

micro:bit

Εισαγωγή στο micro:bit

Με εκπαιδευτικές δραστηριότητες στο περιβάλλον MakeCode 1.x

Το Micro:bit είναι ένας μικρός υπολογιστής που φτιάχτηκε για εκπαιδευτικούς σκοπούς από μια ομάδα εταιριών κάτω από την επίβλεψη του Αγγλικού BBC. Το Micro:bit χρησιμοποιεί βιβλιοθήκες ανοικτού κώδικα και απλά μα ισχυρά προγραμματιστικά περιβάλλοντα και κάνει τον προγραμματισμό προσβάσιμο από όλους. Είναι εξοπλισμένο με ένα πλούσιο σύνολο αισθητήρων και συσκευών διασύνδεσης και μπορεί να αποτελέσει μια ιδεώδη εισαγωγή στην επιστήμη των υπολογιστών και την υπολογιστική σκέψη.

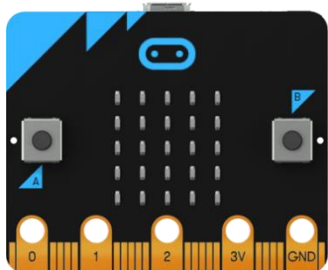
Τάσος Κλείσας
Καλαμάτα, Σεπτέμβριος -Οκτώβριος 2018

Εισαγωγή στο micro:bit

Περιεχόμενα

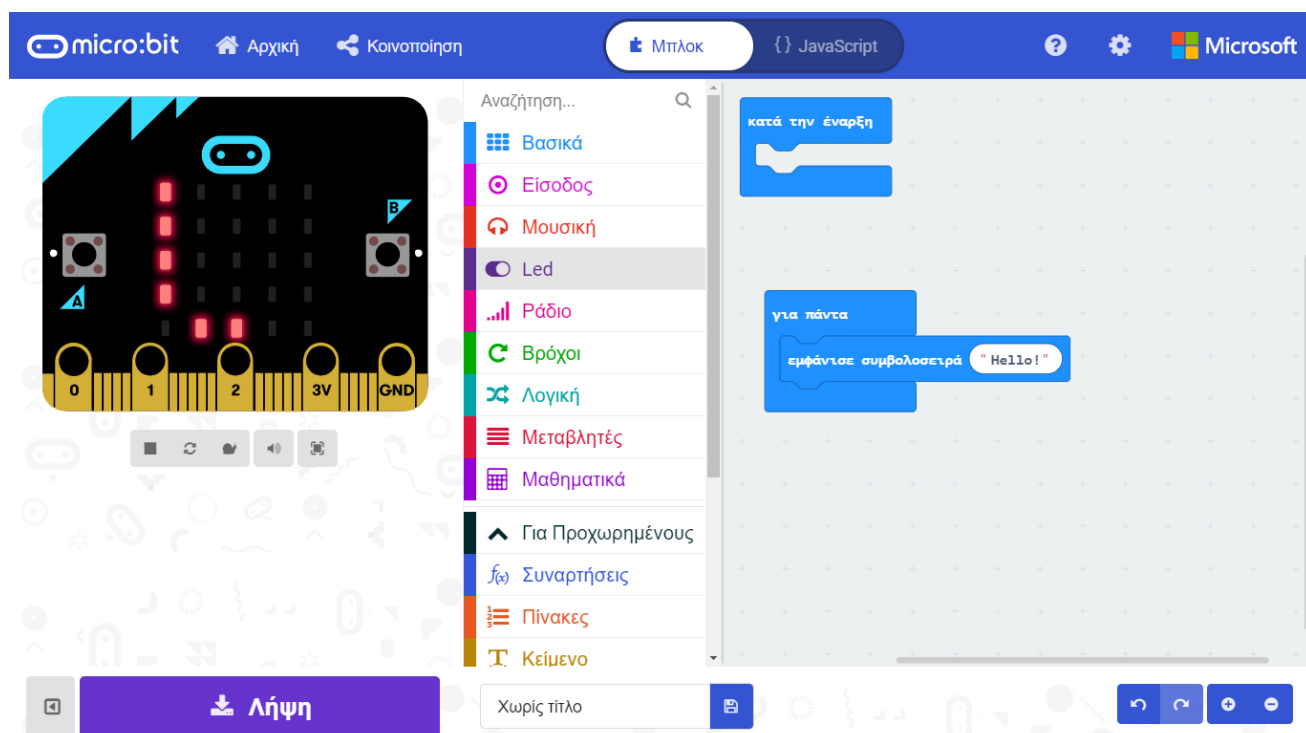
Μια εισαγωγή στο micro:bit.....	3
A) Ξεκινώντας τον προγραμματισμό με το micro:bit	4
Δραστηριότητα A1 – Αναβοσβήνοντας ένα φωτάκι στην οθόνη του micro:bit.....	5
Δραστηριότητα A2 – Δείχνουμε κείμενο στο micro:bit και πατάμε τα κουμπιά του!	15
Δραστηριότητα A3 – Απεικόνιση αριθμών και χρήση των κουμπιών του micro:bit	20
Δραστηριότητα A4 – Χρήση ενός πακέτου για να προσθέσουμε δυνατότητες.....	27
Δραστηριότητα A5 – Το επιταχυνσιόμετρο του micro:bit.....	31
Δραστηριότητα A6 – Ο αισθητήρας φωτός και η πυξίδα του micro:bit	40
Δραστηριότητα A7 – Χρησιμοποιούμε τη λειτουργία Ράδιο του micro:bit για να ανταλλάξουμε μηνύματα!	43
Δραστηριότητα A8 – Συνδέοντας έναν υπερηχητικό αισθητήρα απόστασης	47
Δραστηριότητα A9 – Χρησιμοποιώντας ένα κύκλωμα οδήγησης κινητήρα για να ελέγξουμε ένα κινητήρα με το micro:bit.	52
Δραστηριότητα A10 – Μετρητής βημάτων με το micro:bit.	57

Μια εισαγωγή στο micro:bit



Το Micro:Bit είναι ένα μικροσκοπικός υπολογιστής που φτιάχτηκε από μια ομάδα εταιριών με την καθοδήγηση του Βρετανικού BBC. Το Micro:bit χρησιμοποιεί βιβλιοθήκες ανοιχτού κώδικα και διάφορα προγραμματιστικά περιβάλλοντα που βοηθούν τους μαθητές να μάθουν προγραμματισμό και να δημιουργήσουν χρήσιμα και ανταποδοτικά έργα.

Το micro:bit είναι μια μικρή πλακέτα (όχι μεγαλύτερη από μια πιστωτική κάρτα) που περιλαμβάνει έναν ισχυρό ARM M0+ μικροελεγκτή (microcontroller), 2 φυσικά κουμπιά μαρκαρισμένα A και B, ένα 5x5 πλέγμα από led (led matrix) που μπορεί να απεικονίσει κείμενο και γραφικά, ένα επιταχυνσιόμετρο (accelerometer) και ένα μαγνητόμετρο (magnetometer) που μπορεί να ανιχνεύσει την σχετική θέση και κατεύθυνση της πλακέτας, έναν αισθητήρα φωτός, ένα θερμόμετρο και μια ομάδα από ακροδέκτες στους οποίους μπορούμε να συνδέσουμε διάφορα εξωτερικά εξαρτήματα στο the micro:bit. Το micro:bit έχει 5 στρογγυλούς μεταλλικούς ακροδέκτες. 3 από αυτούς είναι ακροδέκτες εισόδου/εξόδου στους οποίους μπορούμε να συνδέσουμε αισθητήρες, φωτοдиодους (leds), κλπ. Υπάρχει επίσης ένας ακροδέκτης 3V που μπορεί να παρέχει ρεύμα και ένας ακροδέκτης γείωσης (GND).



Το micro:bit μπορεί να προγραμματιστεί με διάφορα προγραμματιστικά περιβάλλοντα (γνωστά και ως ολοκληρωμένα προγράμματα ανάπτυξης, IDEs από το Integrated Development Environments). Από όλα τα περιβάλλοντα, το απλούστερο και πιο προσιτό, κυρίως για πιο μικρά παιδιά είναι το περιβάλλον MakeCode (μπορείτε να το βρείτε στη διεύθυνση <https://makecode.microbit.org/#>). Αυτό θα χρησιμοποιήσουμε στη συνέχεια. Παρόλο που το περιβάλλον MakeCode δουλεύει σε φυλλομετρητές διαδικτύου, μπορεί να δουλέψει και χωρίς σύνδεση δικτύου αν έχουμε μεταγλωττίσει τουλάχιστον ένα έργο.

Το περιβάλλον MakeCode παρέχει επίσης μια εισαγωγική σειρά μαθημάτων αν πατήσουμε το κουμπί «Ξεκινώντας» πάνω δεξιά. Το καλύτερο όμως είναι ότι μπορούμε να ξεκινήσουμε να προγραμματίζουμε χωρίς να έχουμε micro:bit γιατί το περιβάλλον MakeCode παρέχει έναν προσομοιωτή που μπορεί να προσομοιώσει ένα μεγάλο μέρος της λειτουργικότητας του micro:bit. Επιπρόσθετα παρέχουμε έναν μικρό αριθμό από βασικές εισαγωγικές δραστηριότητες έτσι ώστε να μπορέσει κανείς να εξοικειωθεί με το

micro:bit και το περιβάλλον MakeCode. Θα δούμε πως μπορούμε να ξεκινήσουμε να προγραμματίζουμε το micro:bit πως μπορούμε να δημιουργήσουμε τα πρώτα μας ηλεκτρονικά κυκλώματα και να εμπνευστούμε για να φτιάξουμε τα δικά μας καταπληκτικά έργα!

Για κάθε δραστηριότητα, παρέχεται ένα φύλλο εργασίας που περιέχει:

- Μια σύντομη περιγραφή
- Τα απαιτούμενα εξαρτήματα (ηλεκτρονικά) όπου κάτι τέτοιο απαιτείται
- Ένα διάγραμμα του κυκλώματος προς κατασκευή
- Βήμα προς βήμα οδηγίες για την ολοκλήρωση της δραστηριότητας
- Τον κώδικα της δραστηριότητας (ολόκληρο ή μισοτελειωμένο) για να τον δοκιμάσετε
- Σχόλια για το πρόγραμμα και τον τρόπο λειτουργίας του
- Μια λίστα με συνδέσμους στο Internet με περισσότερες πληροφορίες

Αφού τελειώσουμε με αυτή την εισαγωγή θα περάσουμε σε πιο ενδιαφέροντα και πιο «πραγματικά» έργα με το micro:bit.

A) Ξεκινώντας τον προγραμματισμό με το micro:bit

Δραστηριότητα A1 – Αναβοσβήνοντας ένα φωτάκι στην οθόνη του micro:bit

Περίληψη: Σε αυτή τη δραστηριότητα θα μάθουμε πώς να προγραμματίζουμε το micro:bit έτσι ώστε να αναβοσβήσει ένα φωτάκι στην ενσωματωμένη οθόνη 5x5 που διαθέτει. Αρχικά θα φτιάξουμε το πρόγραμμά μας χρησιμοποιώντας τον προσομοιωτή του περιβάλλοντος MakeCode. Θα δούμε επίσης τη βασική δομή ενός προγράμματος και πώς να ανεβάσουμε το πρόγραμμα σε ένα πραγματικό micro:bit έτσι ώστε να το δούμε να δουλεύει πραγματικά!

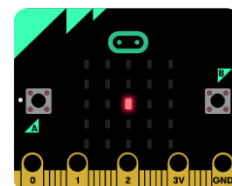
Στόχοι:

- Εξοικείωση με το περιβάλλον προγραμματισμού MakeCode
- Κατανόηση της βασικής δομής ενός προγράμματος
- Δημιουργία του εκτελέσιμου προγράμματος που θα ανεβάσουμε στο micro:bit ώστε να «τρέξει»
- Ανέβασμα του προγράμματος στο micro:bit

Απαιτούμενος εξοπλισμός

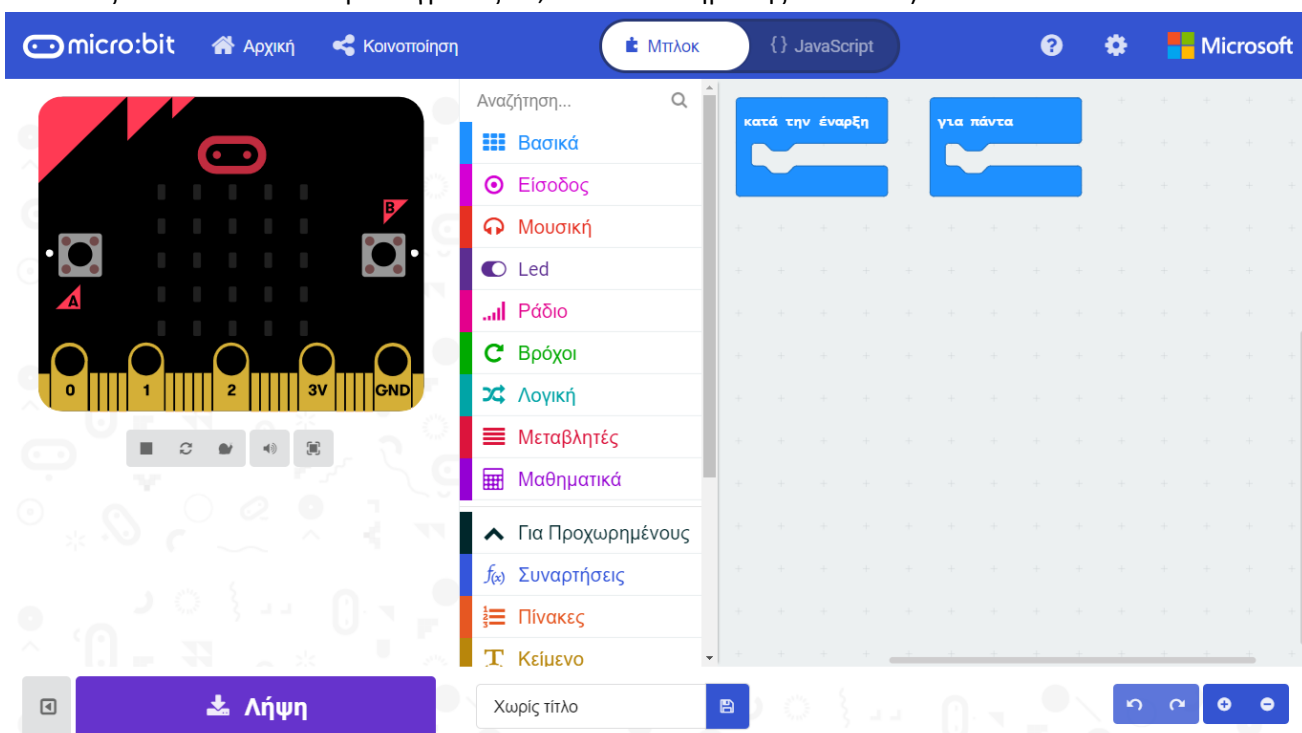
Εξαρτήματα	
1	Micro:bit
2	Micro USB cable (Για να συνδέσουμε το Micro:bit στον υπολογιστή μας)

Το Micro:bit έχει μια οθόνη 5x5 που μπορεί να δείξει κείμενο και απλά γραφικά. Μπορούμε πολύ εύκολα να ανάψουμε και να σβήσουμε τα λαμπάκια της οθόνης. Ας δούμε πως!



Βήματα της δραστηριότητας

1. Επισκεφθείτε την διεύθυνση <https://makecode.microbit.org> . Θα δείτε την ακόλουθη οθόνη. (Αν για κάποιο λόγο η γλώσσα της ιστοσελίδας δεν είναι Ελληνικά μπορείτε να αλλάξετε τη γλώσσα κάνοντας κλικ στο εικονίδιο με το γριναζάκι, δίπλα στο σήμα της Microsoft).



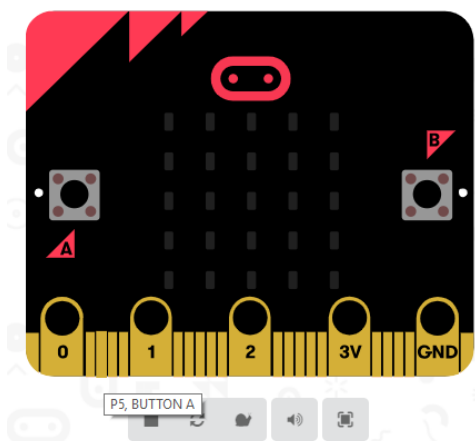
Η οθόνη είναι χωρισμένη σε 5 διαφορετικές περιοχές. Η μπλε ταινία στο πάνω μέρος, με το σήμα του micro:bit logo έχει ένα εικονίδιο με ένα σπιτάκι που γράφει αρχική. Από εκεί μπορούμε να ξεκινήσουμε ένα νέο πρόγραμμα ή να κάνουμε αποθήκευση του προγράμματος που δουλεύουμε.

Δίπλα έχουμε το εικονίδιο της κοινοποίησης. Μπορούμε να το χρησιμοποιήσουμε για να δημοσιεύσουμε το πρόγραμμά μας έτσι ώστε όλοι να μπορούν να το χρησιμοποιήσουν και να το ενσωματώσουν σε μια ιστοσελίδα. Στο κέντρο υπάρχουν δυο κουμπιά

Μπλοκ και JavaScript, το κουμπί Μπλοκ που κάνει το πρόγραμμά μας να απεικονίζεται χρησιμοποιώντας μπλοκ και Javascript που απεικονίζει το πρόγραμμά μας ως κείμενο. Το εικονίδιο

της βοήθειας παρέχει βοήθεια και παραδείγματα για διάφορα θέματα. Το εικονίδιο γρανάζι παρέχει πρόσβαση σε διάφορες ρυθμίσεις. Αν θέλετε να αλλάξετε τη γλώσσα που απεικονίζεται η

ιστοσελίδα πατήστε στο εικονίδιο και επιλέξτε μια από τις διαθέσιμες γλώσσες (μια από τις οποίες είναι και τα ελληνικά).



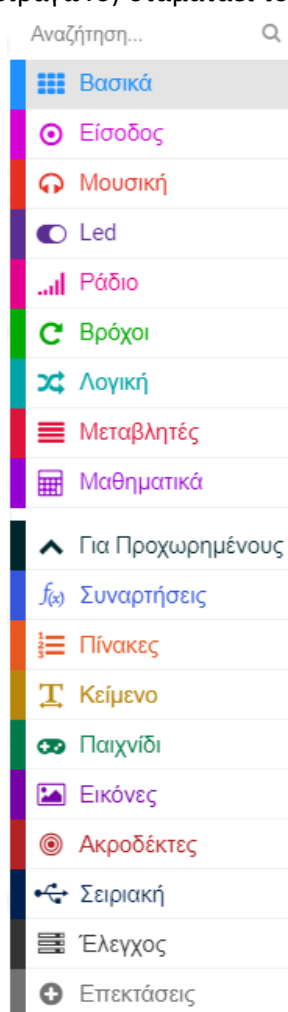
Η κεντρική περιοχή της οθόνης περιέχει τον προσομοιωτή στα αριστερά (μοιάζει με μια εικόνα του micro:bit). Κάτω από το micro:bit υπάρχουν κάποια κουμπιά.

