STEM IB SUPPORT
BY EPAFOS

Οδηγός Εγκατάστασης και Χρήσης Ard:icon

Επιτελική Δομή ΕΣΠΑ
ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ ΘΡΗΣΚΕΥΜΑΤΩΝ ΚΑΙ ΑΘΛΗΤΙΣΜΟΥ
ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ
Υπουργείο Παιδείας, Θρησκευμάτων και Αθλητισμού
Ελλάδα 2.0
ΕΘΝΙΚΟ ΣΧΕΔΙΟ ΑΝΑΚΑΜΨΗΣ ΚΑΙ ΑΝΘΕΚΤΙΚΟΤΗΤΑΣ
Με τη χρηματοδότηση της Ευρωπαϊκής Ένωσης
NextGenerationEU

Περιεχόμενα

1. Εισαγωγή 2
2. Το λογισμικό ARDICON 2
3. Η εγκατάσταση του λογισμικού ARDICON 3
4. Περιήγηση στην Εφαρμογή ARDICON 5
4.1. Μενού Επιλογών Εφαρμογής 5
4.1.1. Εξήγηση Εικονιδίων Διαχείρισης Προγράμματος 5
4.1.2. Εξήγηση Εικονιδίων Διαχείρισης Κώδικα 7
4.2. Μενού επιλογών Εντολών Λειτουργίας 8
4.3. Μενού επιλογών βασικών μαθηματικών εντολών 8
4.4. Μενού Εντολών Προγραμματισμού 9
5. Διερεύνηση Μενού 10
5.1. Μενού Επιλογής Εντολών Λειτουργίας (Συνοπτικός Πίνακας) 10
5.2. Ανάλυση Εντολών Λειτουργίας 13
5.2.1. Η εντολή "Digital write": 13
5.2.2. Η εντολή "Analog write": 13
5.2.3. Η εντολή "IF" (ή "if statement"): 13
5.2.4. Η εντολή "Else": 14
5.2.5. Η εντολή "For": 14
5.2.6. Η εντολή "While": 15
5.2.7. Η εντολή "delay": 16
5.2.8. Η εντολή "Serial print": 16
5.2.9. Η εντολή "digital Read": 17
5.2.10. Η εντολή "break": 19
5.2.11. Η εντολή "analog Read": 20
5.3. Μενού Επιλογής Βασικών Μαθηματικών Εντολών 20
6. Οδηγός Εκτέλεσης Εργασιών 22
6.1.1. 1η Γραμμή Εντολών 22
6.1.2. 2η Γραμμή Εντολών 23
6.1.3. Εξάσκηση Γραμμών Εντολών 24
6.1.4. Παραγωγή Κώδικα 24
6.1.5. Φόρτωση Κώδικα 25
6.1.6. Έλεγχος Προγράμματος 26
7. Σύνδεση Ελεγκτή και Περιφερειακών μονάδων 27

2106993525 stem-ib-support.gr helpdesk@stem-ib-support.gr

1. Εισαγωγή

Το παρόν υλικό έχει δημιουργηθεί από την ομάδας του HelpDesk και περιέχει οδηγίες εγκατάστασης και βασικούς όρους χρήσης του λογισμικού ARD:icon. Για περισσότερες πληροφορίες, επικοινωνήστε με την εταιρεία PolyTech.

2. Το λογισμικό ARDICON

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

• Διαθέτει ένα διαδραστικό περιβάλλον κωδικοποίησης, που βασίζεται σε εικονίδια και μπλοκ εντολών και κάνει τον προγραμματισμό εύκολο σαν παιχνίδι, 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 διακοπτών, 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.

2106993525 stem-ib-support.gr helpdesk@stem-ib-support.gr

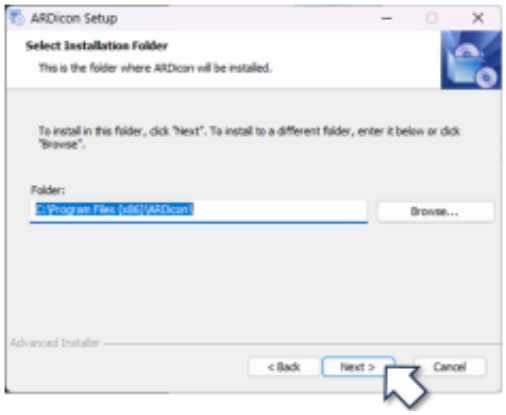
3. Η εγκατάσταση του λογισμικού ARDICON

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

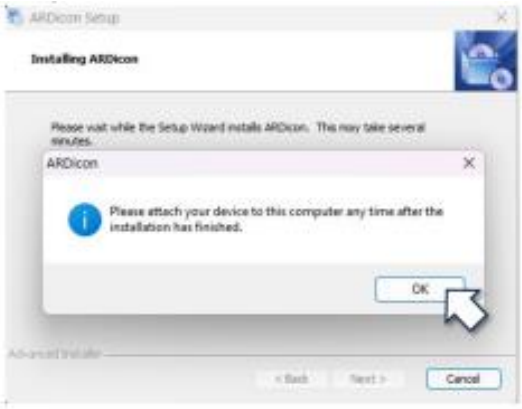
Στο αναδυόμενο παράθυρο, επιλέξτε “Next”

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”.

