματισμού της εντολής

<p><b>Διαίρεση</b></p>	<p><b>Διαίρεση:</b> Αυτή η εντολή / (δια), καθορίζει δύο τελεστές για τη παραγωγή του μεταξύ τους αποτελέσματος διαίρεσης.</p>	 <p>Δείτε το σχετικό βίντεο <a href="#">M6 ENTOLH Division</a> με τις οδηγίες προγραμματισμού της εντολής</p>
<p><b>Υπόλοιπο Διάρεσης</b></p>	<p><b>Υπόλοιπο Διάρεσης:</b> Αυτή η εντολή καθορίζει δύο τελεστές για τη παραγωγή του υπολοίπου της μεταξύ τους διαίρεσης.</p>	 <p>Δείτε το σχετικό βίντεο <a href="#">M7 ENTOLH Division Remainder</a> με τις οδηγίες προγραμματισμού της εντολής</p>
<p><b>Δύναμη</b></p>	<p><b>Δύναμη (pow):</b> Αυτή η εντολή υπολογίζει την τιμή ενός αριθμού υψωμένου σε κάποια δύναμη. Μπορεί να χρησιμοποιηθεί για την αύξηση ενός αριθμού σε κλασματική ισχύ. Αυτό είναι χρήσιμο για τη δημιουργία εκθετικής χαρτογράφησης τιμών ή καμπυλών.</p>	 <p>Δείτε το σχετικό βίντεο <a href="#">M8 ENTOLH Pow</a> με τις οδηγίες προγραμματισμού της εντολής</p>
<p><b>Τετραγωνική Ρίζα</b></p>	<p><b>Τετραγωνική ρίζα (sqrt):</b> Αυτή η εντολή υπολογίζει την τετραγωνική ρίζα ενός αριθμού.</p>	
<p><b>Συνημίτονο</b></p>	<p><b>Συνημίτονο(con):</b> Η εντολή αυτή υπολογίζει το συνημίτονο μια γωνίας, που ορίζουμε στο πρόγραμμα.</p>	 <p>Δείτε το σχετικό βίντεο <a href="#">M9 ENTOLH Trigonometry</a> με τις οδηγίες προγραμματισμού της εντολής</p>
<p><b>Ημίτονο</b></p>	<p><b>Ημίτονο (sin):</b> Η εντολή αυτή υπολογίζει το ημίτονο μια γωνίας, που ορίζουμε στο πρόγραμμα.</p>	 <p>Δείτε το σχετικό βίντεο <a href="#">M9 ENTOLH Trigonometry</a> με τις οδηγίες προγραμματισμού της εντολής</p>

<p><b>Εφαπτομένη</b></p>	<p><b>Εφαπτομένη (tan):</b> Η εντολή αυτή υπολογίζει την εφαπτομένη μια γωνίας που ορίζουμε στο πρόγραμμα..</p>	 <p>Δείτε το σχετικό βίντεο <a href="#">M9 ENTOLH Trigonometry</a> με τις οδηγίες προγραμματισμού της εντολής</p>
--------------------------	---	---


## 1.4 Γρήγορος Οδηγός Δραστηριοτήτων

Σε κάθε δραστηριότητα του παρακάτω εγχειριδίου, εμφανίζεται ένας πίνακας, ο οποίος αλλάζει ανάλογα με τον προγραμματισμό του κυκλώματος που απαιτείται για την κάθε δραστηριότητα. **Ας δούμε ένα παράδειγμα.** Δείτε τον πίνακα παρακάτω. Χωρίζεται σε 4 διαφορετικά μέρη/στήλες. Ας δούμε μία-μία τι σημαίνουν αυτές οι στήλες για κάθε γραμμή του πίνακα.


1) Γραμμή Εντολών	2) Εντολή	3) Συσκευή	4) Θύρα / Κατάσταση
1 <sup>st</sup>	"if" / "then"	DJS08	8 / ΚΛΕΙΣΤΟ
2 <sup>nd</sup>	"digital write"	DJX07	9 / ΑΝΟΙΧΤΟ
1 <sup>st</sup>	"if" / "then"	DJS08	8 / ΑΝΟΙΧΤΟ
2 <sup>nd</sup>	"digital write"	DJX07	9 / ΚΛΕΙΣΤΟ

**1)** Στη στήλη «**Γραμμή Εντολών**», εμφανίζεται η Γραμμή Εντολών της συνάρτησης. Ξεκινάτε πάντα με τη 1<sup>η</sup> Γραμμή Εντολών.

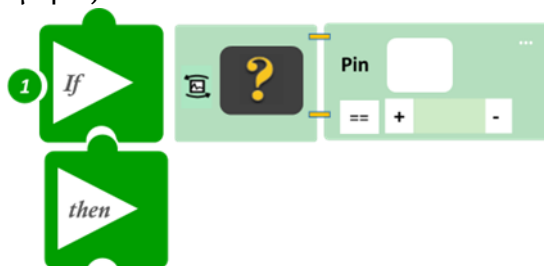
Εάν θέλετε να εισάγετε 2<sup>η</sup> Γραμμή Εντολών ή να μεταφερθείτε από τη γραμμή 2 στη γραμμή

1, πιέστε το εικονίδιο , στην αρχική οθόνη του προγράμματος Ard:icon


**2)** Στη στήλη «**Εντολή**», εμφανίζεται η εντολή που πρέπει να επιλέξετε. Η επιλογή γίνεται

πιέζοντας το εικονίδιο , στην αρχική οθόνη του προγράμματος Ard:icon. Εμφανίζεται ο πίνακας εντολών, όπου επιλέγετε την εντολή που θέλετε. Μπορείτε να μεταβείτε στις υπόλοιπες επιλογές, μετακινώντας τη μπάρα στο κάτω μέρος.

Επιλέξτε την εντολή (στο παράδειγμά μας ξεκινάμε από την εντολή "if").

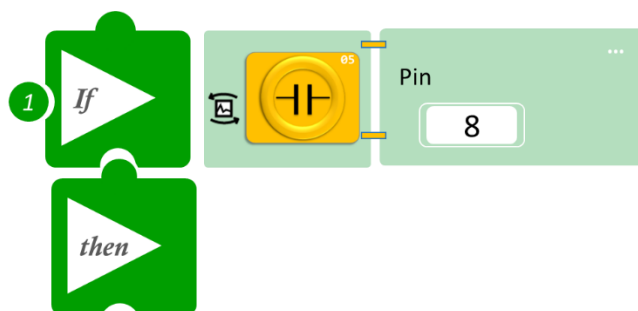


**3)** Στη στήλη «**Συσκευή**», εμφανίζεται η συσκευή που πρέπει να επιλέξετε.

Όπως έχετε ήδη δει, επιλέγοντας το εικονίδιο , πάνω στο πλακίδιο της εντολής, θα εμφανιστεί η λίστα των αναλογικών και ψηφιακών συσκευών. Μπορείτε να μεταβείτε στις υπόλοιπες επιλογές, μετακινώντας την μπάρα στα δεξιά. Επιλέξτε τη συσκευή που ζητάει ο πίνακας **DJS08 Διακόπτης Αφής**.