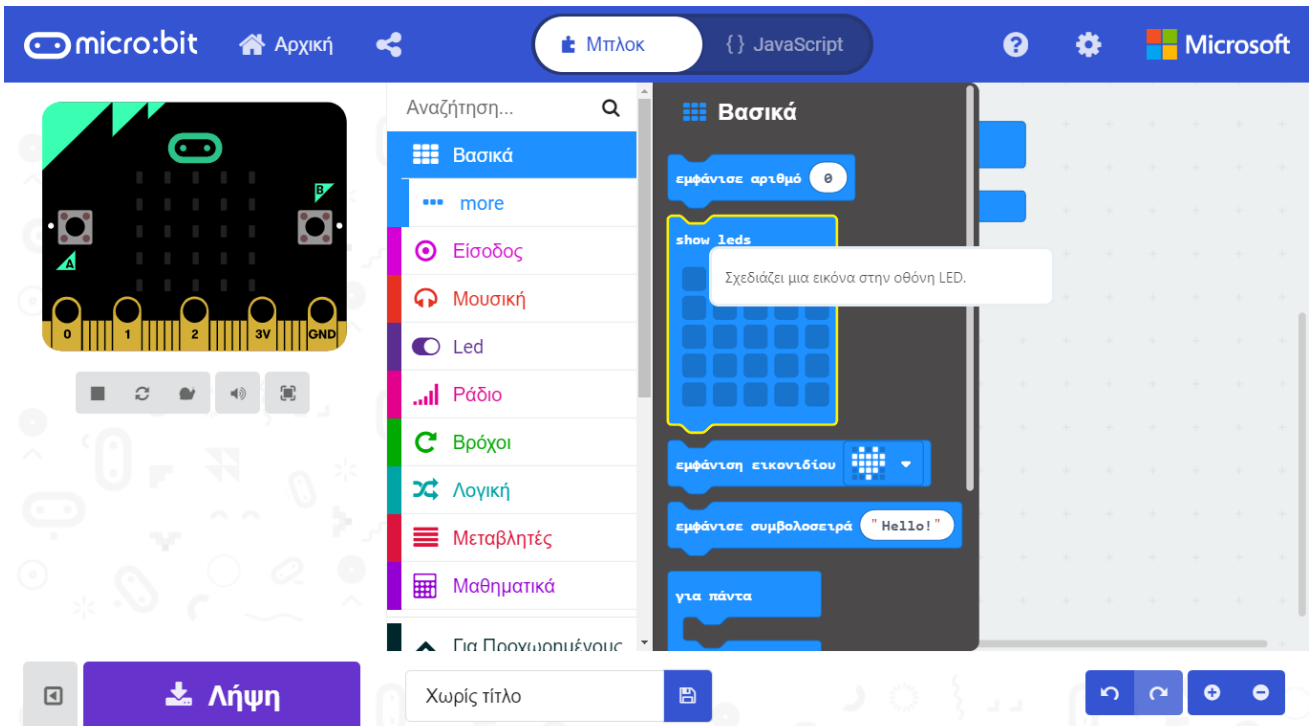
Το πρώτο (που μοιάζει με γκρι τετράγωνο) σταματάει τον προσομοιωτή, το δεύτερο (τα δυο βέλη) επανεκκινεί τον προσομοιωτή, το τρίτο (το σαλιγκαράκι) κάνει τον προσομοιωτή να λειτουργεί αργά (για να μπορέσουμε να βρούμε πιο εύκολα τυχόν λάθη στο πρόγραμμά μας), το

τέταρτο (το ηχείο) απενεργοποιεί / ενεργοποιεί τον ήχο και το πέμπτο (τέσσερα βελόνια που ξεκινάνε από το κέντρο) βάζει τον προσομοιωτή σε πλήρη οθόνη.

Δίπλα στον προσομοιωτή και προς τα αριστερά μπορούμε να δούμε την περιοχή με τα μπλοκ.

Η περιοχή με τα μπλοκ αποτελείται από μια σειρά από κουμπιά τοποθετημένα κάθετα. Κάθε κουμπί έχει ένα εικονίδιο και μια λέξη και είναι ζωγραφισμένο με διαφορετικό χρώμα. Κάθε κουμπί αντιστοιχεί σε μια ομάδα εντολών ομαδοποιημένων με βάση τη λειτουργία τους. Όταν κάνουμε κλικ σε ένα κουμπί, αυτό μεγαλώνει αποκαλύπτοντας μια σειρά από μπλοκ, όπως στην επόμενη εικόνα:





Το μπλοκ show leds μας επιτρέπει να ανάψουμε και να σβήσουμε φωτάκια στην οθόνη του micro:bit και είναι το μόνο μπλοκ που θα χρειαστεί να προσθέσουμε στην περιοχή προγραμματισμού για αυτή τη δραστηριότητα.

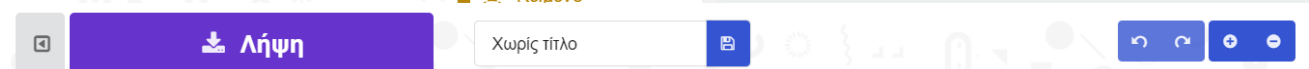
Στα δεξιά, δίπλα από την περιοχή με τα μπλοκ, έχουμε την περιοχή προγραμματισμού. Αρχικά η περιοχή προγραμματισμού έχει δυο μπλοκ που έχουν τους τίτλους «κατά την έναρξη» και «για πάντα».



Αν βάλουμε ένα ή περισσότερα μπλοκ μέσα στο μπλοκ «κατά την έναρξη», τα μπλοκ θα εκτελεστούν μια φορά, μόλις δώσουμε ρεύμα στο micro:bit, ή όταν έχει ήδη ρεύμα και πατάμε το πλήκτρο “reset”. Αν βάλουμε ένα ή περισσότερα μπλοκ μέσα στο μπλοκ «για πάντα», θα εκτελούνται συνεχώς από το πιο πάνω στο πιο κάτω. Η εκτέλεση θα σταματήσει μόνο αν βγάλουμε το micro:bit από το ρεύμα.

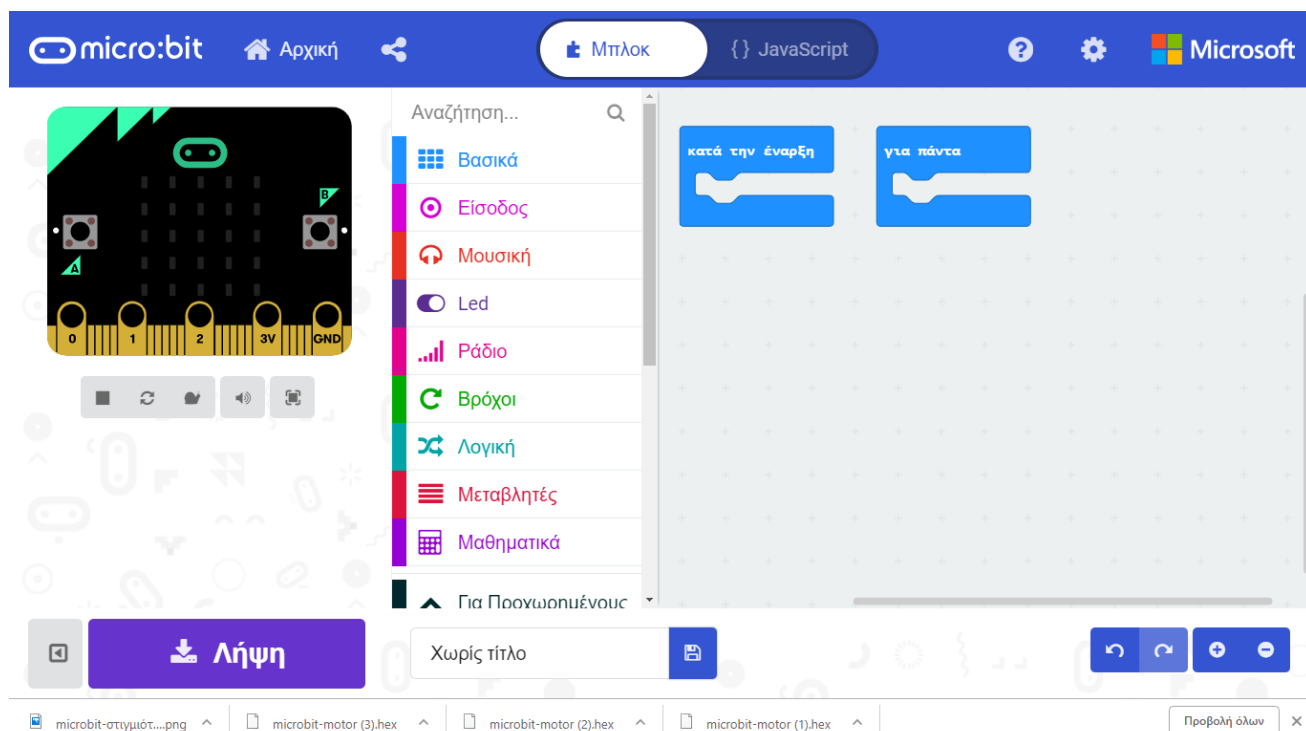
Σημείωση #1: Τα μπλοκ «κατά την έναρξη» και «για πάντα» δεν είναι υποχρεωτικά. Αυτό σημαίνει ότι σε ένα πρόγραμμα, ένα από τα δύο ή και τα δύο μπορεί να απουσιάζουν.

Σημείωση #2: Τα μπλοκ «κατά την έναρξη» και «για πάντα» υλοποιούν λειτουργίες που υπάρχουν και σε άλλες πλατφόρμες όπως είναι το scratch και το Arduino. Στο Arduino για παράδειγμα λέγονται “setup” και “loop” αντίστοιχα.

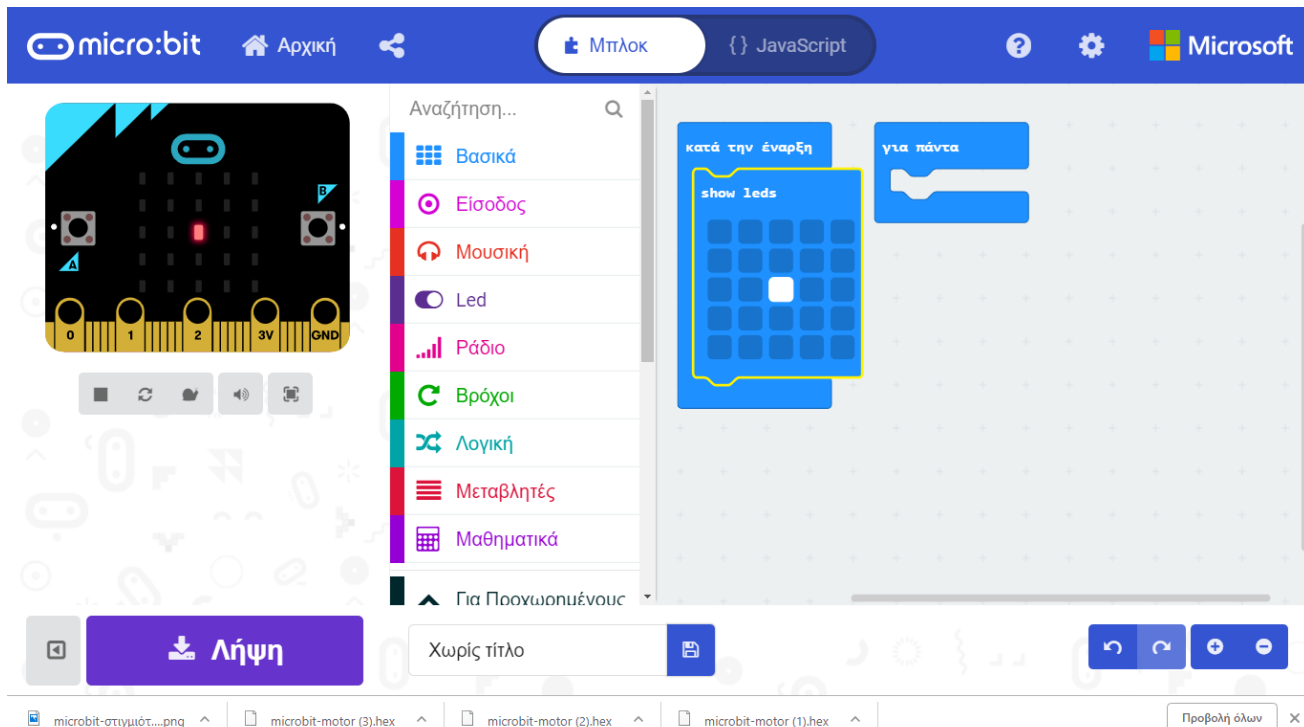


Δίπλα ακριβώς έχουμε ένα κουτί κειμένου που μπορούμε να χρησιμοποιήσουμε για να δώσουμε ένα όνομα στο έργο μας (αρχικά εμφανίζεται ως «Χωρίς τίτλο»). Ακόμη πιο δεξιά έχουμε τα πλήκτρα αναίρεσης /

επανάληψης ενέργειας και τα κουμπιά + και – που μας επιτρέπουν να μεγεθύνουμε και να μικρύνουμε την περιοχή προγραμματισμού.



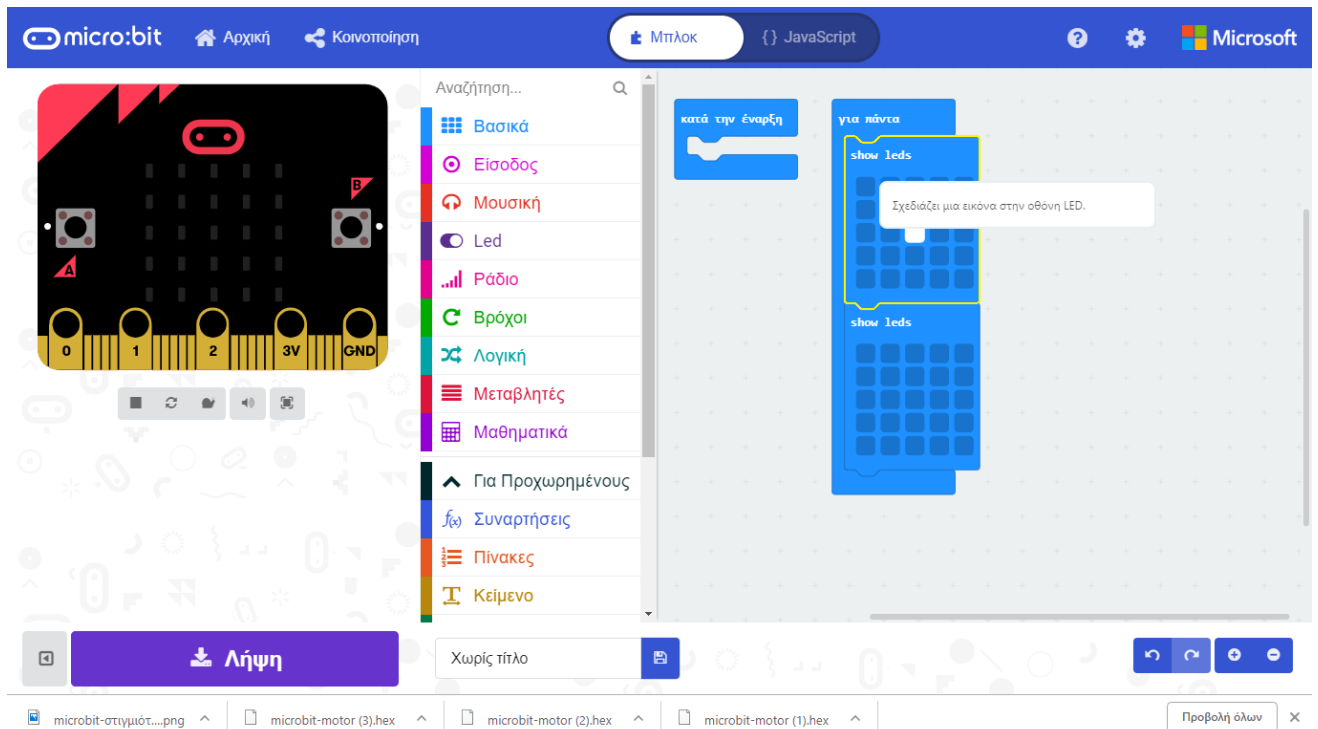
2. Τώρα ας τραβήξουμε το μπλοκ “show leds” (που βρίσκεται στα «Βασικά» μπλοκ) μέσα στο μπλοκ «κατά την έναρξη» μέχρι να κουμπώσει μέσα στο μπλοκ «κατά την έναρξη». Στη συνέχεια κάνουμε κλικ με το ποντίκι μας στο κεντρικό τετραγωνάκι της show leds έτσι ώστε να κοκκινήσει.



Όπως μπορούμε να δούμε το κεντρικό λαμπάκι της οθόνης 5x5 ανάβει! Συγχαρητήρια, μόλις φτιάξαμε το πρώτο μας πρόγραμμα στο micro:bit!