Στην επιφάνεια του υπολογιστή έχει αυτόματα δημιουργηθεί το εικονίδιο συντόμευσης της εφαρμογής . Με διπλό κλικ πάνω στο εικονίδιο, μπορείτε να ανοίξετε την εφαρμογή “ARD:icon”.

4. Περιήγηση στην Εφαρμογή ARDICON

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

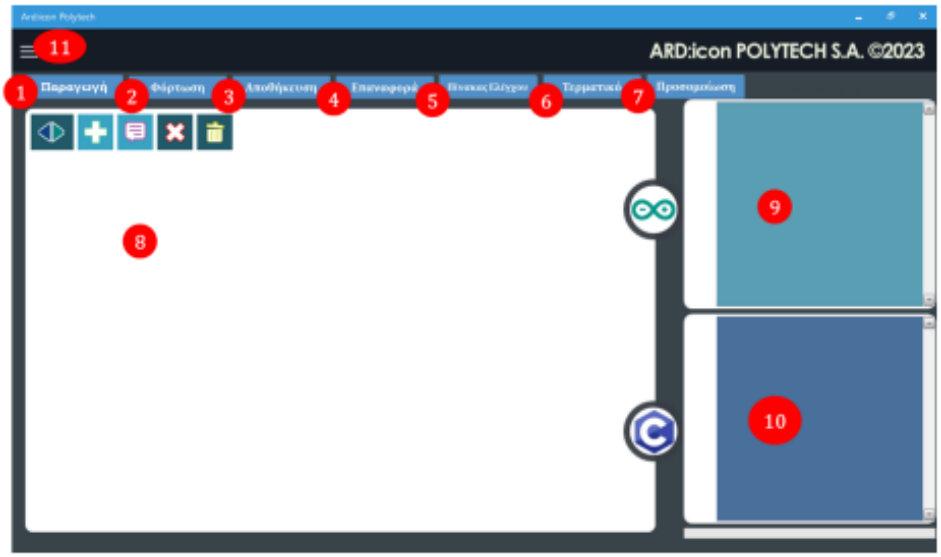
Η αρχική οθόνη της εφαρμογής θα εμφανιστεί στην οθόνη του υπολογιστή σας.

4.1. Μενού Επιλογών Εφαρμογής

4.1.1. Εξήγηση Εικονιδίων Διαχείρισης Προγράμματος

Πάμε να δούμε τα εικονίδια που χρησιμοποιούμε για την διαχείριση του κώδικα στο περιβάλλον μας, δηλαδή την εισαγωγή, τη διαγραφή, την αποθήκευση , την επαλήθευση, την παραγωγή και τη φόρτωση του προγράμματος.

2106993525 stem-ib-support.gr helpdesk@stem-ib-support.gr



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 που προγραμματίσατε, προκειμένου να τον χρησιμοποιήσετε σε κάποια άλλη εργασία.

4. Επαναφορά

Με το κουμπί Επαναφορά, μπορείτε να επαναφέρετε κάποιον κώδικα 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. Στην επόμενη υποενότητα θα δούμε τη λειτουργία καθενός από αυτά ξεχωριστά.

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.

4.1.2. Εξήγηση Εικονιδίων Διαχείρισης Κώδικα

Πάμε να δούμε τα εικονίδια που χρησιμοποιούμε για την εισαγωγή και διαγραφή κώδικα στο περιβάλλον μας.

Α. 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.

Β. Το πλήκτρο σας εισάγει στο περιβάλλον των εντολών 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 εισάγει.