**4)** Στη στήλη «**Θύρα/Κατάσταση**», εμφανίζονται οι επιμέρους παράμετροι της εντολής. Στο παράδειγμά μας είναι η θύρα 8 και κατάσταση OFF. Προγραμματίστε τις παραμέτρους της εντολής "if", όπως ήδη έχετε μάθει.

Η πρώτη εντολή του πίνακα είναι έτοιμη: "ΑΝ ο Διακόπτης Αφής είναι σε κατάσταση **OFF**".



Τώρα ας προχωρήσουμε στη 2<sup>η</sup> γραμμή του πίνακα, όπου θα προγραμματίσετε τη συσκευή που θα ελέγχει η συνθήκη της “if”.

2 <sup>nd</sup>	“digital write”	DJX07	9 / ΑΝΟΙΧΤΟ
-----------------	-----------------	-------	-------------

Στη 1<sup>η</sup> στήλη, εμφανίζεται πάλι η Γραμμή Εντολών που πρέπει να ξεκινήσετε να προγραμματίζετε. Οπότε, μεταφερθείτε στη 2<sup>η</sup> γραμμή, πιέζοντας το εικονίδιο , στην αρχική οθόνη του προγράμματος Ard:icon.

Στη 2<sup>η</sup> στήλη, είναι η εντολή που πρέπει να επιλέξετε “digital write”.



Στην 3<sup>η</sup> στήλη, εμφανίζεται η συσκευή που πρέπει να επιλέξετε, στο παράδειγμα μας **DJX07 Πράσινο LED**.

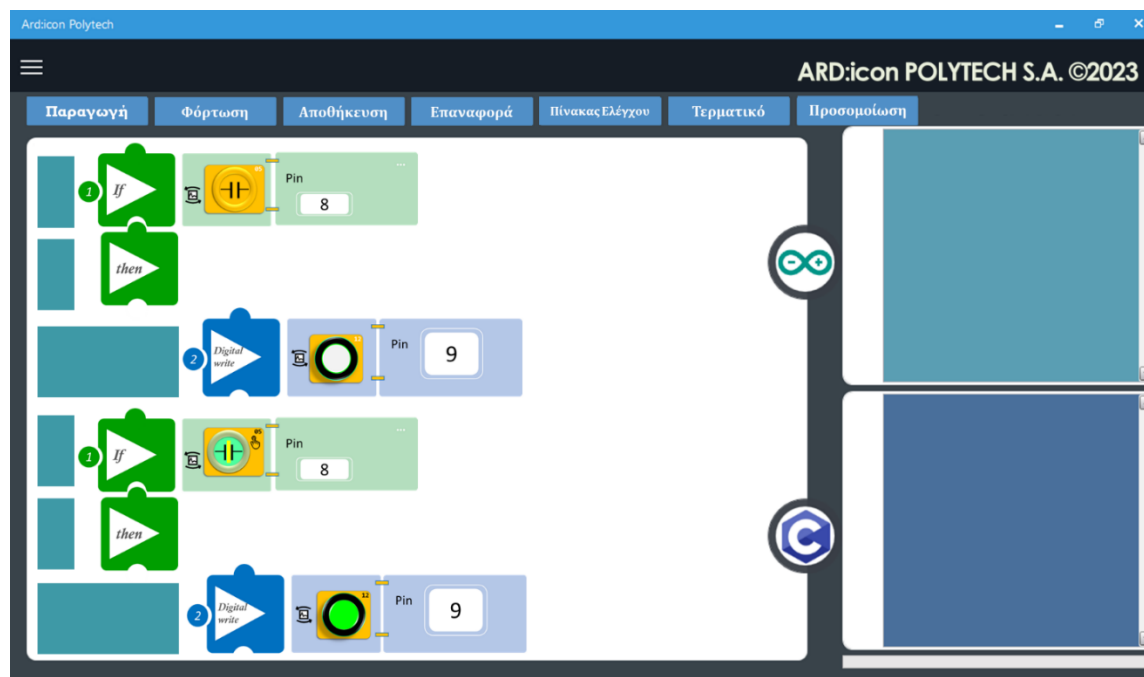
Και τέλος, στην 4<sup>η</sup> στήλη, μπορείτε να δείτε τις επιμέρους παραμέτρους, δηλαδή Θύρα 9 και κατάσταση ON.



Συνεχίστε με τον ίδιο τρόπο να προγραμματίζετε τις υπόλοιπες 2 εντολές του πίνακα

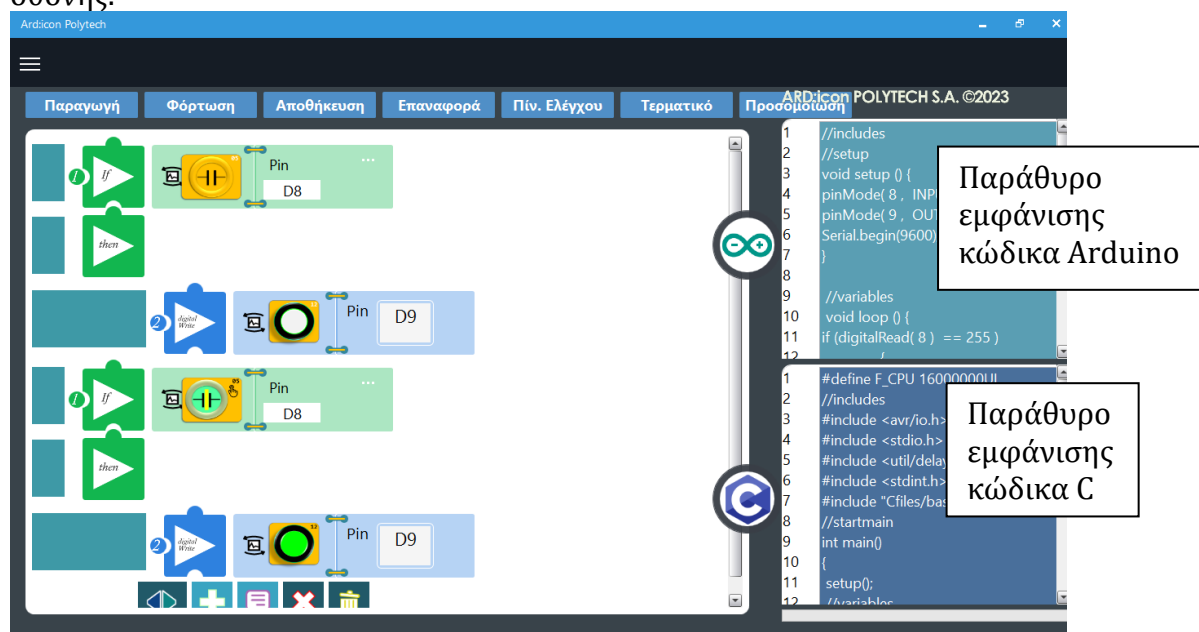
1 <sup>st</sup>	“if” / “then”	DJS08	8 / ΑΝΟΙΧΤΟ
2 <sup>nd</sup>	“digital write”	DJX07	9 / ΚΛΕΙΣΤΟ

Στο τέλος, τα πλακίδια των εντολών θα είναι όπως εμφανίζονται παρακάτω.