3. Όμως ένα φωτάκι που ανάβει δεν είναι πολύ ενδιαφέρον. Τι είναι καλύτερο από ένα αναμμένο φωτάκι; Το βρήκατε, **ένα φωτάκι που αναβοσβήνει!** Έτσι, για να κάνουμε το φωτάκι να αναβοσβήνει

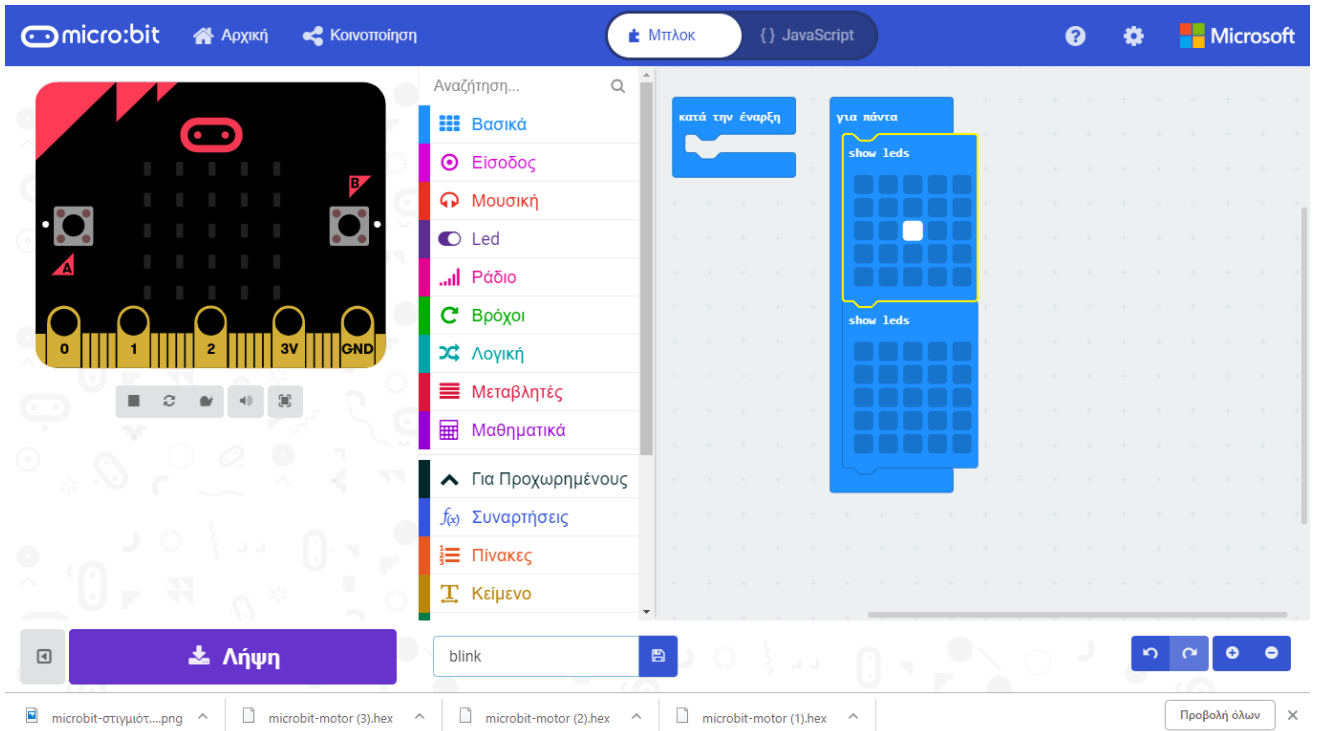
χρειάζεται να εναλλασσόμαστε μεταξύ δύο καταστάσεων: Αναμμένο φως και σβηστό φως. Μα το μπλοκ «κατά την έναρξη» δεν είναι κατάλληλο για αυτό το σκοπό. Θυμηθείτε: το μπλοκ «κατά την έναρξη» εκτελείται μόνο μία φορά όταν δώσουμε ρεύμα ή κάνουμε reset στο micro:bit, έτσι αν θέλουμε να έχουμε ένα φωτάκι που αναβοσβήνει πρέπει να χρησιμοποιήσουμε το μπλοκ «για πάντα». Όπως ειπώθηκε προηγουμένως, το μπλοκ «για πάντα» εκτελεί τα μπλοκ που περιέχει με σειρά από πάνω προς τα κάτω και όταν τελειώνει με το τελευταίο, ξαναρχίζει πάλι από το πρώτο. Έτσι για να έχουμε ένα φωτάκι που αναβοσβήνει, μπορούμε να χρησιμοποιήσουμε το μπλοκ “show leds” δύο φορές, μέσα στο μπλοκ «για πάντα», όπως στην επόμενη εικόνα:



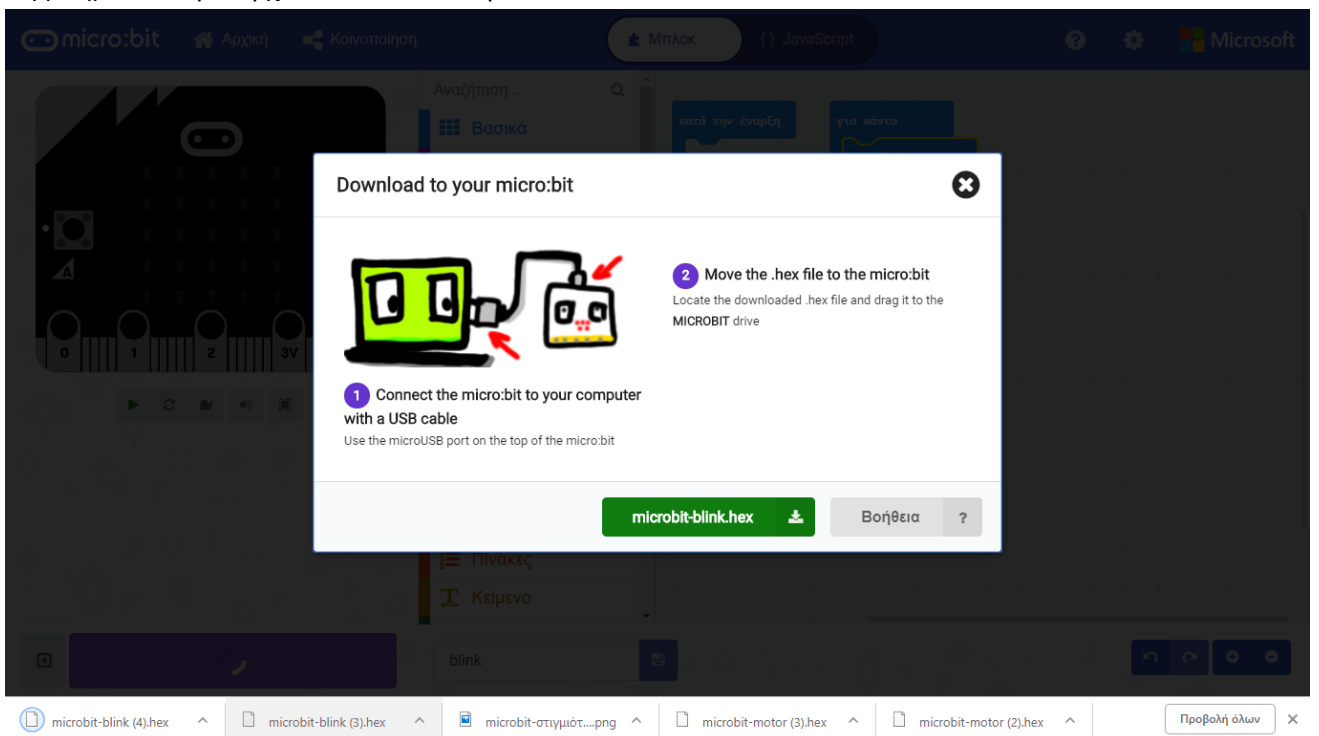
Συμβουλή: Ένας σύντομος τρόπος να τοποθετήσουμε 2 ή περισσότερα όμοια μπλοκ στην περιοχή προγραμματισμού είναι να χρησιμοποιήσουμε την επιλογή «Αντίγραφο» που εμφανίζεται αν κάνουμε δεξί κλικ πάνω σε ένα μπλοκ. Αφού τοποθετήσουμε το αρχικό μπλοκ στην περιοχή προγραμματισμού, κάνουμε δεξί κλικ στο μπλοκ που έχουμε βάλει και διαλέγουμε «Αντίγραφο» από το αναδυόμενο μενού. Ένα αντίγραφο του μπλοκ εμφανίζεται! Μπορούμε να επαναλάβουμε αυτή τη διαδικασία όσες φορές θέλουμε!

Τοποθετήσαμε 2 μπλοκ “show leds” και ανάψαμε το κεντρικό λαμπάκι στο πρώτο “show leds” μπλοκ. Το δεύτερο “show leds” μπλοκ δεν έχει αναμμένα λαμπάκια. Έτσι τώρα αν κοιτάξουμε τον προσομοιωτή θα δούμε το κεντρικό λαμπάκι να αναβοσβήνει.

4. Τώρα μπορούμε να ανεβάσουμε το πρόγραμμα που μόλις φτιάξαμε σε ένα πραγματικό micro:bit. Το πρώτο βήμα είναι να ονομάσουμε το πρόγραμμά μας. Κάνουμε κλικ στο Χωρίς τίτλο στο κάτω μέρος της οθόνης και πληκτρολογούμε ένα όνομα. Ας ονομάσουμε το πρόγραμμά μας blink.

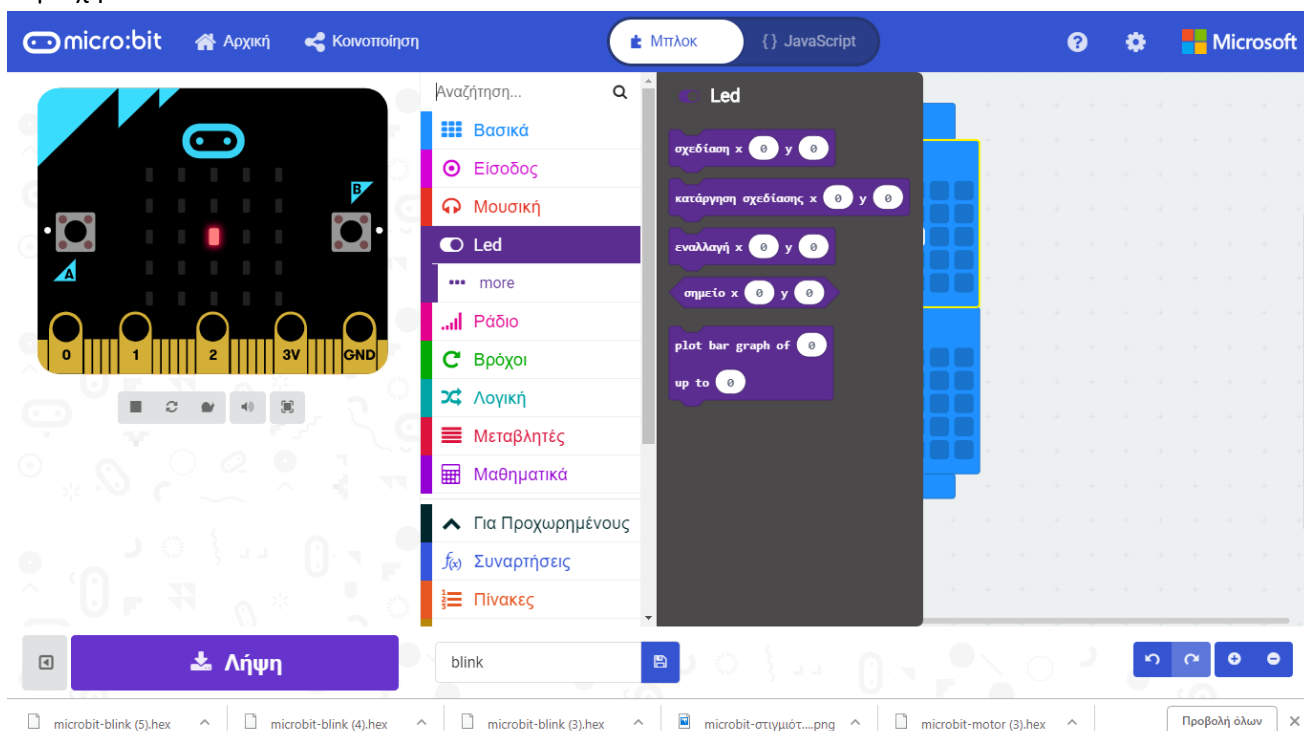


5. Αν κάνουμε κλικ στο εικονίδιο της δισκέτας ή στο πλήκτρο Λήψη το πρόγραμμα θα κατέβει στην προκαθορισμένη περιοχή αρχείων του φυλλομετρητή σας (συνήθως ένας φάκελος με το όνομα Λήψεις). Το όνομα αρχείου που θα κατέβει θα είναι “microbit-blink.hex”.



6. Για να μεταφέρουμε το πρόγραμμα από τον υπολογιστή μας στο micro:bit θα χρειαστούμε ένα micro:bit και ένα καλώδιο τύπου micro usb, σαν κι αυτά που χρησιμοποιούμε για να συνδέσουμε το tablet ή το κινητό μας τηλέφωνο με τον υπολογιστή. Αν συνδέσουμε το micro:bit σε μια θύρα usb του υπολογιστή μας, το micro:bit εμφανίζεται ως αφαιρούμενος δίσκος. Το μόνο που πρέπει να κάνουμε είναι να σύρουμε το αρχείο “microbit-blink.hex” στον αφαιρούμενο δίσκο micro:bit. Όταν τελειώσει η διαδικασία αντιγραφής του προγράμματος, το πρόγραμμά μας εκτελείται άμεσα στο micro:bit.! Πρέπει να δούμε το κεντρικό λαμπάκι να αναβοσβήνει!

7. Όμως, το μπλοκ “show leds” δεν είναι πολύ γρήγορο και δεν συνιστάται αν θέλουμε να έχουμε κινούμενα γραφικά. Μια εναλλακτική είναι να χρησιμοποιήσουμε τα μπλοκ που υπάρχουν στην περιοχή “Led”.



Λοιπόν, τι κάνει το κάθε μπλοκ;

Το **σχεδίαση x y** ανάβει ένα λαμπάκι που βρίσκεται στη στήλη x και τη γραμμή y. Οι γραμμές και οι στήλες παίρνουν τιμές από 0 έως 4. Έτσι για παράδειγμα, η «σχεδίαση 0 0» ανάβει το φωτάκι στην πάνω αριστερή γωνία, η «σχεδίαση 2 2» ανάβει το κεντρικό λαμπάκι και η «σχεδίαση 4 4» φωτίζει το φωτάκι κάτω δεξιά.

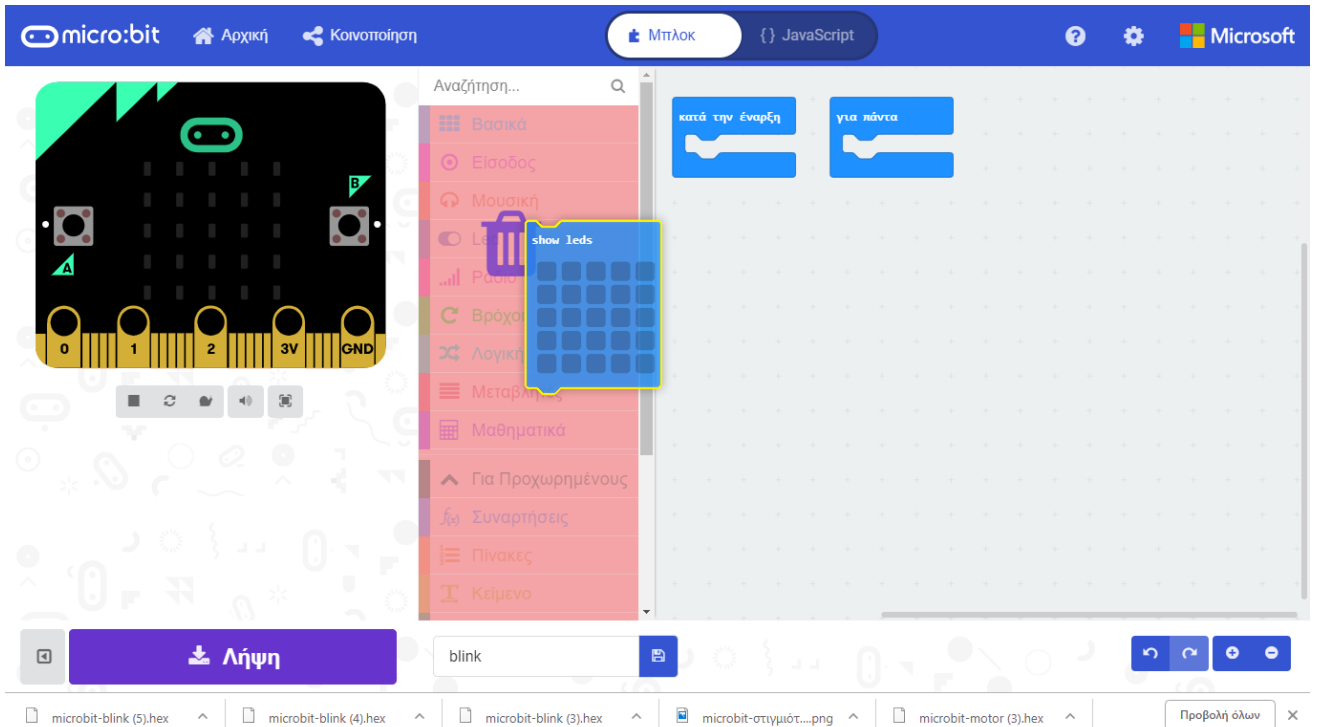
Το **κατάργηση σχεδίασης x y** σβήνει ένα φως που βρίσκεται στη στήλη x και γραμμή y.

Το **εναλλαγή x y** αντιστρέφει την κατάσταση της λάμπας. Αν είναι αναμμένη, τη σβήνει και αντίστροφα.

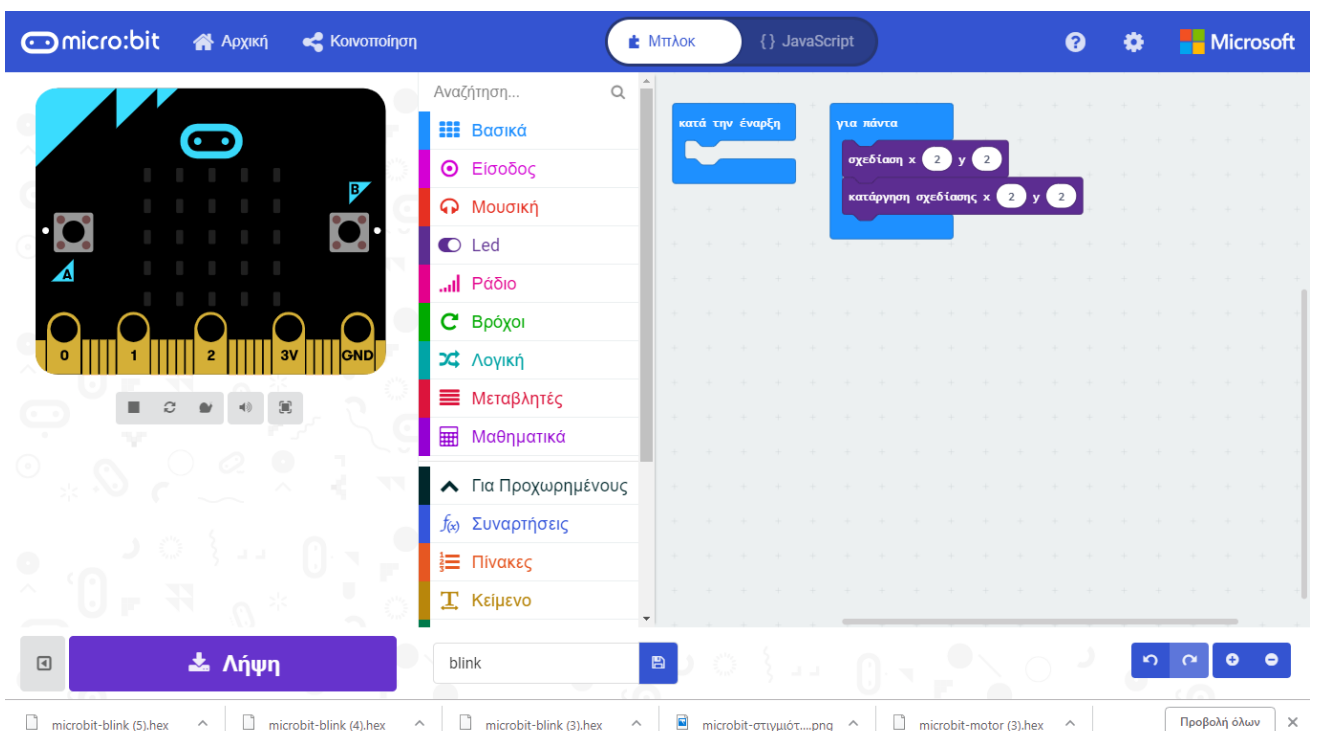
Το **σημείο x y** παίρνει την κατάσταση της λάμπας που βρίσκεται στη στήλη x και την γραμμή y. Αν είναι αναμμένη επιστρέφει true. Αν είναι εκτός λειτουργίας επιστρέφει false.