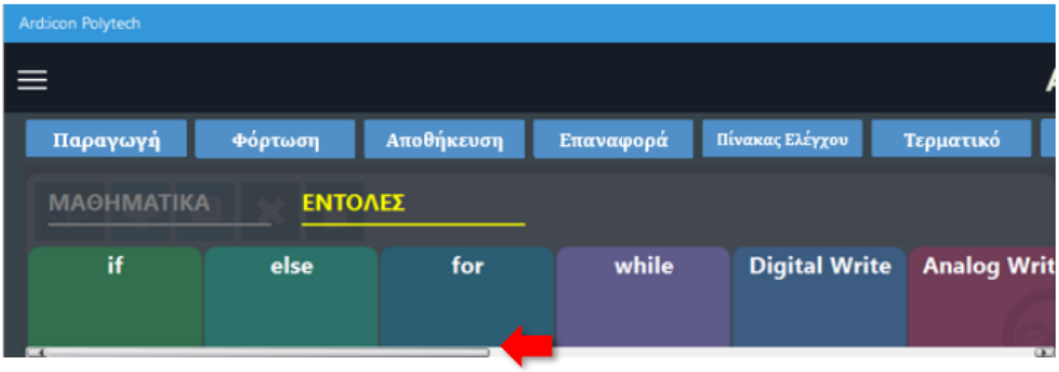
Ε. Το πλήκτρο διαγράφει όλες τις εντολές που έχετε εισάγει.

Για να ξεκινήσετε να προγραμματίζετε ένα έργο, πρώτα επιλέγετε το επίπεδο του κώδικα χρησιμοποιώντας το κουμπί στην περιοχή προγραμματισμού. Πάντα ξεκινάμε μια νέα άσκηση από το επίπεδο ένα.

Στη συνέχεια, πατώντας το κουμπί , εισέρχεστε στο περιβάλλον των εντολών.

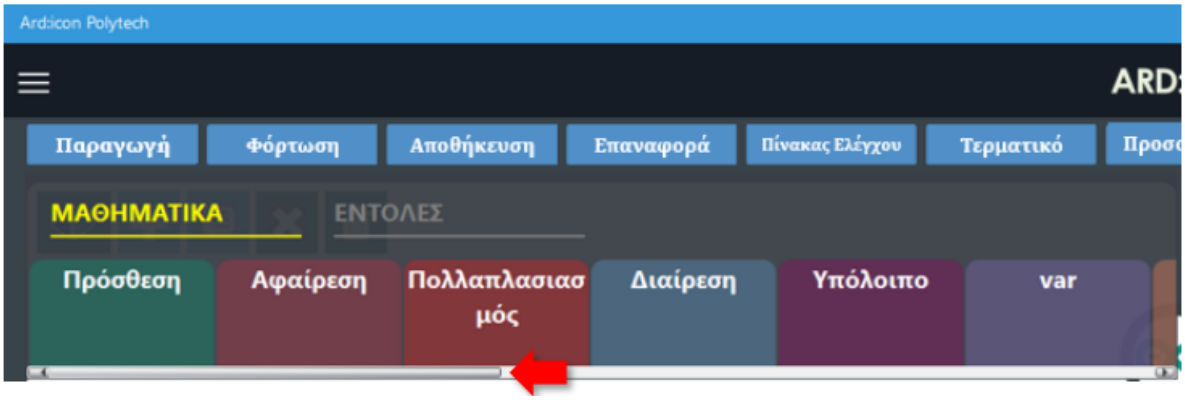
4.2. Μενού επιλογών Εντολών Λειτουργίας

Ας δούμε το μενού των εντολών. Μέσα σε αυτό το μενού θα βρείτε όλες τις βασικές προγραμματιστικές εντολές που υπάρχουν σε μία γλώσσα προγραμματισμού. Πχ. Εάν, Όσο, Μέχρι, ψηφιακό αποτέλεσμα κ.α.



4.3. Μενού επιλογών βασικών μαθηματικών εντολών

Ας δούμε το μενού των μαθηματικών. Μέσα σε αυτό το μενού θα βρείτε όλες τις βασικές προγραμματιστικές εντολές που υπάρχουν σε μία γλώσσα προγραμματισμού και αφορούν μαθηματικές πράξεις. Πχ. πρόσθεση, αφαίρεση, ημίτονο κ.ά.



4.4. Μενού Εντολών Προγραμματισμού

Το μενού επιλογών της εφαρμογής αποτελείται επίσης από:

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**.