### 1.4.1 Παραγωγή Κώδικα Κυκλώματος

Στη συνέχεια, πατήστε το κουμπί «Παραγωγή» **Παραγωγή**, για να παράγετε τον κώδικα. Οι κώδικες σε Arduino και C θα εμφανιστούν στα δύο παράθυρα, δεξιά της αρχικής οθόνης.



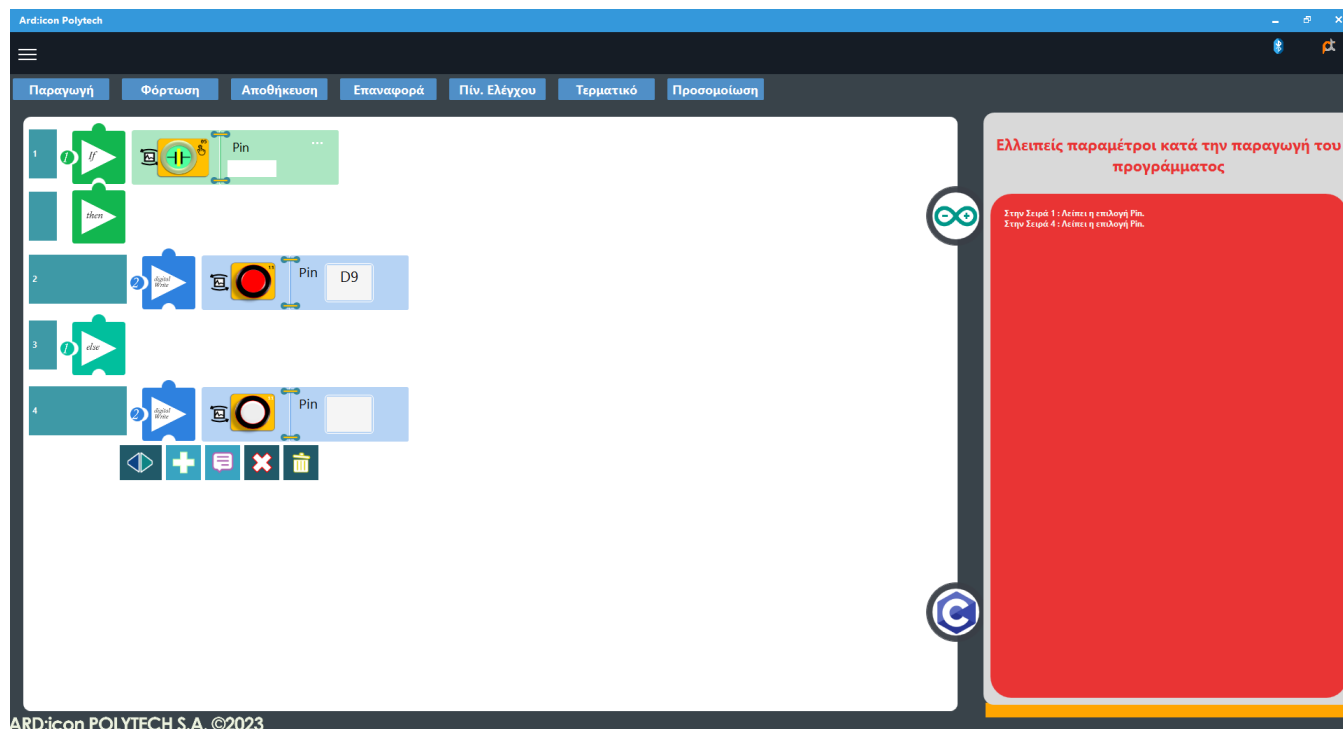
Επιλέγοντας πάνω στο εικονίδιο , μπορείτε να ανοίξετε την οθόνη με τον πλήρη κώδικα



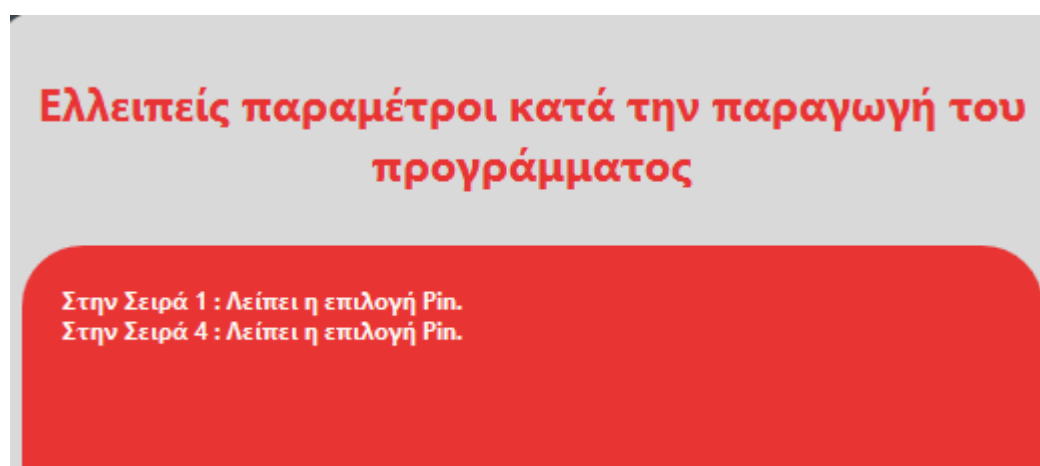
## ARDUINO.

Αντίστοιχα, επιλέγοντας πάνω στο εικονίδιο , μπορείτε να ανοίξετε την οθόνη με τον πλήρη κώδικα C.

Σε περίπτωση που κάποιες παράμετροι δεν έχουν οριστεί στον κώδικα, τότε ένα παράθυρο θα αναδυθεί στη δεξιά πλευρά της οθόνη προγραμματισμού.



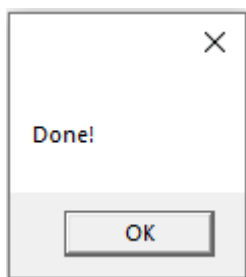
Στο παράθυρο αυτό όπως μπορείτε να δείτε στην παρακάτω εικόνα εμφανίζονται οι ελλείψεις παραμέτρων καθώς και η σειρά που βρίσκονται.



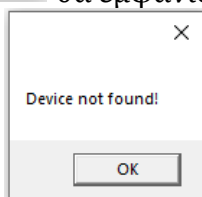
### 1.4.2 Φόρτωση κώδικα στον ελεγκτή ARD:icon

## Φόρτωση

Έπειτα, επιλέξτε το κουμπί «Φόρτωση», για να φορτώσετε το πρόγραμμα στον ελεγκτή σας. Βεβαιωθείτε ότι ο ελεγκτής είναι συνδεδεμένος στον υπολογιστή σας και οι συσκευές στις θύρες, που έχετε προγραμματίσει.