Το **plot bar graph of up to** σχεδιάζει ένα γράφημα της τιμής από 0 που αντιστοιχεί σε μια κενή οθόνη έως ένα τετράγωνο που αντιστοιχεί στη μέγιστη τιμή.

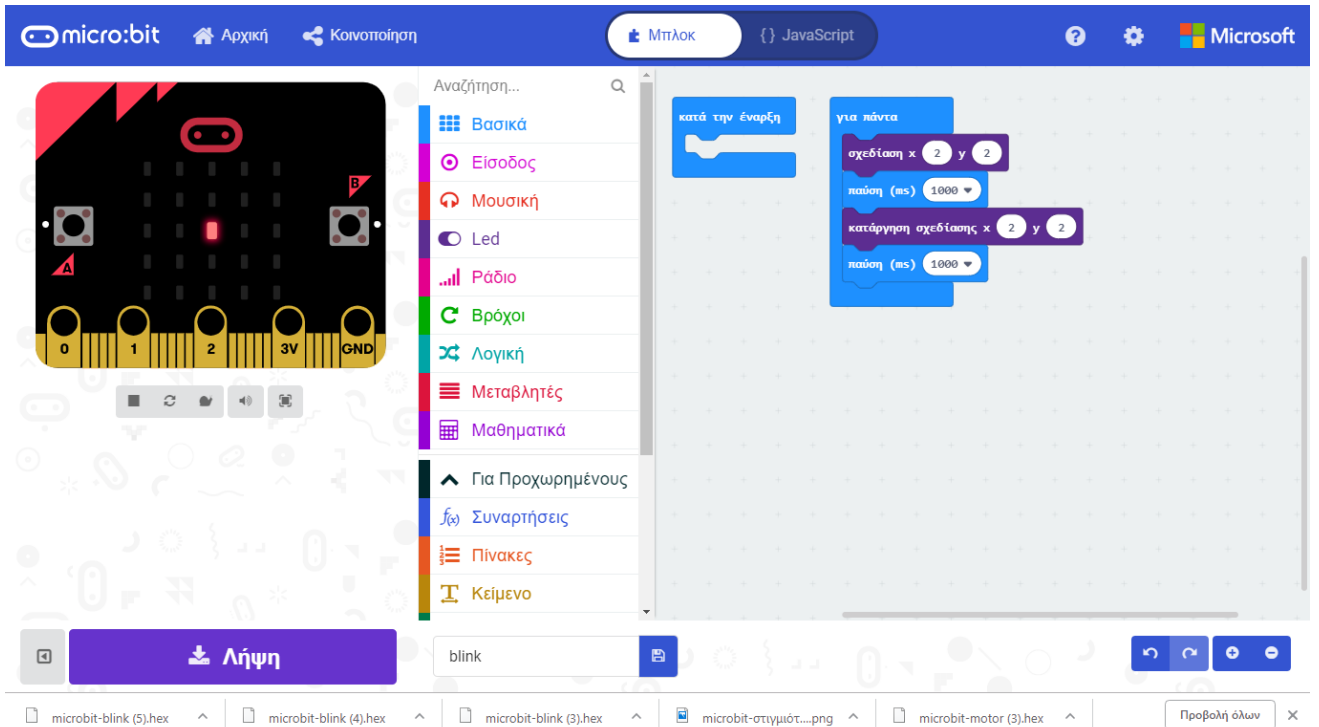
8. Έτσι τώρα μπορούμε να αλλάξουμε το πρόγραμμά μας. Πρώτα όμως ας ξεφορτωθούμε τα μπλοκ “show leds” σύροντας τα στον κάδο.



Στη συνέχεια αντικαθιστούμε τη “show leds” χρησιμοποιώντας τα μπλοκ «σχεδίαση» και «κατάργηση σχεδίασης» όπως στην ακόλουθη εικόνα :



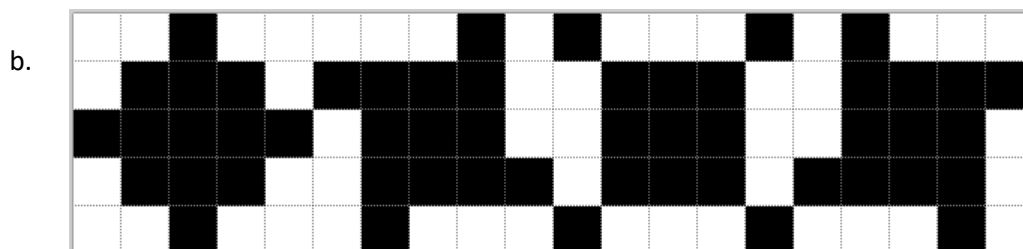
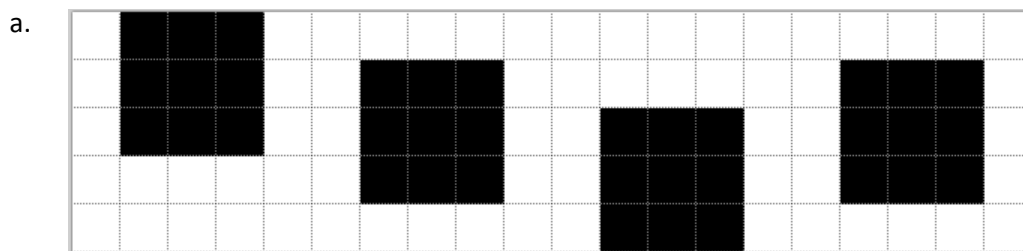
9. Αν δοκιμάσετε το πρόγραμμα θα δείτε ότι κάτι πάει στραβά γιατί δε βλέπουμε φωτάκι να αναβοσβήνει στον προσομοιωτή. Που είναι το πρόβλημα; Φαίνεται ότι το φως αναβοσβήνει τόσο γρήγορα που δεν προλαβαίνουμε να το δούμε. Έτσι ας προσθέσουμε μια καθυστέρηση μεταξύ των μπλοκ «σχεδίαση» και «κατάργηση σχεδίασης» Αυτό θα κάνει το micro:bit να περιμένει λίγο πριν να αλλάξει κατάσταση στο φωτάκι. Μπορούμε να κάνουμε το micro:bit να περιμένει χρησιμοποιώντας το μπλοκ «παύση (ms)» που βρίσκεται στην ομάδα μπλοκ «Βασικά»



Αφού προσθέσαμε αυτό το μπλοκ το λαμπάκι αναβοσβήνει μια χαρά.

Ασκήσεις:

1. Ανεβάστε το πρόγραμμα «blink» με τα μπλοκ «σχεδίαση»/«κατάργηση σχεδίασης» και «παύση (ms)» και κάντε το κεντρικό λαμπάκι της οθόνης του micro:bit να ανάψει για ένα δευτερόλεπτο και στη συνέχεια να σβήσει για ένα δευτερόλεπτο. (Ίσως χρειαστεί να προσθέσετε ένα ακόμη μπλοκ «παύση (ms)» για να το πετύχετε!)
2. Φτιάξτε μια σειρά από κινούμενες εικόνες χρησιμοποιώντας την “show leds” και ανεβάστε τη στο micro:bit. Οι κινούμενες εικόνες μπορούν να είναι κάποιες από τις επόμενες:



3. Φτιάξτε τις δικές σας κινούμενες εικόνες και εντυπωσιάστε!
4. Ανεβάστε τις κινούμενες εικόνες που φτιάξατε στο micro:bit και δείτε πως φαίνονται σε μια πραγματική συσκευή.

Links	
1	https://makecode.microbit.org/ Περιβάλλον ανάπτυξης εφαρμογών για το micro:bit
2	https://el.wikipedia.org/wiki/%CE%94%CE%AF%CE%BF%CE%B4%CE%BF%CF%82_%CE%B5%CE%BA%CF%80%CE%BF%CE%BC%CF%80%CE%AE%CF%82_%CF%86%CF%89%CF%84%CF%8C%CF%82 Πως δουλεύουν τα λαμπάκια τύπου LED;

Δραστηριότητα A2 – Δείχνουμε κείμενο στο micro:bit και πατάμε τα κουμπιά του!

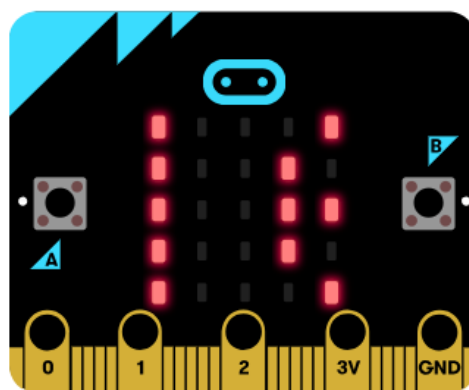
Περίληψη: Σε αυτή τη δραστηριότητα θα δούμε πως να προγραμματίζουμε το micro:bit ώστε να δείχνει κείμενο στην ενσωματωμένη του οθόνη 5x5 και θα δούμε πως μπορούμε να χρησιμοποιήσουμε τα κουμπιά A και B για να επιλέξουμε το κείμενο που θα εμφανίζεται!

Στόχοι:

- Βλέπουμε πώς να απεικονίσουμε κείμενο στην οθόνη του micro:bit
- Χρησιμοποιούμε τα κουμπιά A και B ώστε να δείχνουμε διαφορετικό κείμενο ανάλογα με το κουμπί που πατήθηκε

Το micro:bit έχει μια οθόνη 5x5 εικονοστοιχείων (pixels). Κάθε εικονοστοιχείο είναι στην ουσία μια φωτοδίοδος (Light Emitting Diode ή LED). Η οθόνη μπορεί να απεικονίσει κείμενο και απλά γραφικά. Αν το κείμενο που θέλουμε να απεικονίσουμε είναι μεγαλύτερο από ένα χαρακτήρα, το κείμενο «κυλάει» από τα δεξιά προς τα αριστερά μέχρι να απεικονιστούν όλοι οι χαρακτήρες.

Το micro:bit έχει 2 φυσικά πλήκτρα που ονομάζονται A και B. Μπορούμε να χρησιμοποιήσουμε μπλοκ που ενεργοποιούνται από κάποια ενέργεια (μπλοκ ενεργειών) για να προσδιορίσουμε αν ο χρήστης πάτησε το πλήκτρο A, το πλήκτρο B ή και τα δύο πλήκτρα ταυτόχρονα και να εκτελέσουμε διαφορετικές λειτουργίες βασισμένοι σε αυτό.



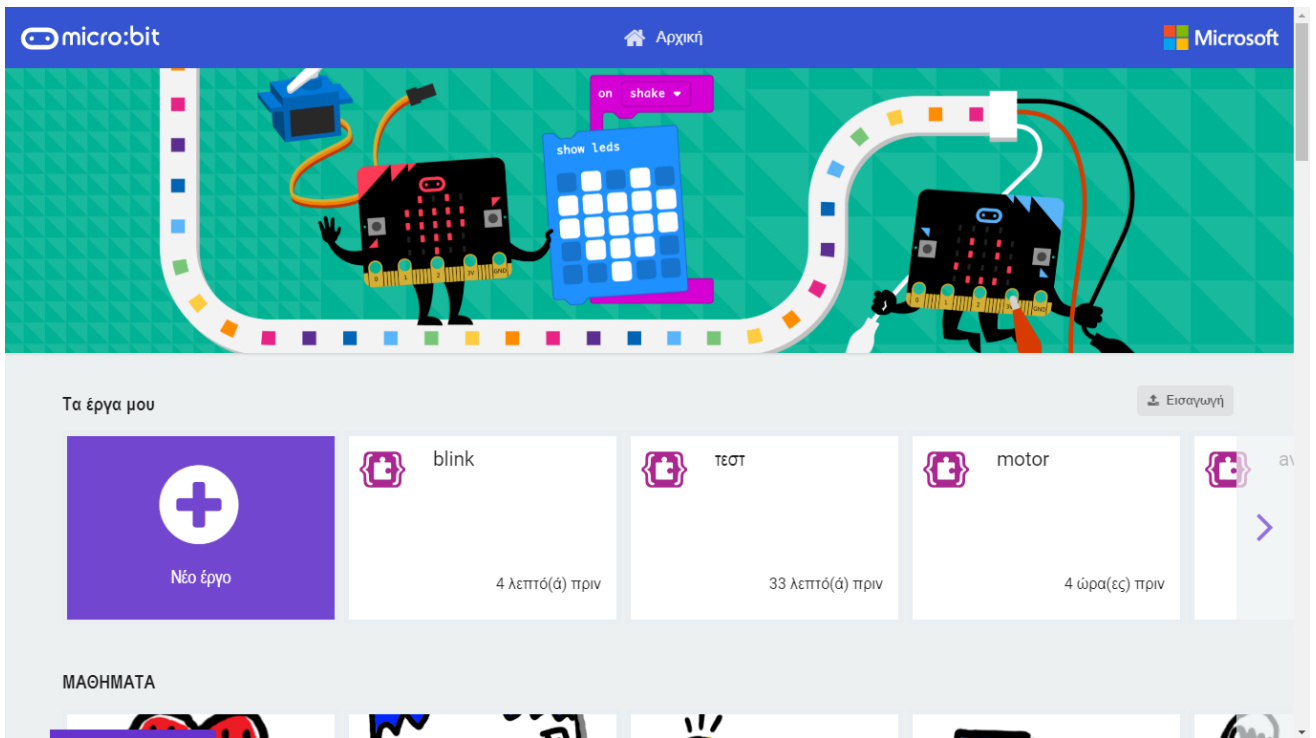
Απαιτούμενος εξοπλισμός

Εξαρτήματα	
1	Micro:bit
2	Micro USB cable (Για να συνδέσουμε το Micro:bit στον υπολογιστή μας)

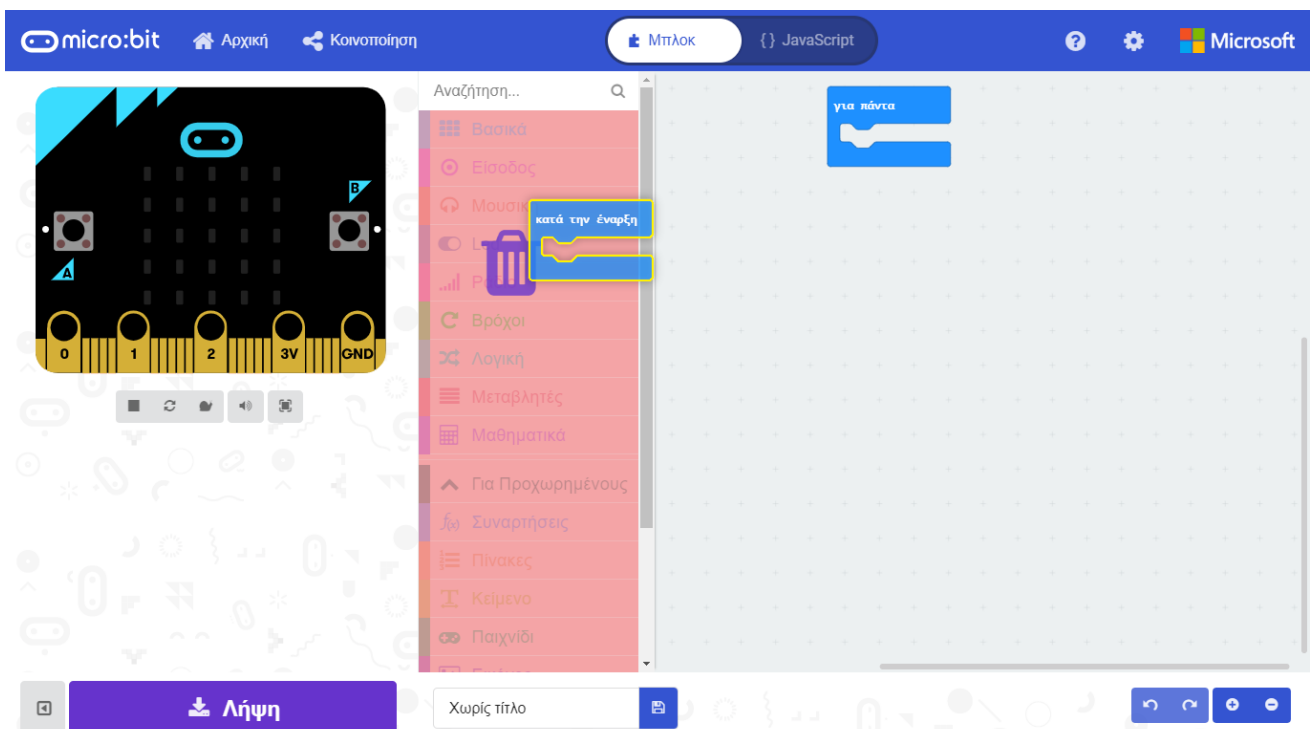
Χρήσιμη πληροφορία: Ένα πρόγραμμα που αντιδρά σε κάτι που κάνει ο χρήστης, όπως το να πιέσει ένα κουμπί, να κάνει κλικ στην οθόνη κλπ. είναι ένα πρόγραμμα που βασίζεται σε γεγονότα (event driven program). Αυτού του είδους ο προγραμματισμός κυριαρχεί στις διαδραστικές εφαρμογές στις οποίες ο χρήστης αλληλεπιδρά με το πρόγραμμα (παιχνίδια, μέσα κοινωνικής δικτύωσης, ιστοσελίδες κλπ.) και το συναντάμε στους επιτραπέζιους υπολογιστές, τα κινητά τηλέφωνα και τις ταμπλέτες.