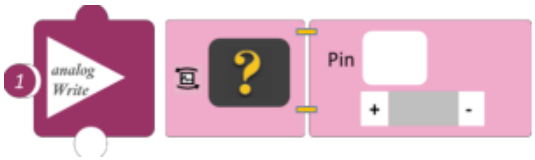
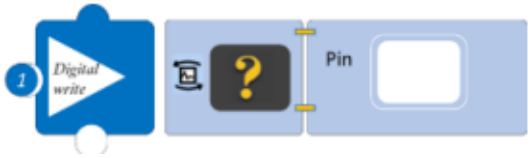
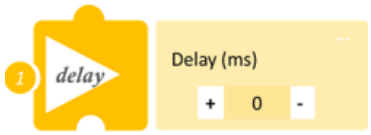
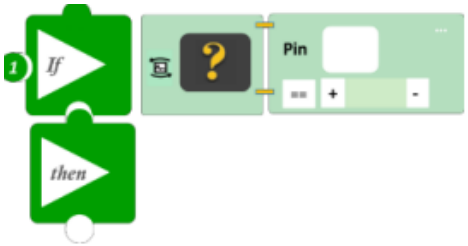
Την ανάλυση των εντολών του συγκεκριμένου μενού στα εγχειρίδια λειτουργίας των ρομπότ.

2106993525 stem-ib-support.gr helpdesk@stem-ib-support.gr

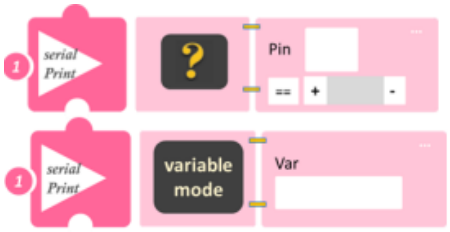
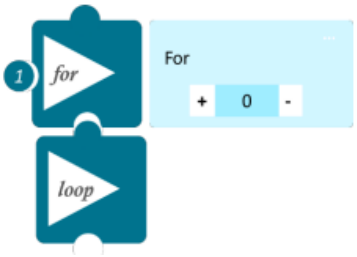
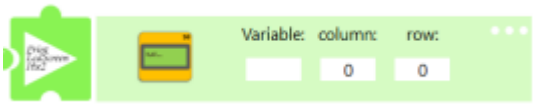
5. Διερεύνηση Μενού

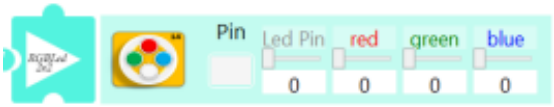
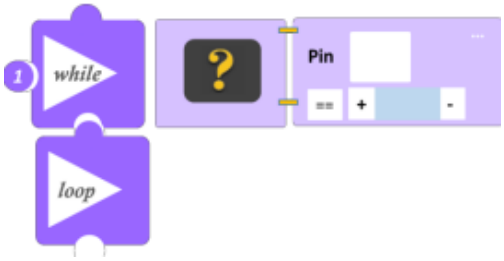
5.1. Μενού Επιλογής Εντολών Λειτουργίας (Συνοπτικός Πίνακας)

Ας δούμε συνοπτικά σε μορφή πίνακα κάποιες από τις βασικές εντολές που θα χρησιμοποιείτε στον κώδικα σας.

Εικονίδιο Επιλογής Εντολής	Περιγραφή	Πλακίδιο Εντολής
Analog Write	Αυτή η εντολή προγραμματίζει τις παραμέτρους της αναλογικής συσκευής εξόδου που συνδέετε στον ελεγκτή ARD:icon. Είναι ενσωματωμένη στις εντολές "if", "while" και "variable".	
Digital Write	Αυτή η εντολή προγραμματίζει τις παραμέτρους της ψηφιακής συσκευής εξόδου που συνδέετε στον ελεγκτή ARD:icon. Είναι ενσωματωμένη στις εντολές "if", "while" και "variable".	
Delay Time	Επιλέγουμε την παύση του προγράμματος για συγκεκριμένο αριθμό χιλιοστών του δευτερολέπτου. Δημιουργεί μία παύση μεταξύ 2 εντολών.	
if	Αυτή η εντολή προγραμματίζει τις παραμέτρους της συνθήκης. Εάν η συνθήκη είναι αληθής, τότε θα εκτελεστεί η εντολή.	

2106993525 stem-ib-support.gr helpdesk@stem-ib-support.gr

else	Συνδέεται ΠΑΝΤΑ με την δήλωση if. Αν η δήλωση if είναι ψευδής, τότε θα εκτελεστεί η εντολή else. Η εντολή else πάντα συνοδεύεται από τουλάχιστον μία εντολή προγραμματισμού αναλογικής ή ψηφιακής εξόδου.	
Serial Print	Η εντολή αυτή χρησιμοποιείται για να 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.	
for	Επιλέγουμε τη συνθήκη βρόχου “for”, για να προγραμματίσουμε πόσες φορές κάτι πρόκειται να επαναληφθεί για μια δεδομένη συνθήκη. Η επανάληψη θα εκτελείται, έως ότου η δεδομένη συνθήκη σταματήσει να ισχύει. Η εντολή “for” πάντα συνοδεύεται τουλάχιστον από μία εντολή προγραμματισμού αναλογικής ή ψηφιακής εξόδου.	
Lcd Οθόνη (AJX04)	Με αυτή τη συσκευή επιλέγουμε τις μεταβλητές που θέλουμε να τυπώνονται στη σειρά 1 και 2 της οθόνης LCD.	