Το εικονίδιο θα εμφανιστεί πάνω στην οθόνη της εφαρμογής. Επιλέξτε “OK”



Αν εμφανιστεί το εικονίδιο, αποσυνδέστε τον ελεγκτή από τον υπολογιστή, επανασυνδέστε τον ελεγκτή και επιλέξτε “OK” και επαναφορτώστε το πρόγραμμα στον

## Φόρτωση

ελεγκτή.

Μόλις προγραμματίσατε το κύκλωμα έτσι, ώστε:

“ΑΝ ο Διακόπτης Αφής είναι σε κατάσταση OFF, ΤΟΤΕ να σβήνει το Πράσινο LED.  
ΑΝ ο Διακόπτης Αφής είναι σε κατάσταση ON, ΤΟΤΕ να ανάβει το Πράσινο LED”.  
Δοκιμάστε το κύκλωμα σας, να δείτε πώς λειτουργεί.

### 1.4.3 Πίνακας Ελέγχου Εφαρμογής ARD:icon

Όπως μπορείτε να δείτε πάνω στη οθόνη της εφαρμογής, υπάρχουν και τα κουμπιά

Πίνακας Ελέγχου

και

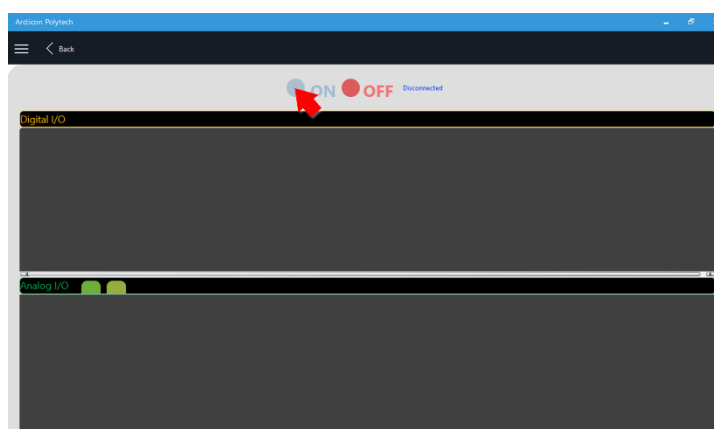
Προσομοίωση

Ας δούμε τώρα τα κουμπιά αυτά.

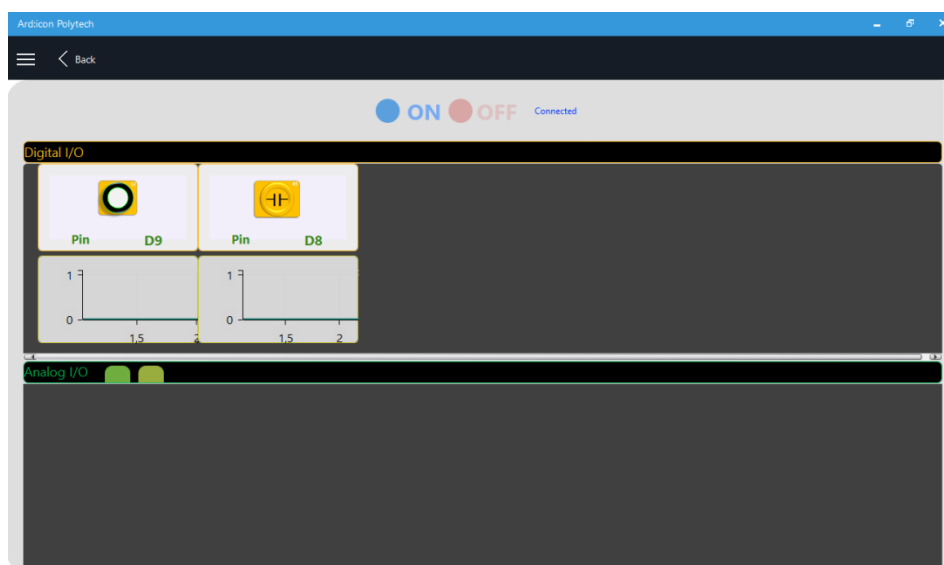
## Πίνακας Ελέγχου

Πατήστε το κουμπί. Θα εισέλθετε στην οθόνη του πίνακα ελέγχου.

Επιλέξτε το εικονίδιο ON

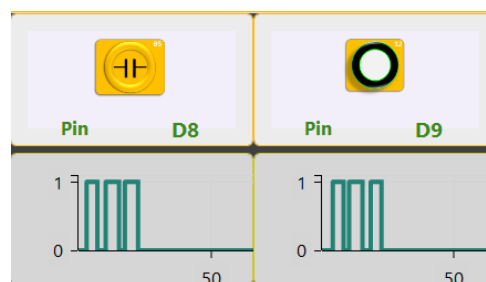


Αμέσως εμφανίζονται οι συσκευές που έχετε προγραμματίσει και οι παράμετροί τους. Πιέστε τον διακόπτη και δείτε τις αλλαγές που εμφανίζονται στις συσκευές. Επιλέξτε OFF , πριν βγείτε από την οθόνη του πίνακα ελέγχου.



Στη διπλανή οθόνη, βλέπετε τις δύο συσκευές που έχετε προγραμματίσει στο κύκλωμα σας. Το εικονίδιο του LED αναβοσβήνει, ανάλογα με την κατάσταση του Διακόπτη ON/OFF.

Επίσης, μπορείτε να δείτε και τα γραφήματα των δύο συσκευών (ψηφιακά 0 και 1), καθώς και ότι λειτουργούν παράλληλα, όπως τα προγραμματίσατε.