Βήματα της δραστηριότητας

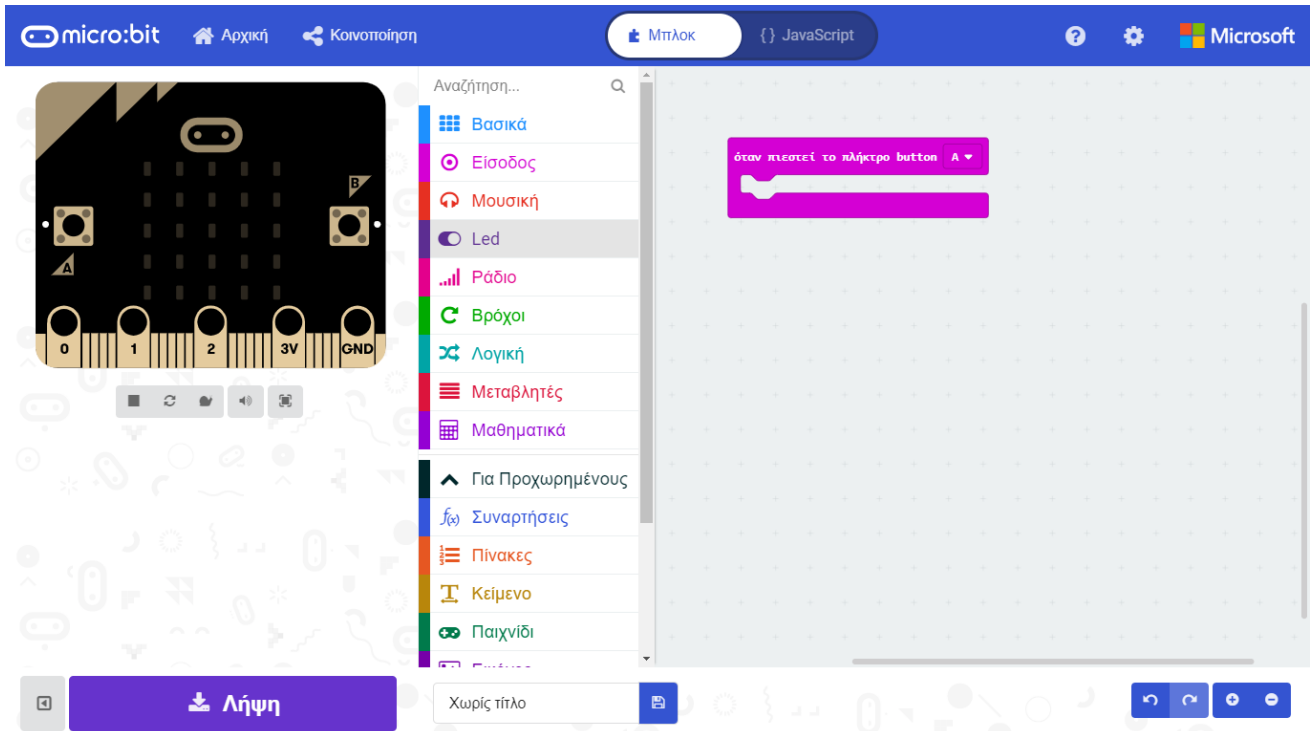
1. Ξεκινήστε ένα νέο Έργο. Για να το κάνετε αυτό πατήστε στο κουμπί «Αρχική» στο πάνω αριστερά μέρος της οθόνης.



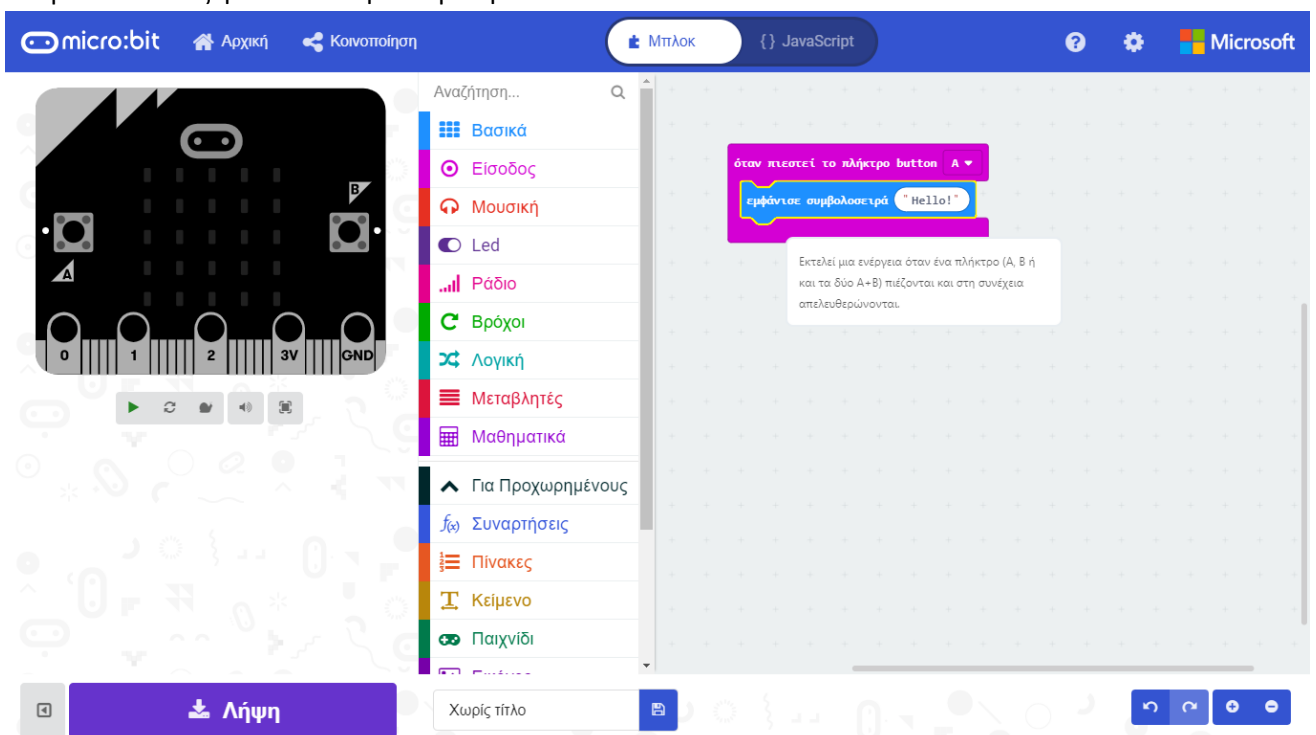
2. Πατήστε το κουμπί «Νέο έργο...»
3. Σβήστε τα μπλοκ «κατά την έναρξη» και «για πάντα» σύροντάς τα στο σκουπιδοντενεκέ. Ο σκουπιδοντενεκές εμφανίζεται άμα πάρετε ένα μπλοκ από την περιοχή προγραμματισμού και το σύρετε στα μπλοκ.



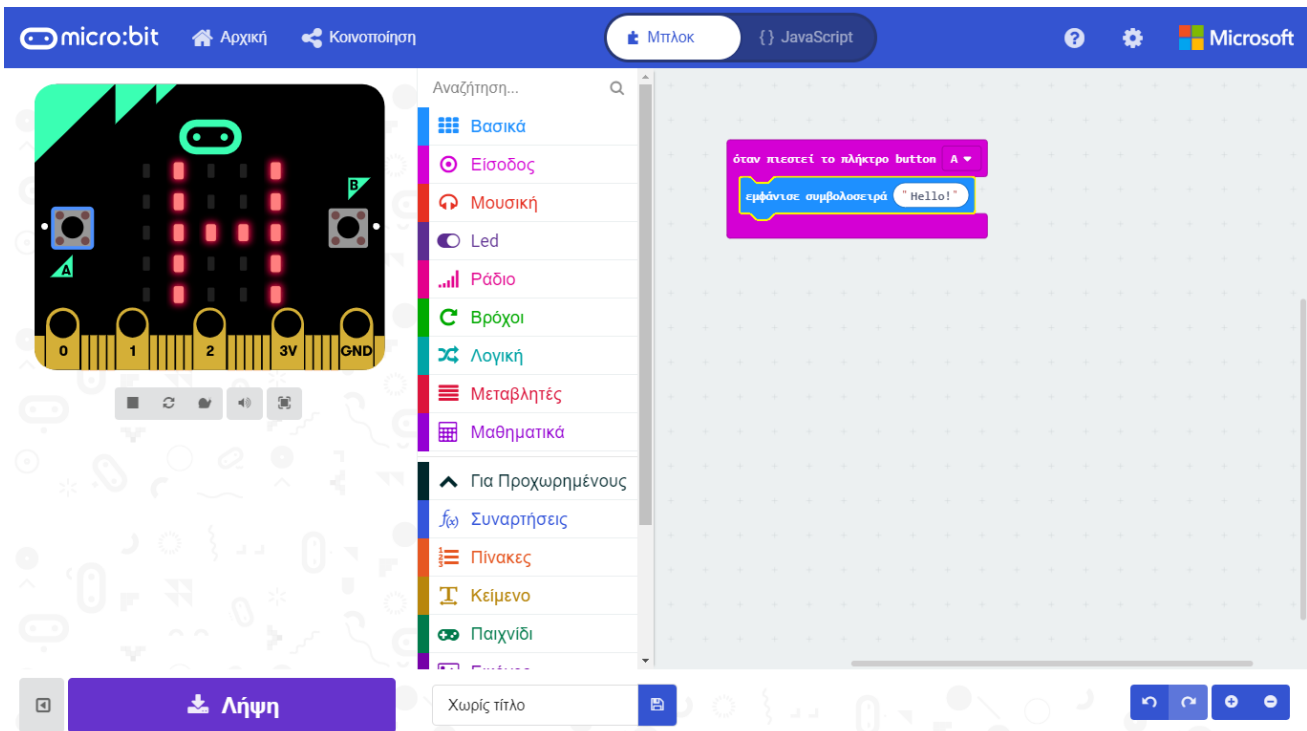
Μπορούμε τώρα να προσθέσουμε τον κώδικα για την απεικόνιση μηνυμάτων. Πηγαίνετε στα μπλοκ και επιλέξτε «Είσοδος». Σύρετε ένα μπλοκ «όταν πιεστεί το πλήκτρο button A» στην περιοχή προγραμματισμού.



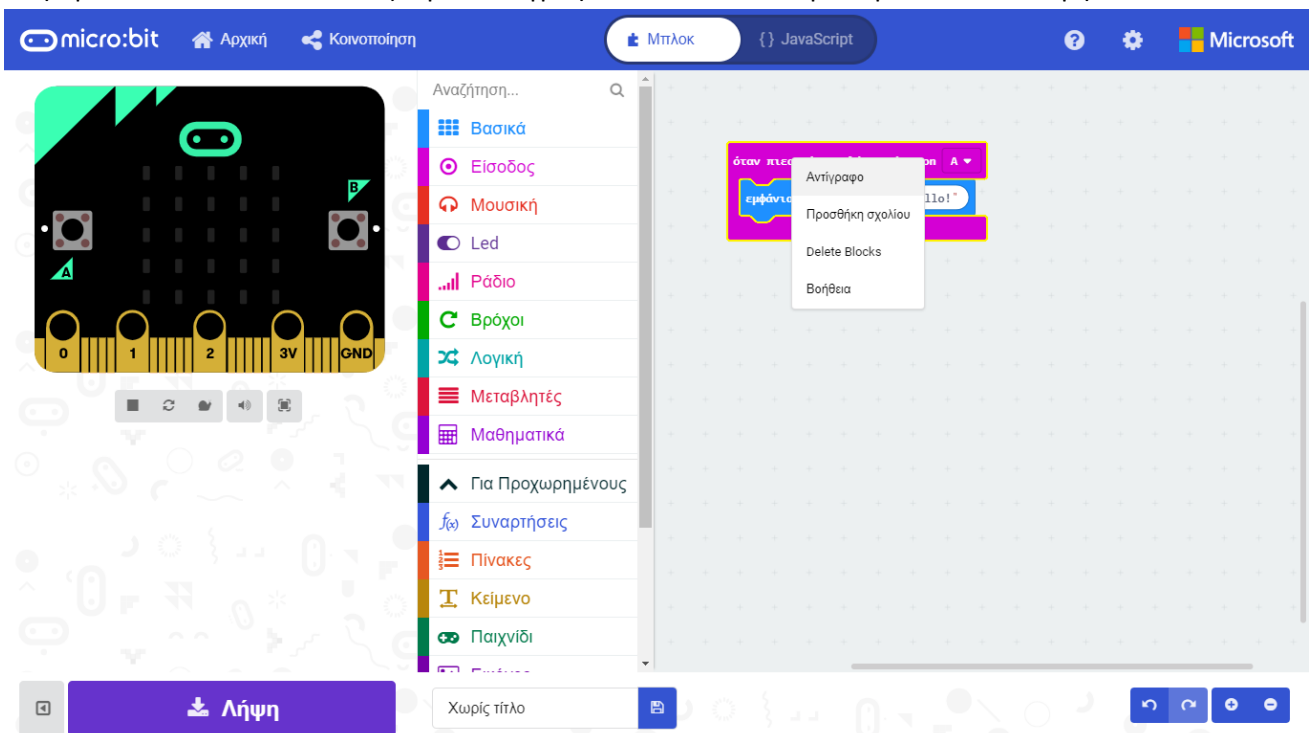
Τώρα πηγαίνετε στην περιοχή των μπλοκ και από την περιοχή «Βασικά» σύρετε ένα μπλοκ «εμφάνισε συμβολοσειρά» και σύρετέ το στο μπλοκ «όταν πιεστεί το πλήκτρο button A» μέχρι να κουμπώσει όπως φαίνεται στην επόμενη εικόνα.



4. Τώρα αν πιέσετε το πλήκτρο A στον προσομοιωτή θα δείτε τη συμβολοσειρά «Hello!» να κυλάει από δεξιά προς τα αριστερά. Αν στη συμβολοσειρά είχατε βάλει μόνο ένα χαρακτήρα, δηλαδή μόνο το «H» από το «Hello!» θα βλέπατε συνεχώς τον χαρακτήρα «H» χωρίς αυτός να κυλάει. Όμως αν στη συμβολοσειρά βάλουμε 2 ή περισσότερους χαρακτήρες, θα έχουμε κύλιση του κειμένου από δεξιά προς τα αριστερά.

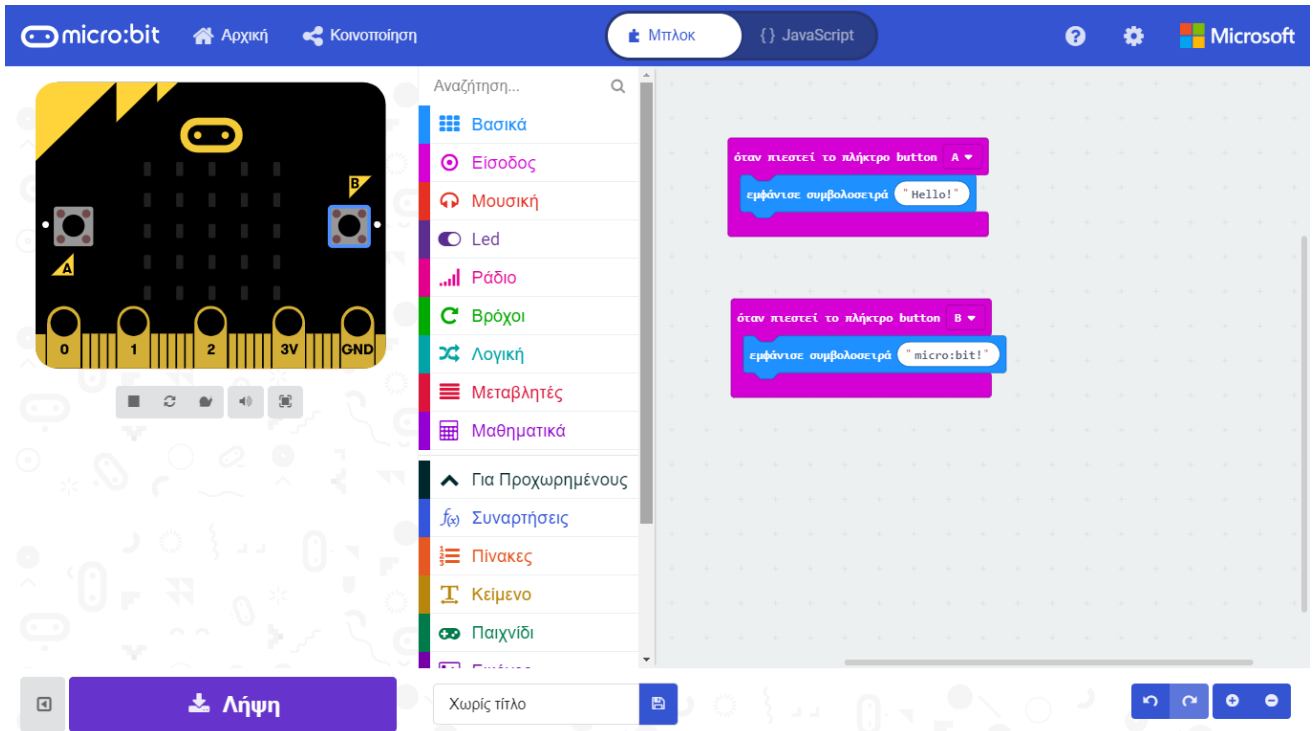


5. Τώρα θα προσθέσουμε ένα δεύτερο κείμενο που θα εμφανίζεται όταν πατάμε το πλήκτρο «B». Ο πιο γρήγορος τρόπος να κάνουμε κάτι τέτοιο είναι να κάνουμε δεξί κλικ στο μπλοκ «όταν πιεστεί το πλήκτρο button A» και να επιλέξουμε «Αντίγραφο» από το αναδυόμενο μενού που θα εμφανιστεί.



6. Ένα αντίγραφο του μπλοκ θα εμφανιστεί. Αλλάξτε το κουμπί από A σε B επιλέγοντας από την λίστα επιλογών του μπλοκ. Στη συνέχεια αλλάξτε το κείμενο του εμφάνισε συμβολοσειρά από «Hello!» σε

«micro:bit!».



7. Τώρα όταν πατάτε το κουμπί B στον προσομοιωτή, η λέξη «micro:bit!» κυλάει από δεξιά προς τα αριστερά.
8. Στο μπλοκ «όταν πιεστεί το πλήκτρο ...» υπάρχει ακόμη μια επιλογή εκτός από τις «A» και «B», η «A+B» που ενεργοποιείται, όταν πατήσουμε ταυτόχρονα τα κουμπιά «A» και «B».

Ασκήσεις:

1. Χρησιμοποιήστε το μπλοκ «όταν πιεστεί το πλήκτρο A + B» για να εμφανίζεται το όνομά σας όταν πατάτε ταυτόχρονα τα πλήκτρα «A» και «B». Τι παρατηρείτε αν γράψετε το πρόγραμμά σας με ελληνικούς χαρακτήρες και αν το γράψετε με λατινικούς;
2. Ανεβάστε τον κώδικα στο micro:bit και δοκιμάστε τον. Πιέζοντας τα πλήκτρα «A» και «B»

Σύνδεσμοι	
1	https://el.wikipedia.org/wiki/%CE%A0%CF%81%CE%BF%CE%B3%CF%81%CE%B1%CE%BC%CE%BC%CE%B1%CF%84%CE%B9%CF%83%CE%BC%CF%8C%CF%82_%CF%87%CE%B5%CE%B9%CF%81%CE%B9%CF%83%CE%BC%CE%BF%CF%8D_%CE%B3%CE%B5%CE%B3%CE%BF%CE%BD%CF%8C%CF%84%CF%89%CE%BD Προγραμματισμός χειρισμού γεγονότων

Δραστηριότητα A3 – Απεικόνιση αριθμών και χρήση των κουμπιών του micro:bit

Περίληψη: Σε αυτή τη δραστηριότητα θα δημιουργήσουμε ένα απλό πρόγραμμα που θα εμφανίζει αριθμούς στην οθόνη του micro:bit. Θα δούμε πως μπορούμε να χρησιμοποιήσουμε τα πλήκτρα «Α» και «Β» του micro:bit για να αλλάξουμε τον αριθμό που εμφανίζεται.