RGB Led 2x2 (DJX13)	Με αυτή τη συσκευή, επιλέγουμε το χρώμα που θέλουμε να ανάβει το κάθε LED της συσκευής RGB LED.	
while	Επιλέγουμε τη συνθήκη βρόχου “while”, για να προγραμματίσουμε ότι κάτι πρόκειται να επαναληφθεί, έως ότου μια δεδομένη συνθήκη σταματήσει να ισχύει και ενώ εκτελείται μια άλλη συνθήκη. 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 (analog write) ή ψηφιακής εξόδου (digital write).	
break	Επιλέγουμε την εντολή break για να τερματίσουμε την if, το βρόχο της while, και τέλος και της for.	

5.2. Ανάλυση Εντολών Λειτουργίας

Ας περιγράψουμε κάποιες από τις βασικές εντολές που θα χρησιμοποιείτε στον κώδικα σας.

5.2.1. 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.

5.2.2. Η εντολή "Analog write":

Χρησιμοποιείται συνήθως σε πλατφόρμες προγραμματισμού π.χ. 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.

5.2.3. 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"

5.2.4. 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.

5.2.5. 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 οποιεσδήποτε 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.

5.2.6. 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) {
 count << 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.

5.2.7. 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 μιλιδευτερόλεπτα (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.

5.2.8. 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.

5.2.9. 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.

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.

5.2.10. 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): (Γλώσσα c)

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

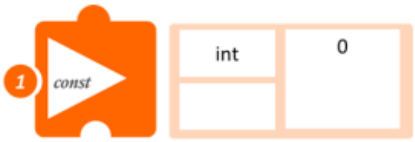
Σε αυτό το παράδειγμα, ο βρόχος for θα τρέξει μέχρι το i να γίνει 5, και τότε η εντολή break θα τερματίσει τον βρόχο.

5.2.11. Η εντολή "analog Read":

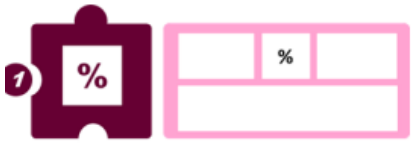
Χρησιμοποιείται συνήθως σε γλώσσες προγραμματισμού, όπως η Arduino C++, για την ανάγνωση τιμών από αναλογικές εισόδους. Μια αναλογική είσοδος παρέχει συνεχείς τιμές σήματος, σε αντίθεση με τις διακριτές τιμές "HIGH" και "LOW", που παρέχονται από τις ψηφιακές εισόδους. Για να χρησιμοποιήσετε την εντολή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

5.3. Μενού Επιλογής Βασικών Μαθηματικών Εντολών

Ας δούμε συνοπτικά σε μορφή πίνακα κάποιες από τις βασικές μαθηματικές εντολές που θα χρησιμοποιείτε στον κώδικα σας.

Εικονίδιο Επιλογής Εντολής	Περιγραφή	Πλακίδιο Εντολής
	Αυτή η εντολή προσδιορίζει μια μεταβλητή με σταθερή τιμή, η οποία δεν μπορεί να αλλάξει κατά τη διάρκεια εκτέλεσης του προγράμματος.	
	Αυτή η εντολή προσδιορίζει μια μεταβλητή που μπορεί να πάρει διαφορετικές τιμές. Οι τιμές τις μεταβλητής Var μπορούν να αλλάζουν και κατά τη ροή του προγράμματος.	
	Αυτή η εντολή + (συν), καθορίζει δύο τελεστές για την παραγωγή ενός αθροίσματος.	
	Αυτή η εντολή –(μείον), καθορίζει δύο τελεστές για τη παραγωγή της διαφοράς του δεύτερου από τον πρώτο.	

2106993525 stem-ib-support.gr helpdesk@stem-ib-support.gr

Πολ/σμός	Αυτή η εντολή * (επί), καθορίζει δύο τελεστές για την παραγωγή του μεταξύ τους γινομένου.	
Διαίρεση	Αυτή η εντολή / (δια), καθορίζει δυο τελεστές για τη παραγωγή του μεταξύ τους αποτελέσματος διαίρεσης.	
Υπόλοιπο Διαίρεσης	Αυτή η εντολή καθορίζει δυο τελεστές για τη παραγωγή του υπολοίπου της μεταξύ τους διαίρεσης.	
Δύναμη	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.	
Τετραγωνική Ρίζα	Αυτή η εντολή υπολογίζει την τριγωνική ρίζα ενός αριθμού.	
Τυχαίος Αριθμός	Αυτή η εντολή η δημιουργεί ψευδοτυχαιους αριθμούς, από ένα εύρος τιμών που έχουμε ήδη προκαθορίσει.	
Συνημίτονο	Η εντολή αυτή υπολογίζει το συνημίτονο μιας γωνίας, που ορίζουμε στο πρόγραμμα.	