#### 1.4.4 Προσομοίωση Εφαρμογής ARD:icon

##### Προσομοίωση

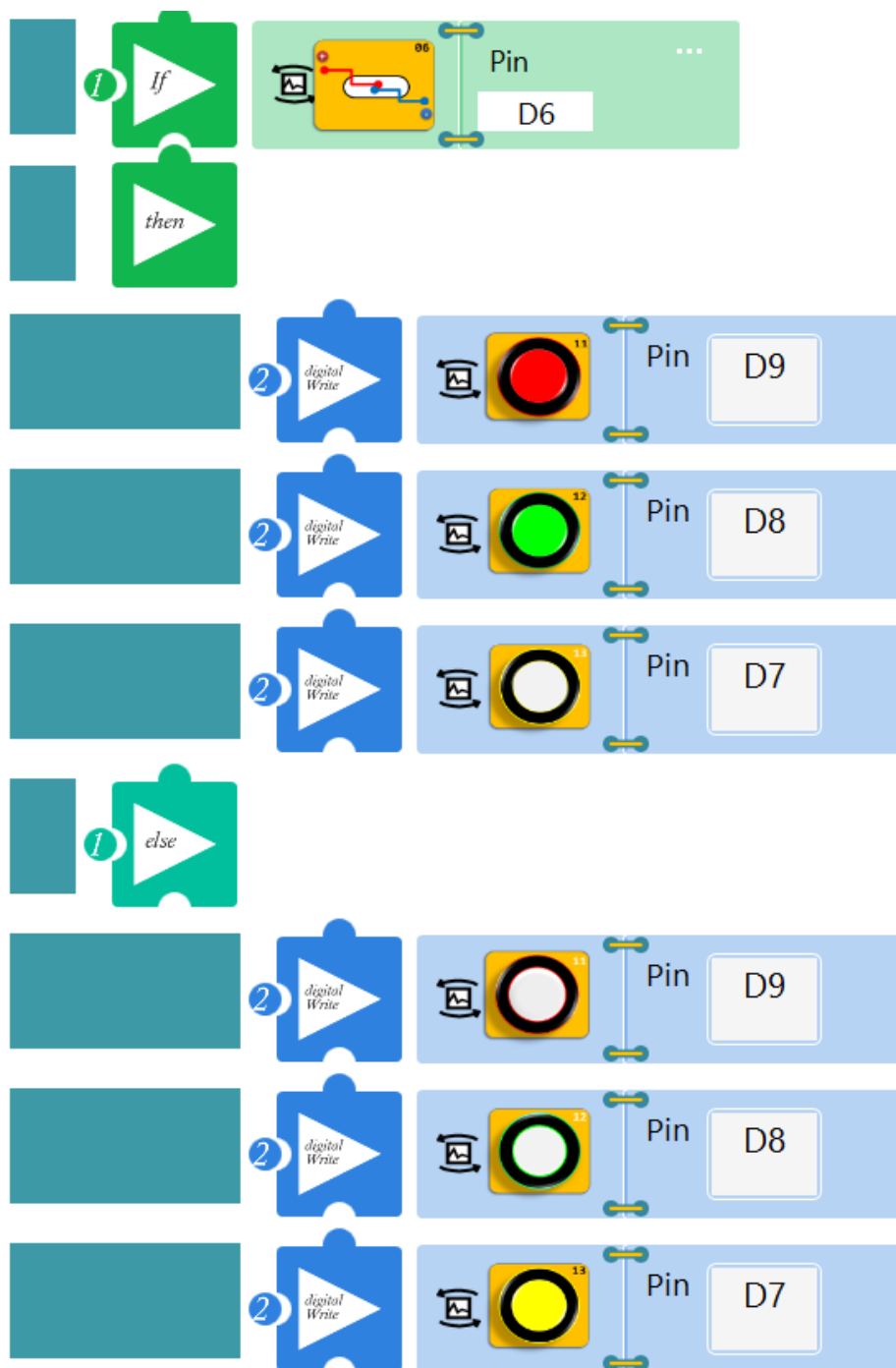
Πατήστε το κουμπί Προσομοίωση. Θα εισέλθετε στην οθόνη του προσομοιωτή. Εδώ, μπορείτε να δείτε τη συνδεσμολογία που έχετε προγραμματίσει, αλλά και την προσομοίωση του κυκλώματός σας, πριν προχωρήσετε στη φόρτωση του προγράμματος.

Βλέποντας την προσομοίωση του κυκλώματος, μπορείτε να διορθώσετε και τυχόν σφάλματα που έχει το πρόγραμμά σας.

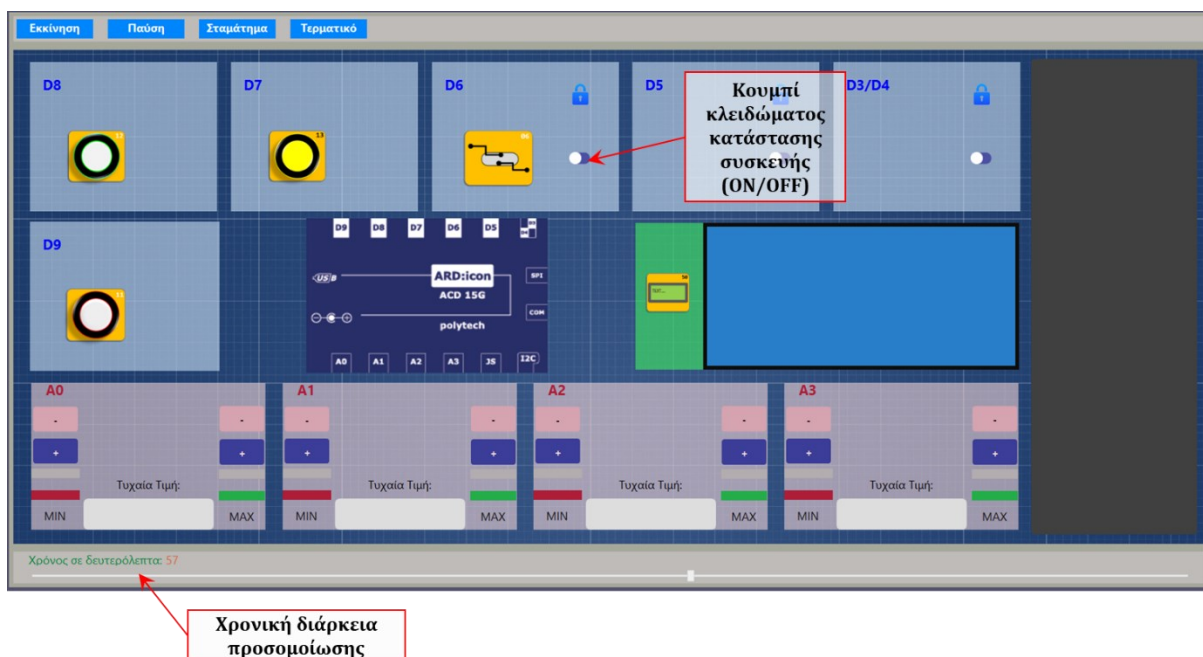
Ας δούμε κάποια παραδείγματα, ένα με ψηφιακή συσκευή εισόδου και ένα με αναλογική συσκευή εισόδου. Για να ελέγξετε τι έχετε προγραμματίσει δε χρειάζεται να έχετε πρώτα συνδέσει τις συσκευές πάνω στον ελεγκτή. Μπορείτε να εισέλθετε στο περιβάλλον της προσομοίωσης μετά την «Παραγωγή» του κώδικα και πριν τη «Φόρτωση» του στον ελεγκτή.

## 1. Ψηφιακή εισόδου – Μαγνητικός διακόπτης (AJS15)

Προγραμματίστε το παρακάτω κύκλωμα και επιλέξτε «Παραγωγή».



Επιλέξτε το κουμπί «Προσομοίωση», **Προσομοίωση** και στην επόμενη οθόνη επιλέξτε «Εκκίνηση», **Εκκίνηση**.



Η προσομοίωση του κυκλώματος ξεκινάει.