Στόχοι:

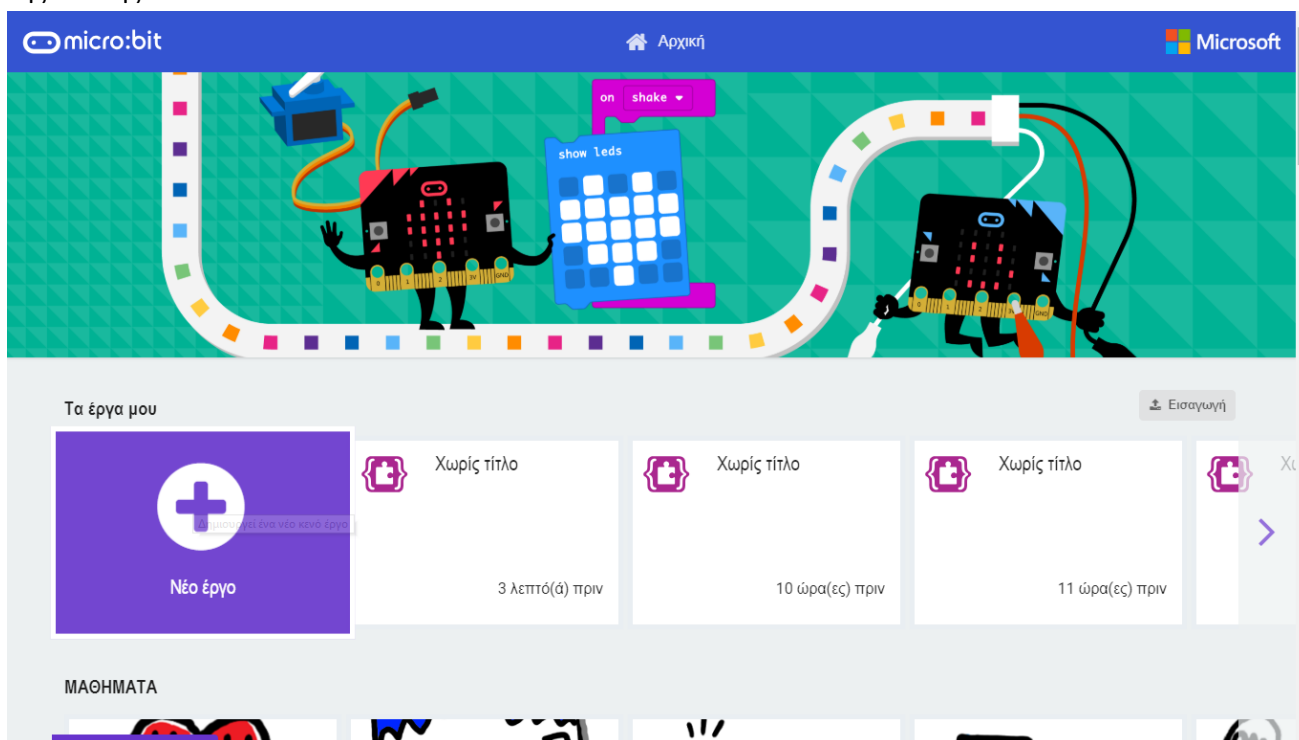
- Η χρήση των κουμπιών Α και Β του micro:bit
- Βλέπουμε πως δουλεύει ο μηχανισμός των γεγονότων στο micro:bit
- Η χρήση μεταβλητών για την αποθήκευση και μεταβολή μιας τιμής.
- Απεικόνιση μιας αριθμητικής τιμής στην οθόνη του micro:bit.

Απαιτούμενος εξοπλισμός

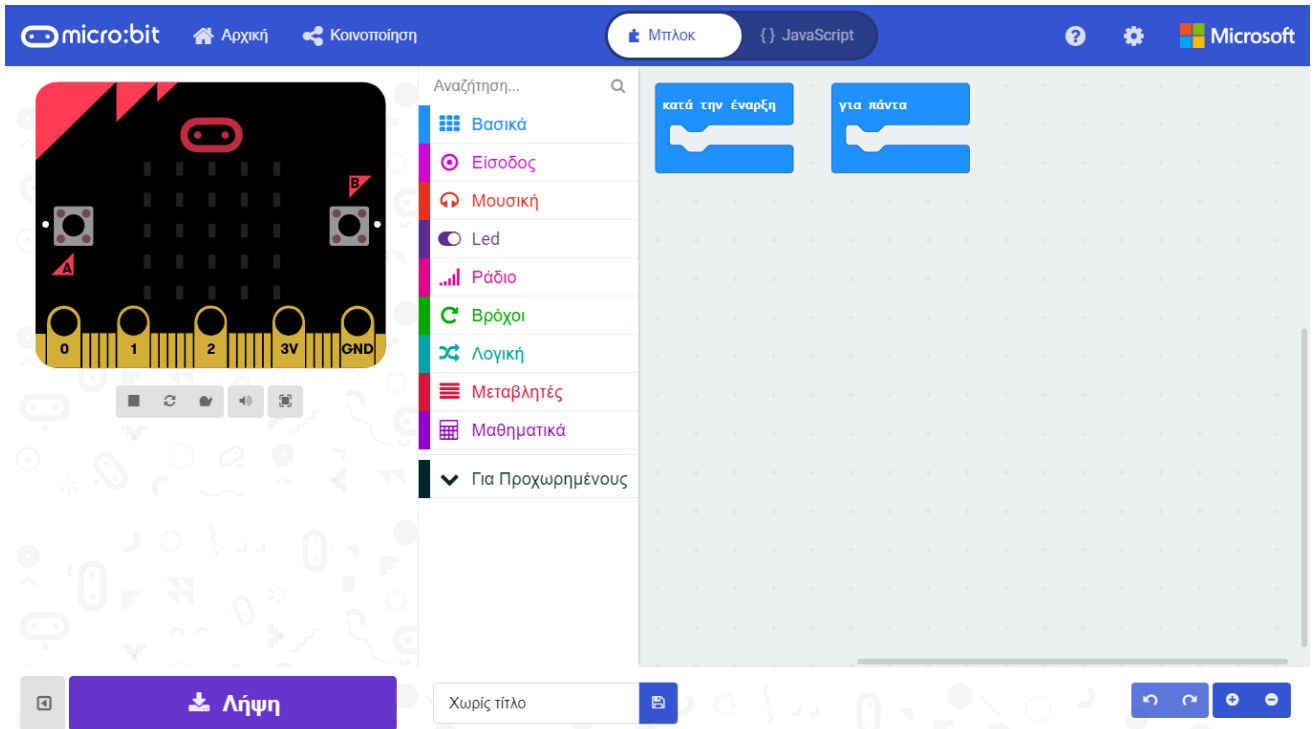
Εξαρτήματα	
1	Micro:bit board
2	Καλώδιο Micro USB (για να συνδέσουμε το micro:bit με τον υπολογιστή)

Βήματα της Δραστηριότητας

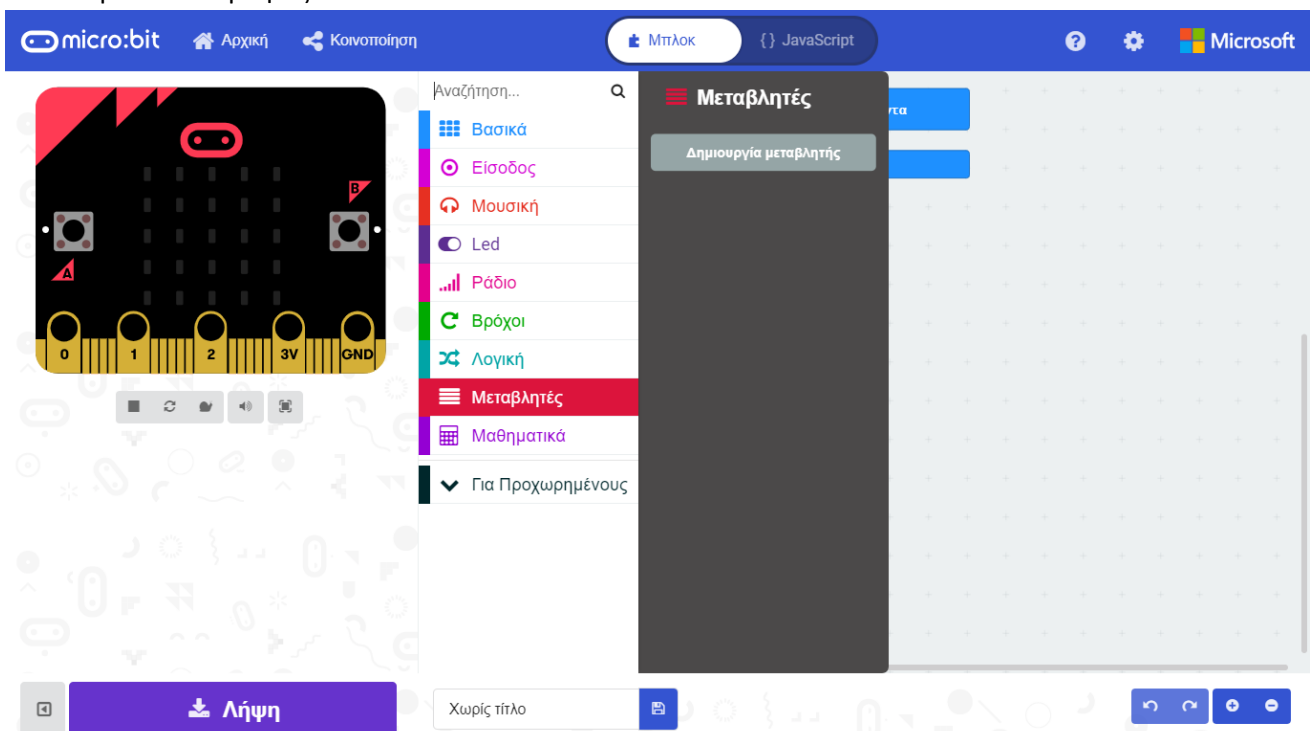
1. Δημιουργήστε ένα καινούριο έργο κάνοντας κλικ στο κουμπί «Αρχική» στην πάνω αριστερά περιοχή της οθόνης:



Πατήστε με το ποντίκι «Νέο έργο...» και θα δείτε μια καινούρια οθόνη με τα δυο μπλοκ «κατά την έναρξη» και «για πάντα».

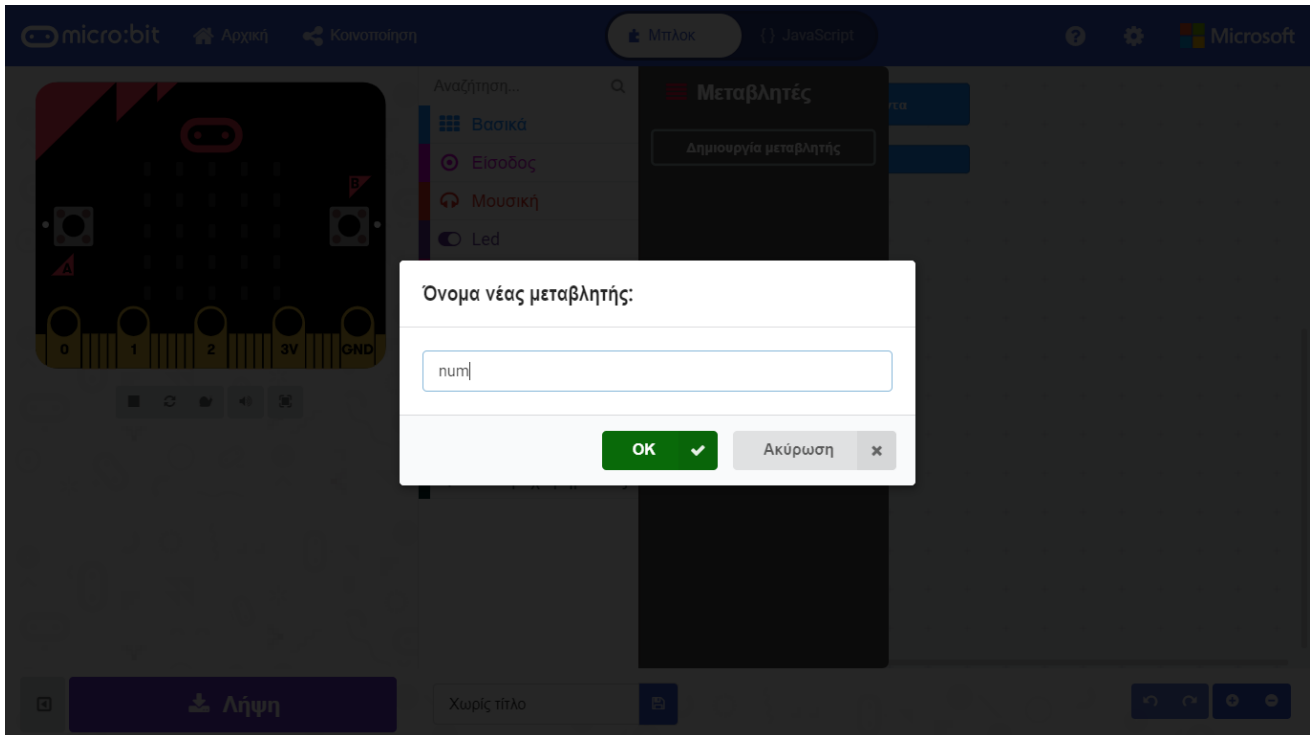


2. Θα χρειαστούμε επίσης και μια **μεταβλητή**. Μια μεταβλητή είναι ένα κομμάτι μνήμης στο οποίο μπορούμε να αποθηκεύσουμε αριθμούς, κείμενο, εικόνες και να το χειριστούμε με πολλούς διαφορετικούς τρόπους. Είναι καλύτερο να αρχικοποιήσουμε μια μεταβλητή, δηλαδή να της δώσουμε μια τιμή πριν τη χρησιμοποιήσουμε. Αν για παράδειγμα θέλουμε να χειριστούμε αριθμούς να της δώσουμε την τιμή 0. Το καλύτερο μέρος για να αρχικοποιήσουμε μια μεταβλητή είναι μέσα σε ένα μπλοκ «κατά την έναρξη». Για να δημιουργήσουμε μια μεταβλητή κάνουμε κλικ στο κουμπί «Μεταβλητές».



Μετά από αυτό εμφανίζονται ένα κουμπί με το κείμενο «Δημιουργία μεταβλητής». Κάνοντας κλικ

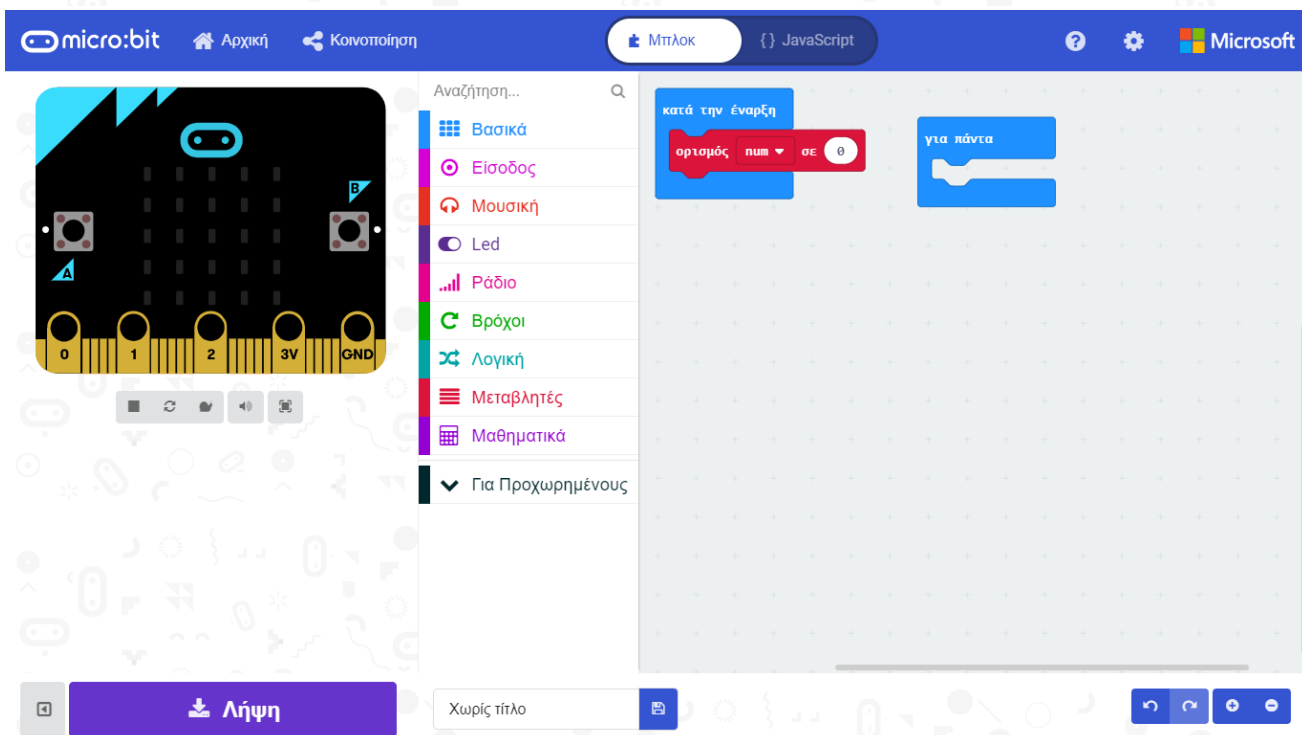
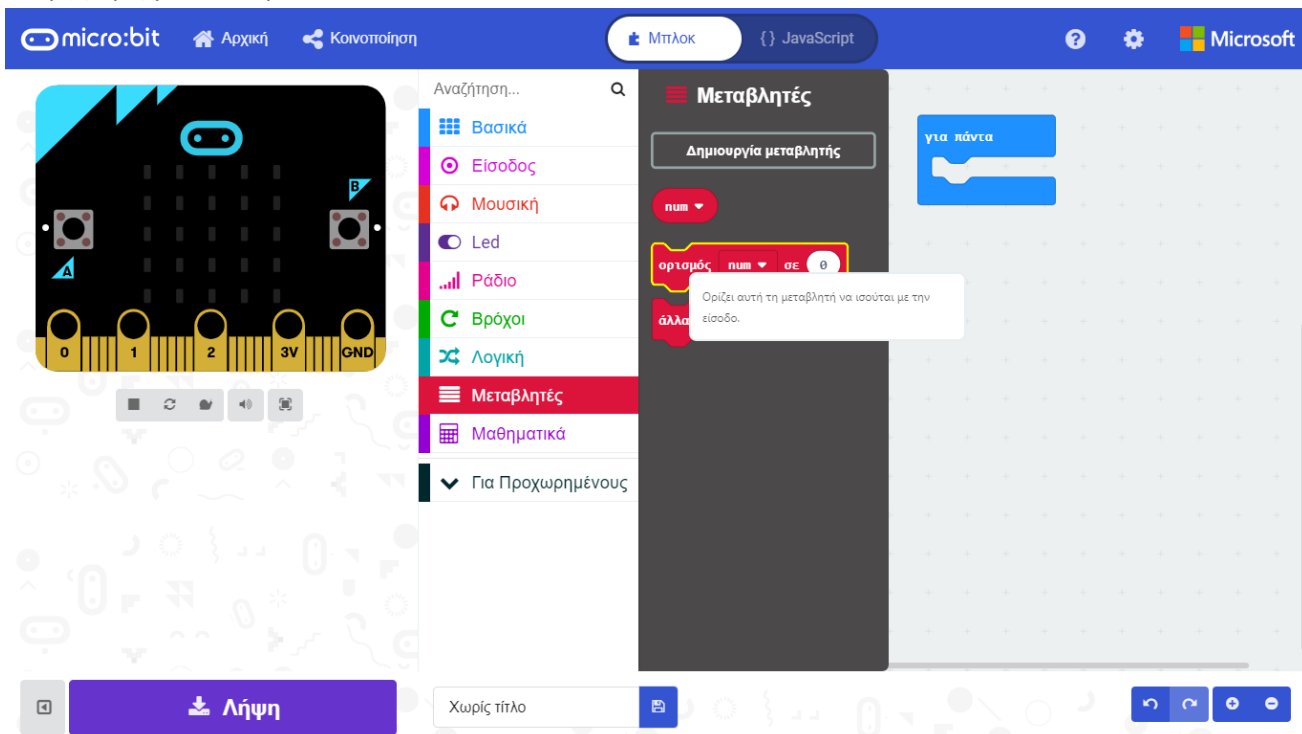
στο κουμπί «Δημιουργία μεταβλητής» εμφανίζεται ένα πλαίσιο κειμένου.