2106993525 stem-ib-support.gr helpdesk@stem-ib-support.gr

Ημίτονο	Η εντολή αυτή υπολογίζει το ημίτονο μιας γωνίας, που ορίζουμε στο πρόγραμμα.	
Εφαπτομένη	Η εντολή αυτή υπολογίζει το εφαπτομένη μιας γωνίας, που ορίζουμε στο πρόγραμμα.	

6. Οδηγός Εκτέλεσης Εργασιών

Ας δούμε έναν μικρό οδηγό για τον τρόπο διαχείρισης των δραστηριοτήτων.

Σε κάθε δραστηριότητα που θα προσπαθήσετε να υλοποιήσετε, μπορείτε να χωρίσετε χρωματικά τη σύνταξη που θα έχετε. Παρακάτω, εμφανίζεται ένας πίνακας, ο οποίος αλλάζει ανάλογα με τον προγραμματισμό του κυκλώματος που απαιτείται για την κάθε δραστηριότητα. Χωρίζεται σε 4 διαφορετικά μέρη/στήλες. Ας δούμε μία-μία τι σημαίνουν αυτές οι στήλες για κάθε γραμμή του πίνακα.

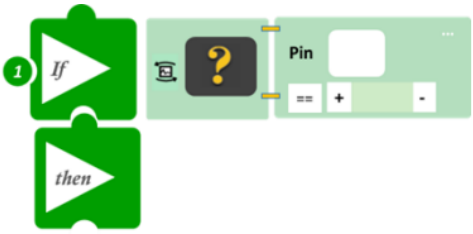
1) Γραμμή Εντολών	2) Εντολή	3) Συσκευή	4) Θύρα / Κατάσταση
1st	"if" / "then"	DJS08	8/ ΚΛΕΙΣΤΟ
2nd	"digital write"	DJX07	9 / ΑΝΟΙΧΤΟ
1st	"if" / "then"	DJS08	8/ ΑΝΟΙΧΤΟ
2nd	"digital write"	DJX07	9 / ΚΛΕΙΣΤΟ

6.1.1. 1η Γραμμή Εντολών

- Στη στήλη «Γραμμή Εντολών», εμφανίζεται η Γραμμή Εντολών της συνάρτησης. Ξεκινάμε πάντα με τη 1η Γραμμή Εντολών. Εάν θέλετε να εισάγετε 2η Γραμμή Εντολών ή να μεταφερθείτε από τη γραμμή 2 στη γραμμή 1, πιέστε το εικονίδιο ,στην αρχική οθόνη του προγράμματος ARD:icon.

- 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 την μπάρα στο κάτω μέρος.

- Επιλέξτε την εντολή (στο παράδειγμά μας ξεκινάμε από την εντολή “if”).

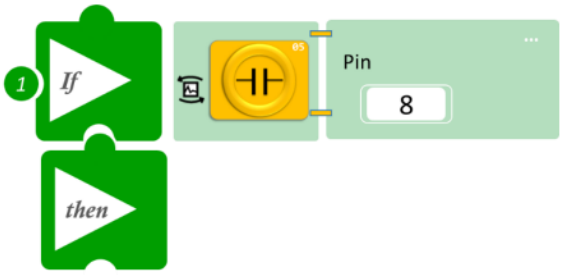


- Στη στήλη «Συσκευή», εμφανίζεται η συσκευή που πρέπει να επιλέξετε επιλέγοντας το εικονίδιο ? πάνω στο πλακίδιο της εντολής, θα εμφανιστεί η λίστα των αναλογικών και ψηφιακών συσκευών. Μπορείτε να μεταβείτε στις υπόλοιπες επιλογές, μετακινώντας την μπάρα στα δεξιά.

  Για παράδειγμα, επιλέξτε τη συσκευή που ζητάει ο πίνακας DJS08 - Διακόπτης Αφής.

- Στη στήλη «Θύρα/Κατάσταση» , εμφανίζονται οι επιμέρους παράμετροι της εντολής. Στο παράδειγμά μας είναι η Θύρα 8 και κατάσταση OFF.

  Προγραμματίστε τις παραμέτρους, όπως ήδη έχετε μάθει.