### Λειτουργίες προσομοίωσης

- Πάνω από το εικονίδιο της κάθε συσκευής μπορείτε να δείτε τη θύρα που έχετε προγραμματίσει να συνδεθεί η συσκευή, π.χ. D8, D7.
- Πατήστε το κουμπί που βλέπετε δίπλα από τον μαγνητικό διακόπτη προκειμένου να κλειδώσετε τη κατάσταση του σε **ON** ή **OFF**.
- Στο κάτω μέρος της οθόνης μπορείτε να δείτε τη χρονική διάρκεια της προσομοίωσης.
- Η επιλογή της κατάστασης του μαγνητικού διακόπτη από την εφαρμογή είναι τυχαία και δεν ακολουθεί κάποιο συγκεκριμένο μοτίβο.

Επιλέξτε το κουμπί «Παύση» ή «Σταμάτημα», **Παύση** **Σταμάτημα**, για να σταματήσετε την προσομοίωση και πριν εξέλθετε της οθόνης προσομοίωσης. Επιλέξτε το κουμπί “Back”, **< Back**, για να επιστρέψετε στην οθόνη προγραμματισμού.

## 2. Αναλογική συσκευή εισόδου – Αισθητήρας γωνίας (AJS03)

Προγραμματίστε το παρακάτω κύκλωμα και επιλέξτε «Παραγωγή».

The image shows a sequence of programming blocks for a Scratch-like environment, designed to control an angle sensor circuit. The blocks are arranged in a vertical sequence:

- Block 1:** A green 'If' block with a '1' in a circle. The condition is 'Pin A1 < 90'. The 'then' block contains a 'digital Write' block with a yellow sensor icon and 'Pin D9'.
- Block 2:** A green 'then' block.
- Block 3:** A blue 'digital Write' block with a '2' in a circle, containing a yellow sensor icon and 'Pin D8'.
- Block 4:** A green 'If' block with a '1' in a circle. The condition is 'Pin A1 >= 90'. The 'then' block contains a 'digital Write' block with a green sensor icon and 'Pin D9'.
- Block 5:** A green 'then' block.
- Block 6:** A green 'If' block with a '1' in a circle. The condition is 'Pin A1 >= 180'. The 'then' block contains a 'digital Write' block with a red sensor icon and 'Pin D8'.
- Block 7:** A green 'then' block.

At the bottom of the blocks, there is a toolbar with icons for: back, add, comment, delete, and trash.

Επιλέξτε το κουμπί «Προσομοίωση», **Προσομοίωση** και στην επόμενη οθόνη επιλέξτε «Εκκίνηση», **Εκκίνηση**.



Η προσομοίωση του κυκλώματος ξεκινάει.

### Λειτουργίες προσομοίωσης

- Πάνω από το εικονίδιο της κάθε συσκευής μπορείτε να δείτε τη θύρα που έχετε προγραμματίσει να συνδεθεί η συσκευή, π.χ. D9, A1.
- Στο κάτω μέρος της οθόνης μπορείτε να δείτε τη χρονική διάρκεια της προσομοίωσης.
- Η επιλογή της τιμής που λαμβάνει ο αισθητήρας γωνίας από την εφαρμογή είναι τυχαία και δεν ακολουθεί κάποιο συγκεκριμένο μοτίβο.



- Όπως βλέπεται στο κάτω κεντρικό κελί του εικονιδίου του αισθητήρα γωνίας μπορείτε να δείτε την τυχαία τιμή που έχει τη συγκεκριμένη χρονική στιγμή ο αισθητήρας.
- Χρησιμοποιώντας τα κουμπιά + και - που βρίσκονται στις πλευρές "MIN" και "MAX" μπορείτε να αυξήσετε η να μειώσετε το εύρος των τιμών που παίρνει ο αισθητήρας.
- Για παράδειγμα αν θέλετε να ελέγξετε πως προγραμματίσατε το κύκλωμά σας για το εύρος τιμών από 90 - 180 προσαρμόστε ανάλογα το τμήμα "MIN" / ελάχιστο στο επίπεδο

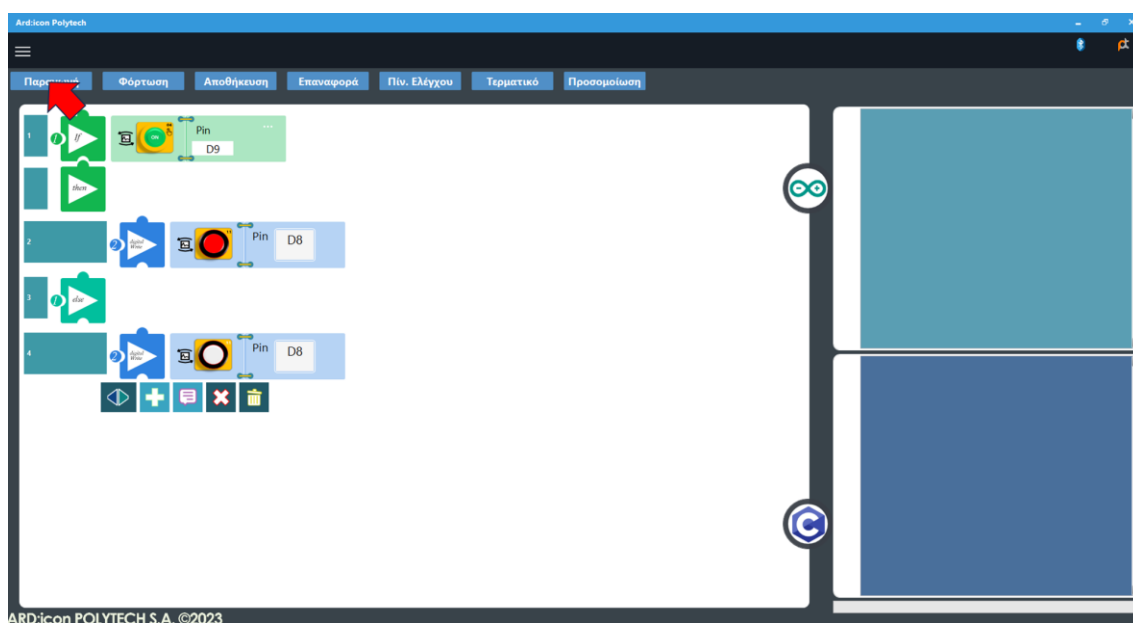
90 και το τμήμα “MAX”/μέγιστο στο επίπεδο 180.

**Σημείωση:** Αν θέλετε να ελέγξετε το επίπεδο <180 και όχι  $\leq 180$  τότε το επίπεδο του “MAX” πρέπει να είναι 175.

Επιλέξτε το κουμπί «Παύση» ή «Σταμάτημα», **Παύση** **Σταμάτημα**, για να σταματήσετε την προσομοίωση και πριν εξέλθετε της οθόνης προσομοίωσης. Επιλέξτε το κουμπί “Back”, **< Back**, για να επιστρέψετε στην οθόνη προγραμματισμού.