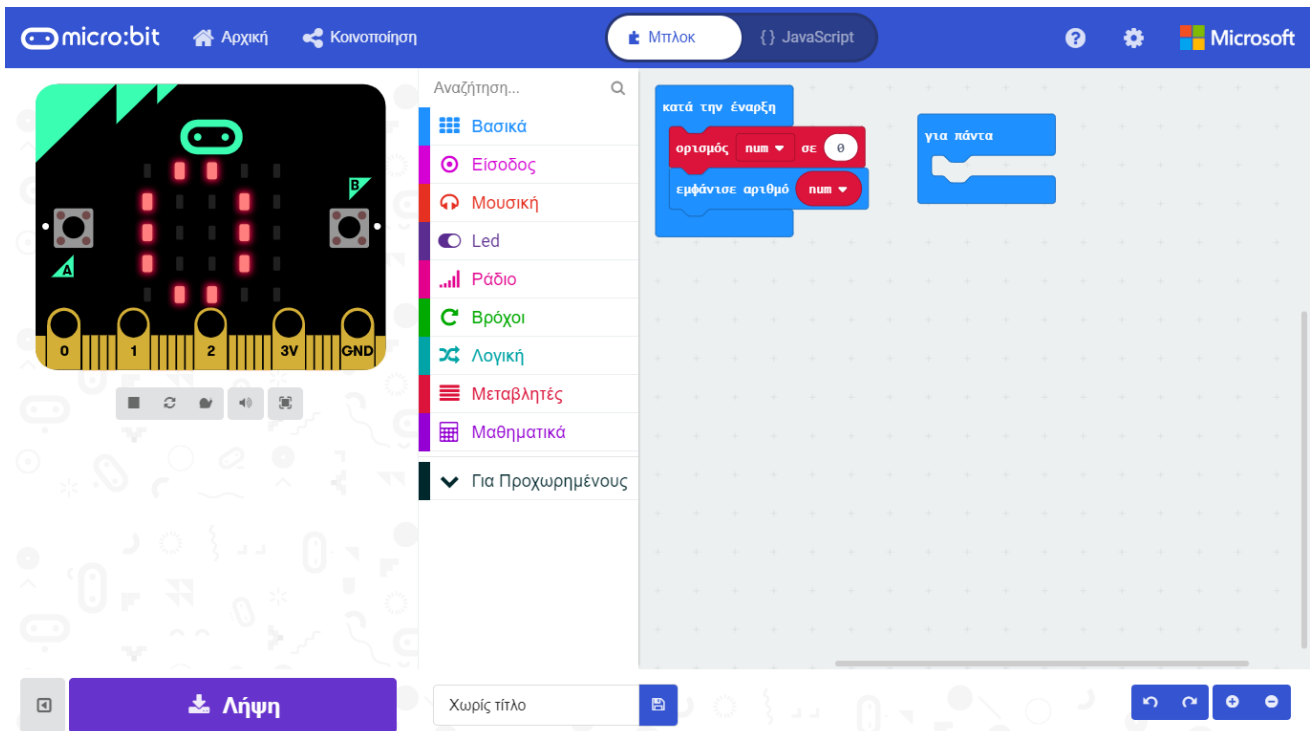
3. Πληκτρολογήστε «num» στο πλαίσιο κειμένου και πατήστε το πλήκτρο «OK». Αυτό θα είναι το όνομα της μεταβλητής που θα χρησιμοποιήσουμε για να αποθηκεύσουμε τον αριθμό μας. Τώρα διαλέγουμε το μπλοκ «ορισμός num σε 0» και το τραβάμε με το ποντίκι μέσα στο μπλοκ «κατά την

Χρήσιμη πληροφορία: Μια **μεταβλητή** είναι ένα κομμάτι μνήμης του υπολογιστή που μπορεί να αποθηκεύσει μια τιμή που μπορεί να τροποποιηθεί από ένα πρόγραμμα (σε αντιδιαστολή με μια **σταθερά** που αποθηκεύει τιμές που δεν μπορούν να τροποποιηθούν). Η τιμή που αποθηκεύεται στη μεταβλητή μπορεί να είναι αριθμητική, για παράδειγμα ένας ακέραιος αριθμός όπως είναι το 1, το 25 ή το -1000 ή ένας αριθμός κινητής υποδιαστολής όπως είναι το 3,14159 ή το 0,104, κείμενο όπως το «Hello World» ή το «το αποτέλεσμα είναι:», ημερομηνία ή ώρα όπως: 23 Ιουλίου 2018), δυαδικά δεδομένα δηλαδή μια σειρά από δυαδικά ψηφία (bytes). Συνήθως είναι δυνατόν να δημιουργηθούν σύνθετοι τύποι δεδομένων που αποτελούνται από ένα σύνολο απλούστερων τύπων δεδομένων. Συχνά οι μεταβλητές χρειάζονται κάποια αρχικοποίηση, δηλαδή κάποια συγκεκριμένη αρχική τιμή. Σε κάποιες γλώσσες προγραμματισμού, μια μεταβλητή μπορεί να αποθηκεύσει μόνο ένα είδος τιμών, για παράδειγμα μόνο κείμενο, που ορίζεται κατά την αρχικοποίηση του προγράμματος. Άλλες γλώσσες προγραμματισμού επιτρέπουν σε μια μεταβλητή να αποθηκεύει διαφορετικά είδη τιμών.

έναρξη» μέχρι να κουμπώσει.

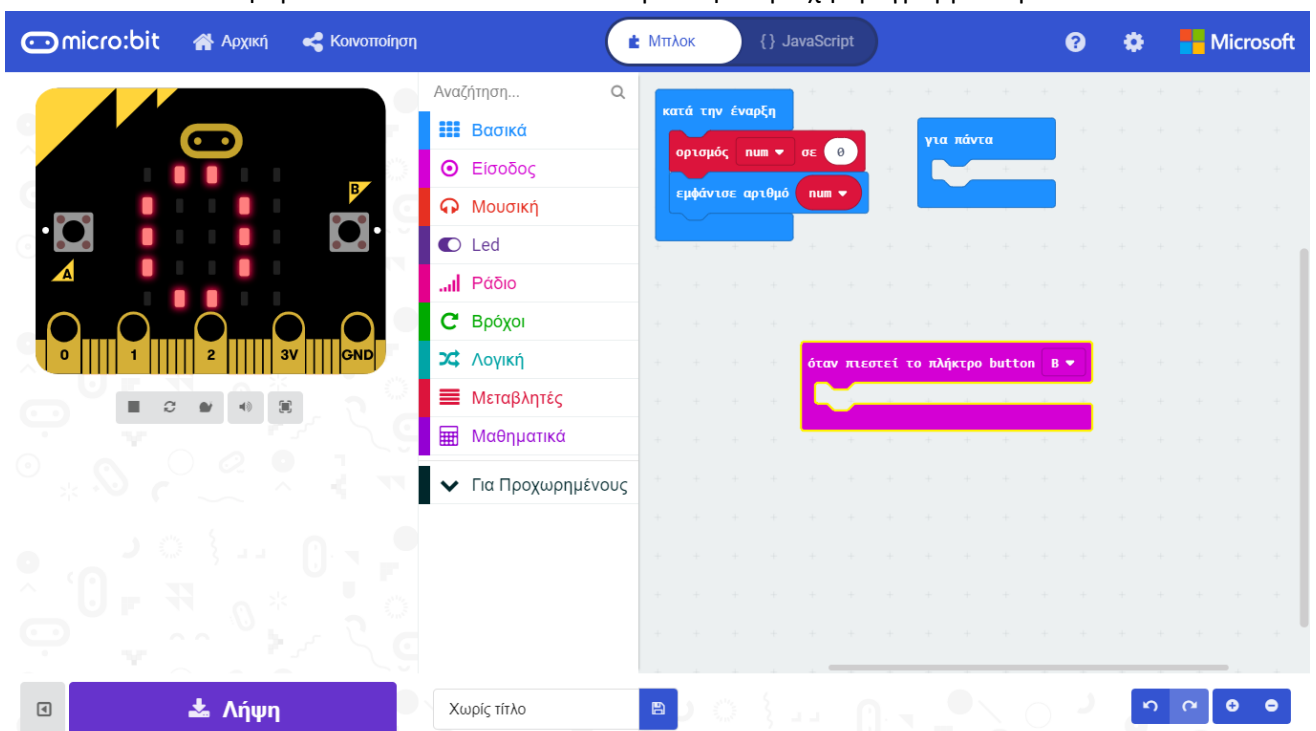


4. Τώρα που έχουμε τη μεταβλητή μας ας την δείξουμε στην οθόνη χρησιμοποιώντας το μπλοκ «εμφάνισε αριθμό...» που βρίσκεται στην «Βασική» κατηγορία μπλοκ.



Βάζουμε το μπλοκ «εμφάνισε αριθμό...» κάτω από το μπλοκ «ορισμός... σε...». Στη συνέχεια πηγαίνουμε στο «Μεταβλητές» και παίρνουμε το μπλοκ num και το κουμπώνουμε στο μπλοκ «εμφάνισε αριθμό». Ο προσομοιωτής θα μας δείξει την τιμή 0. Ας αλλάξουμε την τιμή της μεταβλητής μας σε 1 κάνοντας κλικ στο μπλοκ «ορισμός... σε ...» στο 0 και γράφοντας 1. Θα δούμε τον αριθμό που εμφανίζεται να γίνεται 1. Αφού το είδαμε αυτό, ας αλλάξουμε πάλι την τιμή της μεταβλητής σε 0.

5. Τώρα είναι η ώρα να προσθέσουμε στο πρόγραμμά μας περισσότερες δυνατότητες. Θα προσθέσουμε την δυνατότητα να αυξήσουμε κατά 1 την τιμή της μεταβλητής «num» πιέζοντας το κουμπί «B». Για να το κάνουμε αυτό επιλέγουμε τα μπλοκ «Είσοδος» και επιλέγουμε το μπλοκ «όταν πιεστεί το πλήκτρο button B» και το αποθέτουμε στην περιοχή προγραμματισμού.

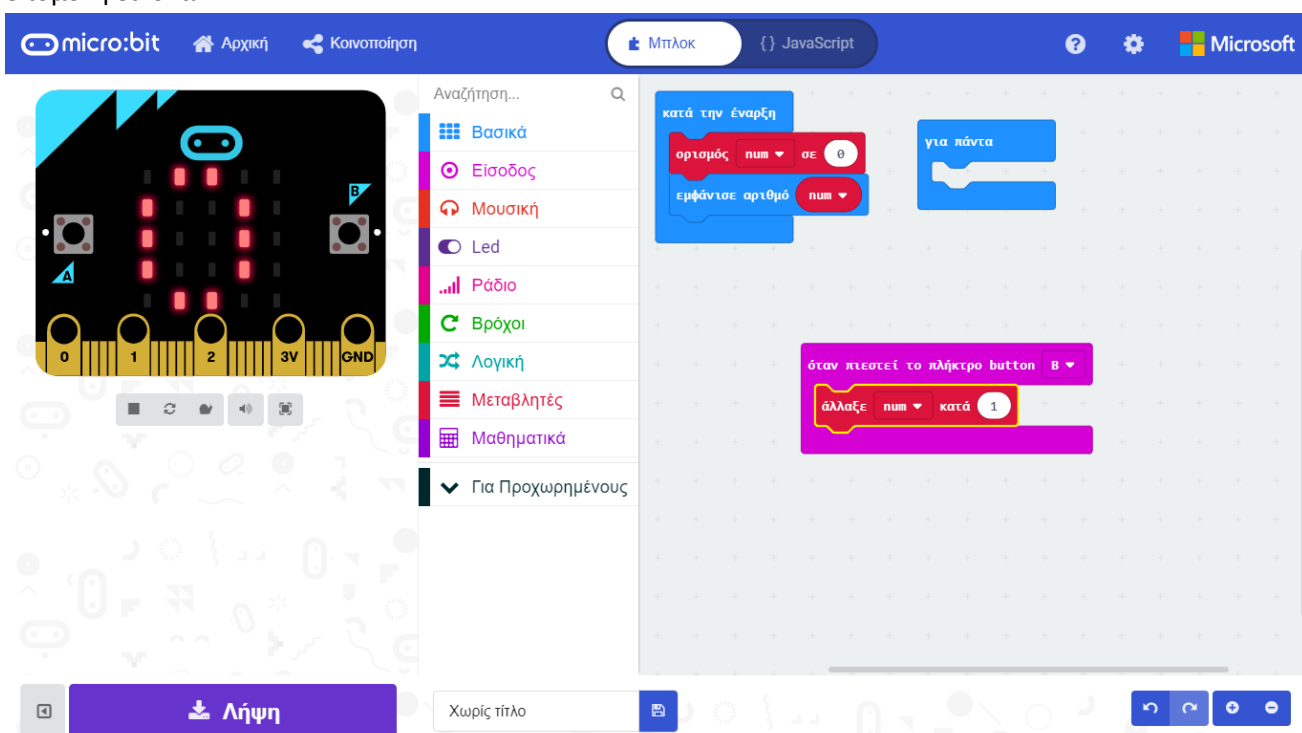


6. Το μπλοκ «όταν πιεστεί το πλήκτρο button ...» που μόλις προσθέσαμε είναι ένα παράδειγμα διαχειριστή γεγονότων (event handler) Ένας διαχειριστής γεγονότων είναι ένα κομμάτι κώδικα που εκτελείται όταν συμβεί το αντίστοιχο γεγονός. Σε αυτή την περίπτωση κάποιος κώδικας εκτελείται

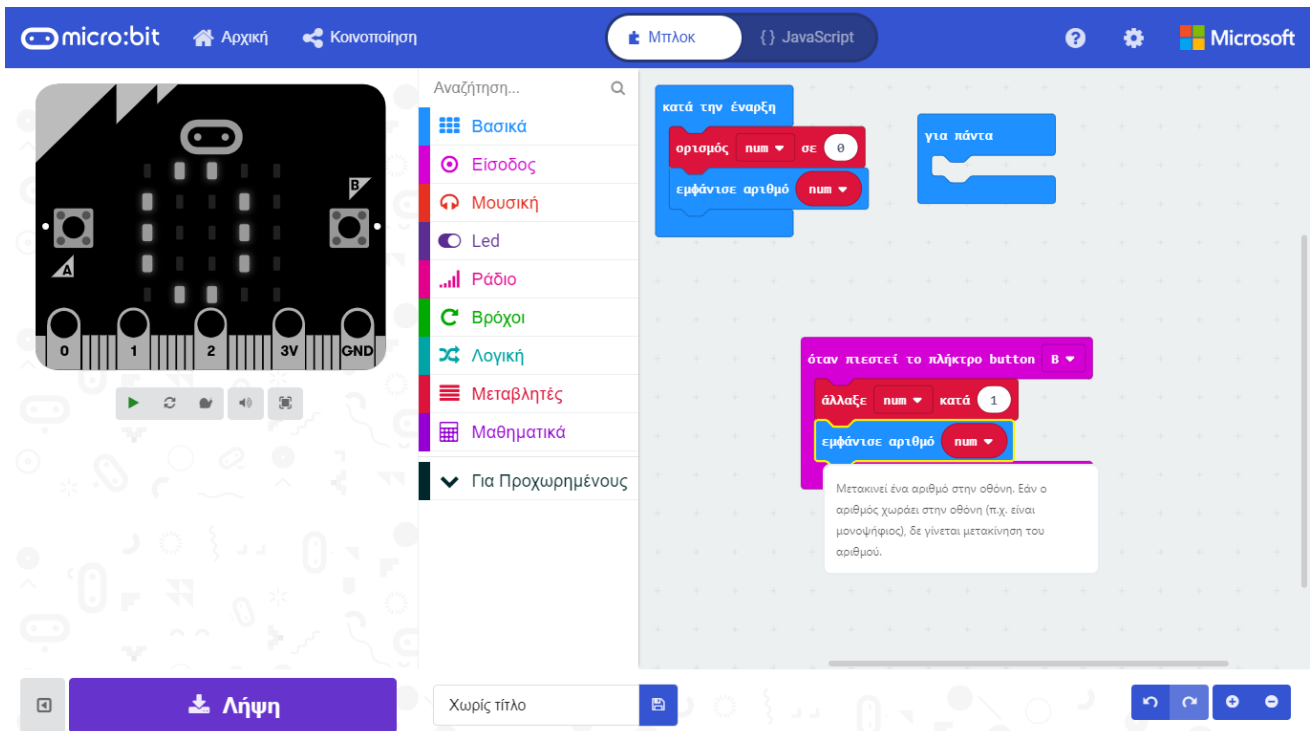
Χρήσιμη πληροφορία: Η επιλογή «Αντιγραφή» σε ένα μπλοκ αντιγράφει και τα μπλοκ που αυτό περικλείει. Έτσι πρέπει να ήμαστε προσεκτικοί στο που κάνουμε κλικ πριν πιέσουμε «Αντιγραφή» Επίσης μερικές φορές είναι αναγκαίο να σβήσουμε κάποια μπλοκ.

όταν πιέσουμε το πλήκτρο B στο micro:bit. Επειδή τώρα το μπλοκ είναι άδειο για την ώρα αν πιέσουμε το πλήκτρο B δεν πρόκειται να γίνει τίποτα. Ας το αλλάξουμε αυτό. Θέλουμε να αυξήσουμε κατά 1 την τιμή της μεταβλητής «num», έτσι κάνουμε κλικ στις μεταβλητές και επιλέγουμε το μπλοκ «άλλαξε ... κατά 1».

Τοποθετούμε το μπλοκ μέσα στο «όταν πιεστεί το πλήκτρο button ...» μπλοκ όπως φαίνεται στην επόμενη εικόνα



Αν πιέσουμε το πλήκτρο «B» ο κώδικάς μας εκτελείται αλλά δείχνει ακόμα 0. Γιατί; Γιατί δεν δείχνουμε την καινούρια τιμή. Ο πιο εύκολος τρόπος είναι να χρησιμοποιήσουμε το μπλοκ «εμφάνισε αριθμό» που υπάρχει μέσα στο μπλοκ «κατά την έναρξη» και να του κάνουμε δεξί κλικ, επιλέγοντας «Αντιγραφή». Στη συνέχεια μπορούμε να σύρουμε το μπλοκ κάτω από το μπλοκ «άλλαξε num κατά 1» .