AN ο Διακόπτης Αφής είναι σε κατάσταση OFF.

Η πρώτη εντολή του πίνακα είναι έτοιμη.

6.1.2. 2η Γραμμή Εντολών

Τώρα, ας προχωρήσουμε στη 2η γραμμή του πίνακα, όπου θα προγραμματίσετε τη συσκευή που θα ελέγχει η συνθήκη της “if” . Προσπαθήστε να κάνετε μόνοι σας τη διαδικασία πριν τη δείτε παρακάτω.

2nd	“digital write”	DJX07	9 / ANOIXTO

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

Στη 2η στήλη είναι η εντολή που πρέπει να επιλέξετε “digital write”.

Στη 3η στήλη, εμφανίζεται η συσκευή που πρέπει να επιλέξετε, στο παράδειγμα μας είναι η DJX07 ή αλλιώς Πράσινο LED.

Στη 4η στήλη μπορείτε να δείτε τις επιμέρους παραμέτρους, δηλαδή Θύρα 9 και κατάσταση ON.

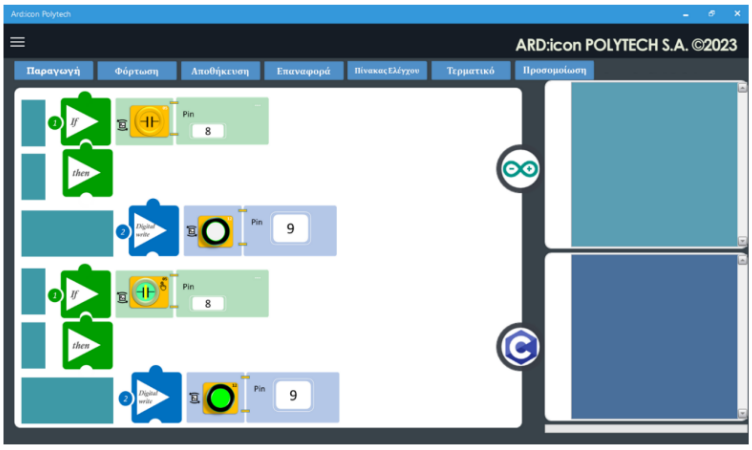
6.1.3. Εξάσκηση Γραμμών Εντολών

Συνεχίστε με τον ίδιο τρόπο να προγραμματίζετε τις υπόλοιπες 2 εντολές του πίνακα στο λογισμικό Ardicon.

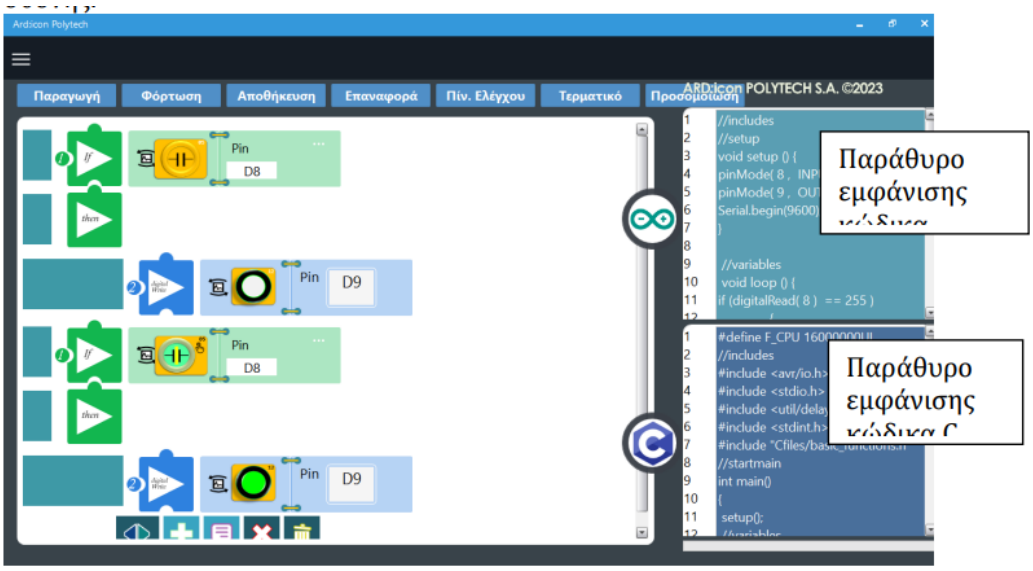
1st	“if” / “then”	DJS08	8/ ΑΝΟΙΧΤΟ
2nd	“digital write”	DJX07	9 / ΚΛΕΙΣΤΟ

6.1.4. Παραγωγή Κώδικα

Εάν έχετε ολοκληρώσει την παραπάνω διαδικασία, ο κώδικας σας θα έχει διαμορφωθεί στην παρακάτω μορφή.