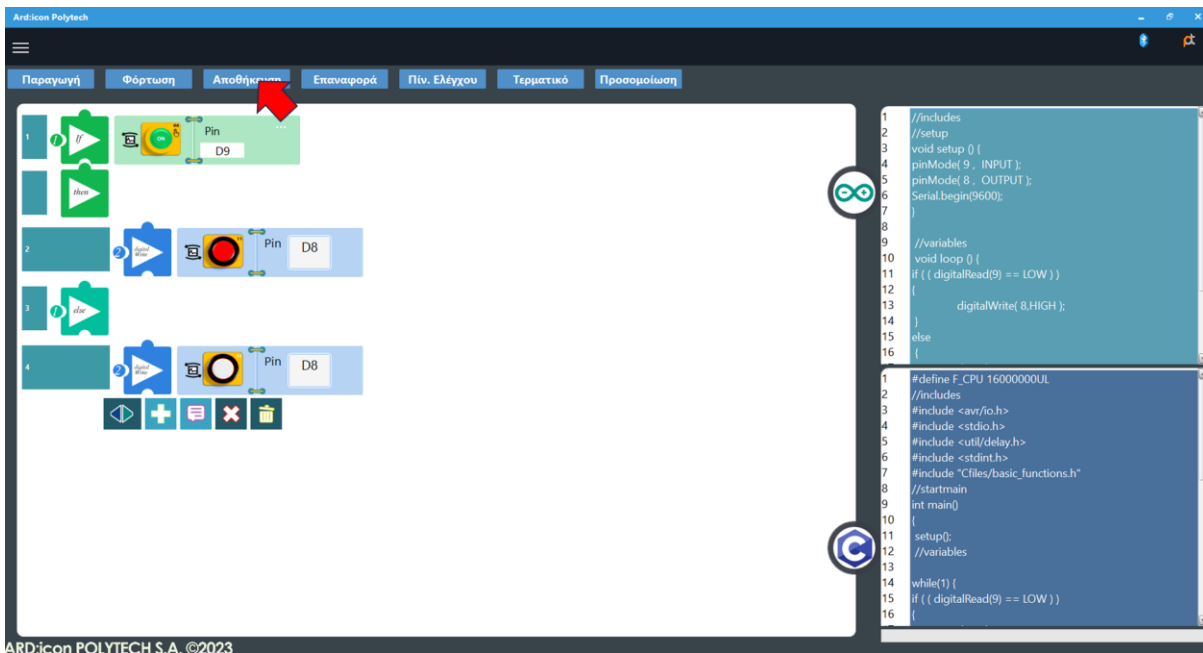
## 1.5 Αποθήκευση και Επαναφορά Ρουτίνας

1. Δημιουργούμε τον κώδικα ανάλογα με το κύκλωμα που θέλουμε να προγραμματίσουμε και έπειτα επιλέγουμε «Παραγωγή».

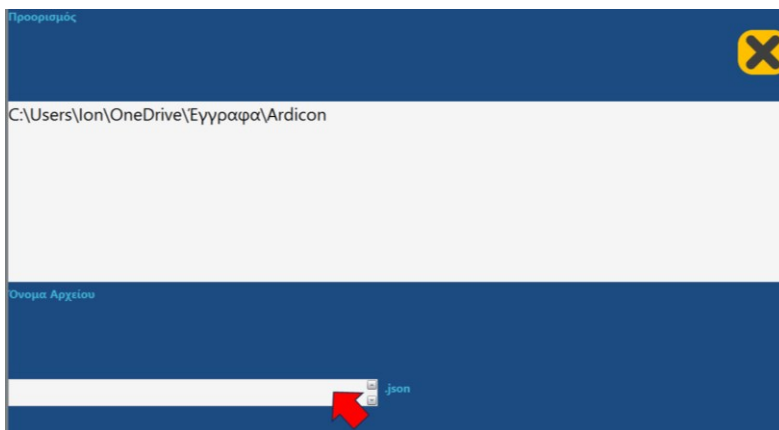


2. Αφού έχει δημιουργηθεί ο κώδικας επιλέγουμε «Αποθήκευση».

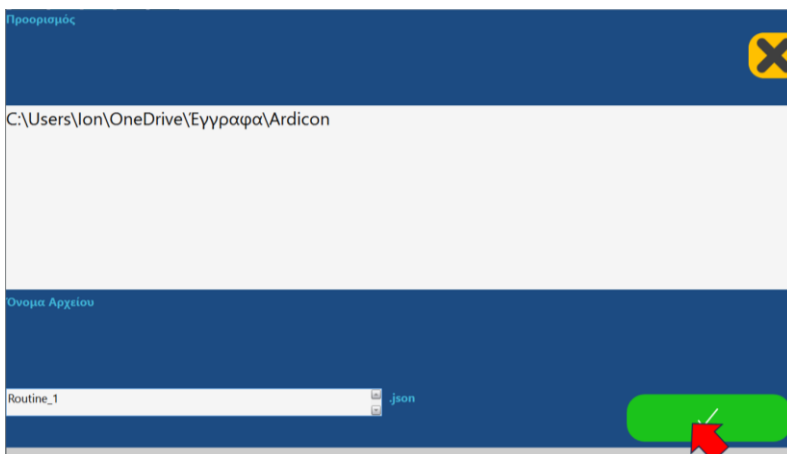




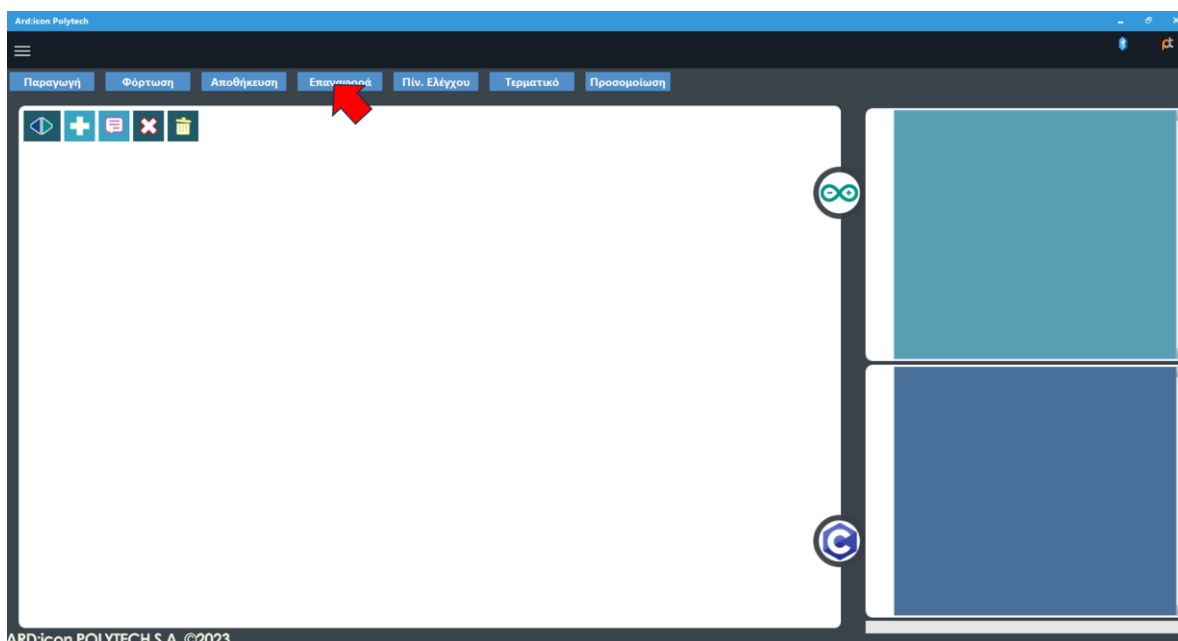
3. Στο αναδυόμενο παράθυρο, στο κελί «Όνομα αρχείου» γράψτε την ονομασία της ρουτίνας που προγραμματίσατε, πχ ROUTINA\_1 (χρησιμοποιήστε μόνο λατινικούς χαρακτήρες και το ειδικό σύμβολο \_)



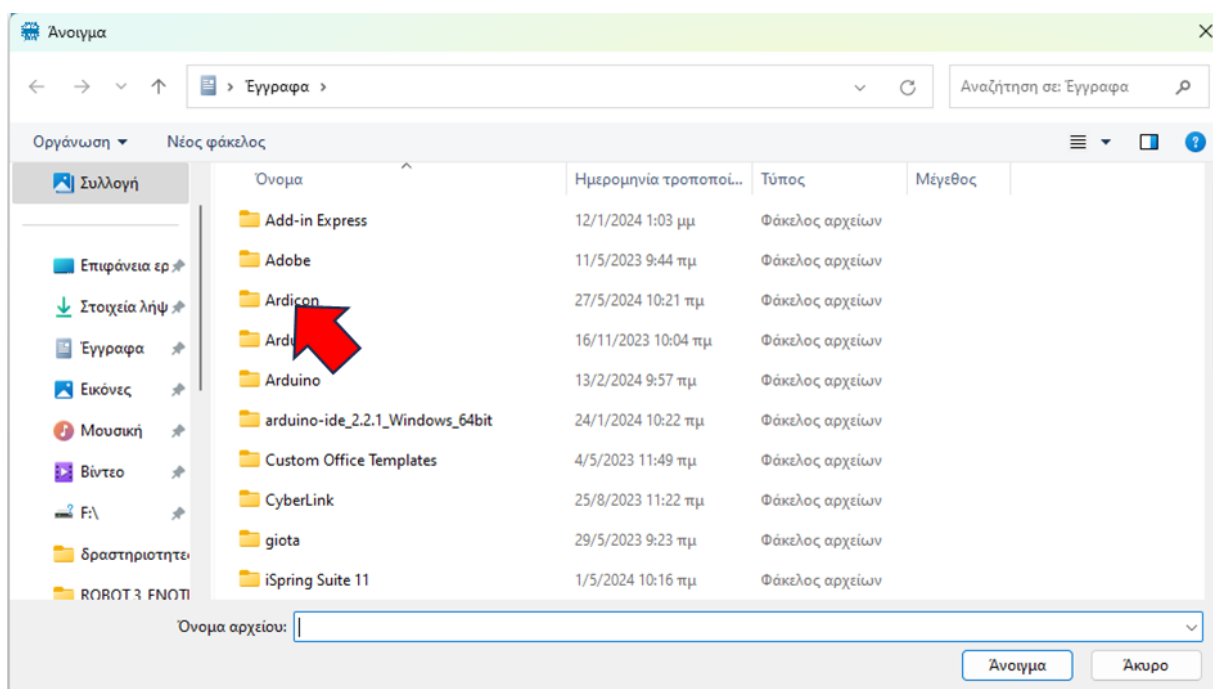
4. Επιλέξτε οκ .



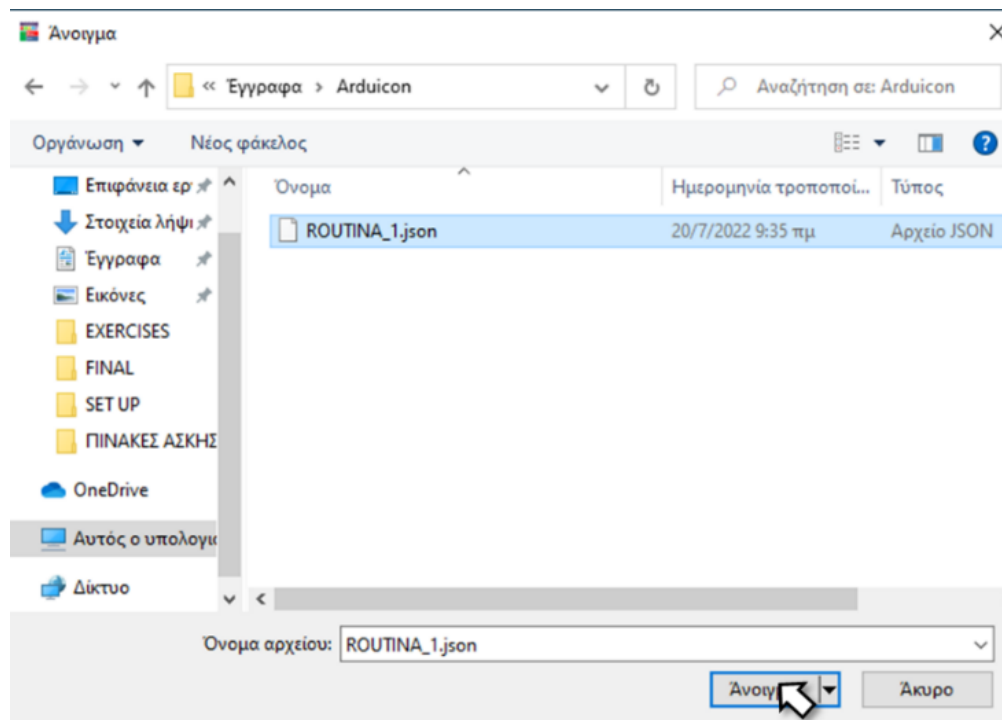
5. Αν θέλετε να επαναφέρετε τη Ρουτίνα που προγραμματίσατε, τότε στην οθόνη εντολών της εφαρμογής πρέπει να επιλέξετε «**Επαναφορά**».



6. Στο αναδυόμενο παράθυρο, ανοίξτε τον φάκελο “**Ardicon**”.



7. Στο αναδυόμενο παράθυρο, επιλέξτε τη Ρουτίνα που θέλετε π.χ. ROUTINA\_1 και επιλέξτε **Άνοιγμα**.



Τα πλακίδια προγραμματισμού της ρουτίνας θα εμφανιστεί στην οθόνη προγραμματισμού της εφαρμογής ARDi:icon.