Στη συνέχεια, πατήστε το κουμπί **Παραγωγή** , για να παράγετε τον κώδικα, δηλαδή να τον μετατρέψετε από μπλοκ εντολών σε γλώσσα κατανοητή για την πλακέτα του επεξεργαστή.

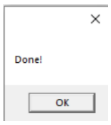


Οι κώδικες σε Arduino και C, θα εμφανιστούν στα δύο παράθυρα δεξιά της αρχική οθόνης.

- Επιλέγοντας πάνω στο εικονίδιο , μπορείτε να ανοίξετε την οθόνη με τον πλήρη κώδικα ARDUINO.
- Επιλέγοντας πάνω στο εικονίδιο ,μπορείτε να ανοίξετε την οθόνη με τον πλήρη κώδικα C.

6.1.5. Φόρτωση Κώδικα

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 σύμφωνα με τις οδηγίες.

Εάν το εικονίδιο 

εμφανιστεί πάνω στην οθόνη της εφαρμογής έχετε φορτώσει με επιτυχία τον κώδικα σας (πρόγραμμα) στον ελεγκτή σας.

Επιλέξτε “OK”.

Εάν το εικονίδιο εμφανιστεί πάνω στην οθόνη της εφαρμογής δεν έχετε φορτώσει με επιτυχία τον κώδικα σας (πρόγραμμα) στον ελεγκτή σας.

Αποσυνδέστε τον ελεγκτή από τον υπολογιστή, επανασυνδέστε τον ελεγκτή και επιλέξτε “OK” και επαναφορτώστε το πρόγραμμα στον ελεγκτή με τον παραπάνω τρόπο.

Μόλις προγραμματίσατε το κύκλωμα 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.

6.1.6. Έλεγχος Προγράμματος

Πάμε να δοκιμάσουμε το κύκλωμα για να δούμε πώς λειτουργεί.

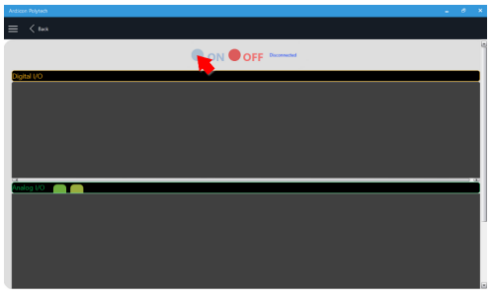
Ελέγχος με Πίνακα Ελέγχου

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

Πίνακας Ελέγχου και Προσομοίωση

Ας δούμε τώρα τα κουμπιά αυτά και πώς μπορούμε με αυτά να δούμε την εφαρμογή του κώδικα.

Πατήστε το κουμπί Πίνακας Ελέγχου



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.

Βλέπετε τις δύο συσκευές που έχετε προγραμματίσει στο κύκλωμα. 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, όπως τα προγραμματίσατε.

Επιλέξτε **OFF** , πριν βγείτε από την οθόνη του πίνακα ελέγχου.

Προσομοίωση

Πατώντας το κουμπί **Προσομοίωση**

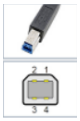
Θα εισέλθετε στην οθόνη του προσομοιωτή. Εδώ, μπορείτε να δείτε τη συνδεσμολογία που έχετε προγραμματίσει, αλλά και την προσομοίωση του κυκλώματος σας, πριν προχωρήσετε στη φόρτωση του προγράμματος. Βλέποντας την προσομοίωση του κυκλώματος, μπορείτε να διορθώσετε και τυχόν σφάλματα που έχει το πρόγραμμα σας.

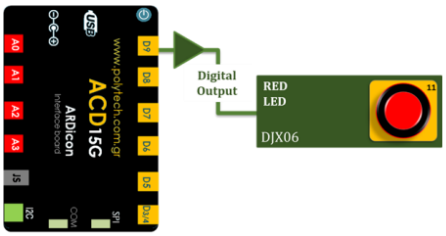
Σε αυτό το περιβάλλον, θα περιηγηθείτε σε έναν εικονικό εκλεκτή, όπου θα μπορέσετε μέσω εικονιδίων να κάνετε το πείραμα και να σιγουρευτείτε για τη λειτουργία πριν την φόρτωση του στον φυσικού ελεγκτή.

7. 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

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/ψηφιακή θύρα του ελεγκτή σύμφωνα με τον κώδικα που θέλετε να υλοποιήσετε. Πχ.

Κόκκινο LED (**DJX06**) στη θύρα **D9**