Αυτό ήταν! Τώρα μπορούμε να αυξήσουμε την τιμή της μεταβλητής «num» πατώντας το κουμπί «B» και να δούμε το αποτέλεσμα στην οθόνη του micro:bit.

Ασκήσεις:

1. Προσθέστε τη δυνατότητα να μειώνετε την τιμή της μεταβλητής num κατά 1 πατώντας το κουμπί «A» (Χρήσιμη πληροφορία: Η επιλογή «Αντιγραφή» μπορεί να σας φανεί χρήσιμη).
2. Προσθέστε την δυνατότητα να αρχικοποιήσετε την τιμή του «num» σε 0 πατώντας ταυτόχρονα τα πλήκτρα «A» και «B».
3. Ανεβάστε το πρόγραμμα στο micro:bit και δοκιμάστε το

Links	
1	https://en.wikipedia.org/wiki/Variable_(computer_science) Τι είναι μεταβλητή;

Δραστηριότητα A4 – Χρήση ενός πακέτου για να προσθέσουμε δυνατότητες

Περίληψη: Σε αυτή τη δραστηριότητα θα χτίσουμε πάνω στο προηγούμενο πρόγραμμα κάνοντας κάποιες βελτιώσεις. Θα δούμε πως μπορούμε να χρησιμοποιήσουμε ένα πακέτο για να προσθέσουμε νέα λειτουργικότητα στο micro:bit και πως τα πακέτα μας επιτρέπουν να διαχειριστούμε καλύτερα τους περιορισμένους πόρους του micro:bit.

Στόχοι:

- Χρήση του διαχειριστή πακέτων για προσθήκη νέων μπλοκ στο micro:bit. Εξαιτίας της περιορισμένης μνήμης και πόρων, το micro:bit έχει μια περιορισμένη βασική λειτουργικότητα και πρόσθετες δυνατότητες προστίθενται με τη μορφή πακέτων που εγκαθιστούμε.

Απαιτούμενος εξοπλισμός

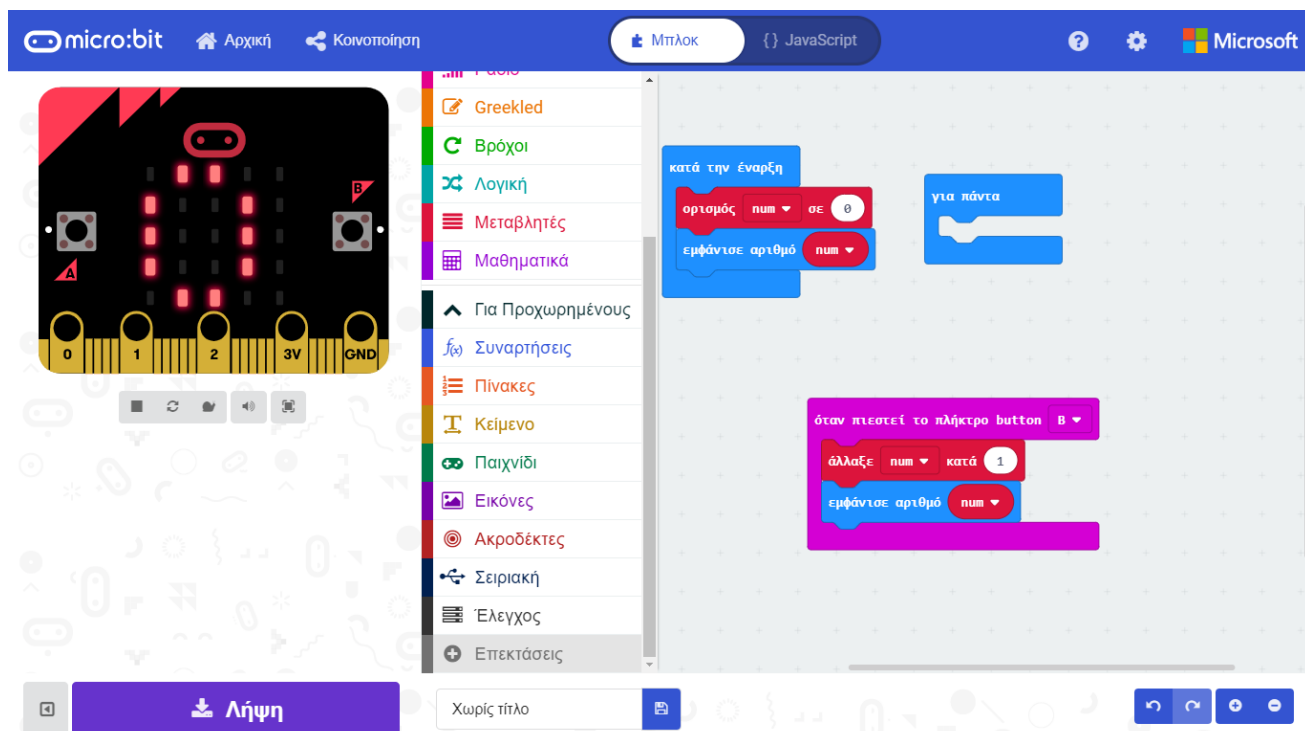
Εξαρτήματα	
1	Micro:bit board
2	Καλώδιο Micro USB (για να συνδέσουμε το micro:bit με τον υπολογιστή)

Βήματα της δραστηριότητας

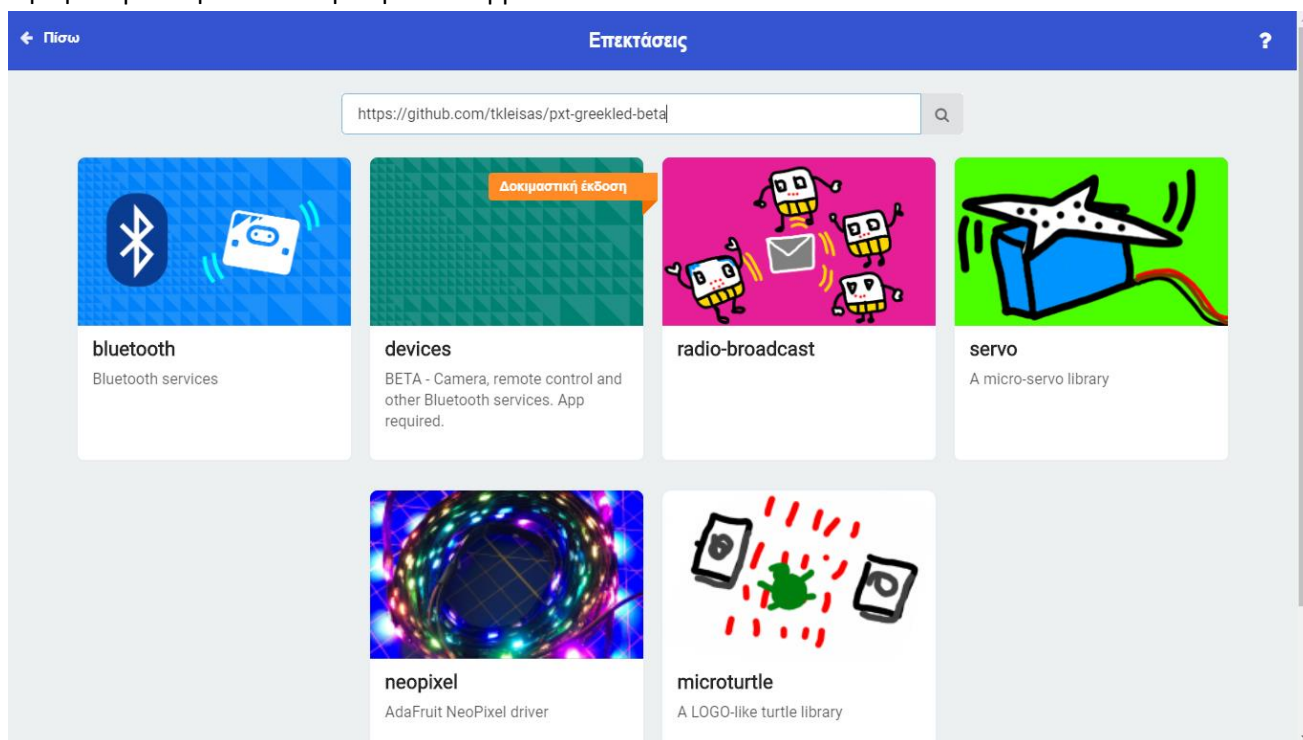
1. Στην προηγούμενη δραστηριότητα φτιάξαμε ένα πρόγραμμα που μπορεί να απεικονίσει αριθμούς που μπορούμε να αυξήσουμε και να μειώσουμε χρησιμοποιώντας τα δυο πλήκτρα «Α» και «Β» Το πρόβλημα με αυτό το πρόγραμμα είναι ότι όταν ο αριθμός μας είναι μεγαλύτερος από ένα ψηφίο η οθόνη κυλάει έτσι ώστε να δείξει ολόκληρο τον αριθμό. Έτσι έχουμε αναπτύξει ένα πακέτο που επιτρέπει την απεικόνιση αριθμών με 2 ή και 3 ψηφία χωρίς να κυλάει την οθόνη και χρησιμοποιώντας ένα 2x5 led matrix για την απεικόνιση των αριθμών. Επιπρόσθετα μας επιτρέπει να δείχνουμε μηνύματα στα Ελληνικά στην οθόνη του micro:bit. Για να προσθέσουμε ένα πακέτο, ανοίγουμε την ενότητα «Για προχωρημένους» και βρίσκουμε το κουμπί «επεκτάσεις».

Χρήσιμη πληροφορία: Ένα πακέτο ή μια βιβλιοθήκη είναι ένας τρόπος να επαναχρησιμοποιήσουμε κώδικα σε πολλά προγράμματα. Τον παλιό καιρό όταν γραφόταν ένα πρόγραμμα, γραφόταν από την αρχή. Αυτό ήταν βαρετό και προέκυπταν πολλά λάθη. Η λύση που προτάθηκε ήταν η δημιουργία βιβλιοθηκών κώδικα ή πακέτα κώδικα που περιέχουν καλά ορισμένες συναρτήσεις και έχουν ελεγχθεί για λάθη. Τα πακέτα και οι βιβλιοθήκες ανήκουν και αυτά στην λογική των υποπρογραμμάτων των διαδικασιών και των συναρτήσεων και επιτρέπουν τη δημιουργία προγραμμάτων σπάζοντας ένα μεγαλύτερο πρόβλημα σε πολλά μικρότερα, καλώς ορισμένα και κατά προτίμηση ήδη λυμένα προβλήματα.

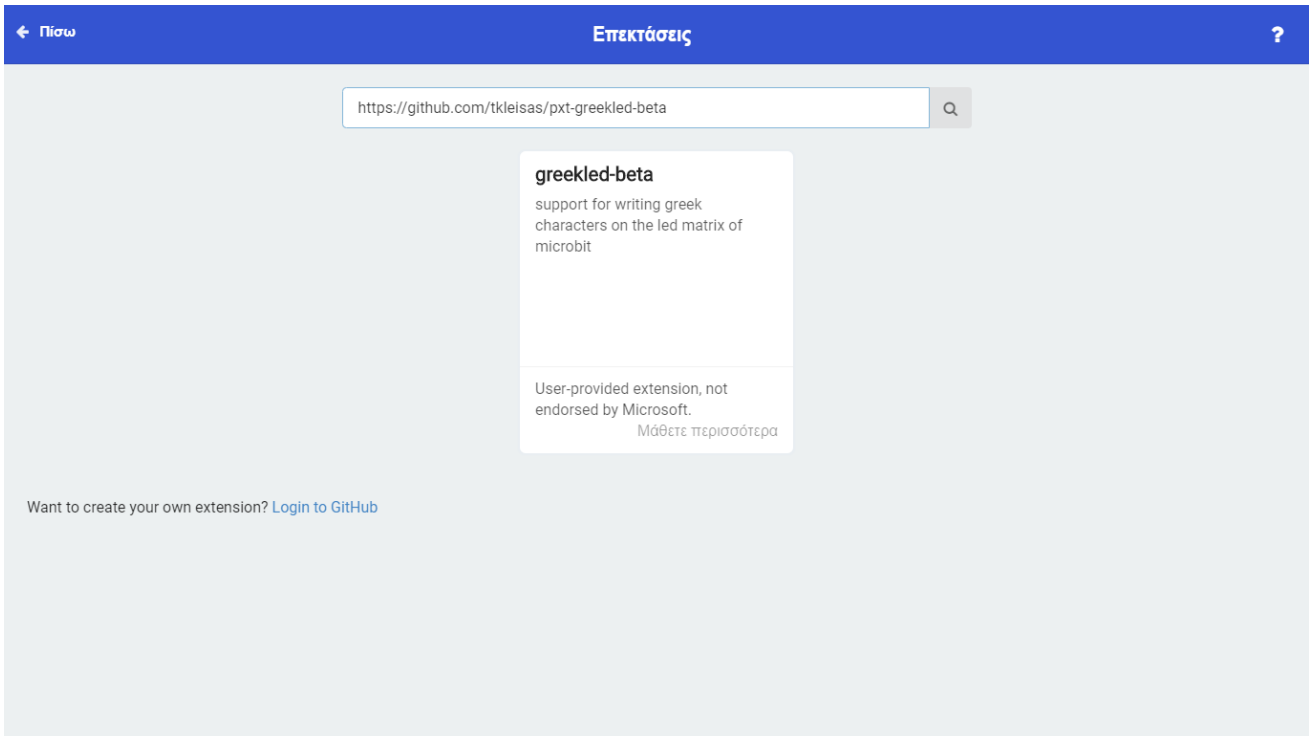
2.



Πατώντας το «Προσθήκη πακέτου» εμφανίζεται μια καινούρια οθόνη που μας επιτρέπει να γράψουμε την ηλεκτρονική διεύθυνση στην οποία βρίσκεται το πακέτο.

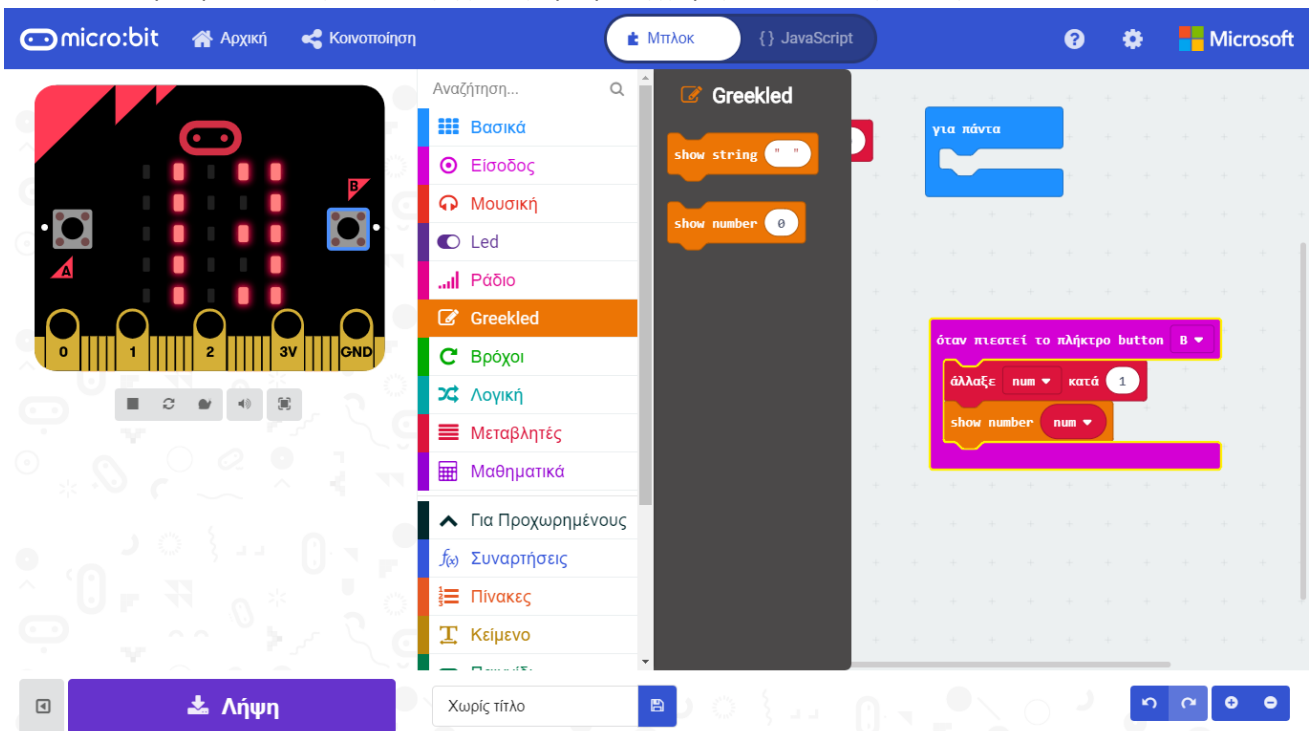


Γράψτε τη διεύθυνση <https://github.com/tkleisas/pxt-greekled-beta> εκεί που λέει «Αναζητήστε ή εισάγετε τη διεύθυνση URL του έργου...» και πατήστε το πλήκτρο ENTER. Στη συνέχεια εμφανίζεται η ακόλουθη οθόνη:



Αν κάνετε κλικ στο «greekled-beta» το πακέτο θα προστεθεί στα διαθέσιμα μπλοκ.

3. Αυτό έχει σαν συνέπεια να εμφανιστεί μια νέα κατηγορία στα μπλοκ με την ονομασία Greekled. Η κατηγορία αυτή περιέχει 2 μπλοκ. «show string» που μας επιτρέπει να γράψουμε και ελληνικά στην οθόνη του micro:bit και «show number» που μας επιτρέπει να δείξουμε πολλά ψηφία στην οθόνη του micro:bit. Αν αντικαταστήσουμε την αρχική εντολή «εμφάνισε αριθμό» με την εντολή «show number» θα δείτε ότι μπορεί να δείξει και διψήφιους αριθμούς χωρίς να κυλίσει η οθόνη.



Ασκήσεις:

1. Αντικαταστήστε την «εμφάνισε αριθμό» με την εντολή του πακέτου greekled “show number”.
2. Ανεβάστε το πρόγραμμα με την πρόσθετη λειτουργικότητα στο micro:bit.

Σύνδεσμοι

1	https://github.com Το Github είναι το μεγαλύτερο αποθετήριο ανοιχτού κώδικα. Τα πακέτα του Micro:bit ζουν εκεί! Μπορούμε να τα κατεβάσουμε, να τα μελετήσουμε και κυρίως να τα τροποποιήσουμε φέρνοντάς τα στα μέτρα μας.
---	--

Δραστηριότητα A5 – Το επιταχυνσιόμετρο του micro:bit

Περίληψη: Σε αυτή τη δραστηριότητα θα δούμε έναν πολύ ενδιαφέροντα αισθητήρα που είναι ενσωματωμένος στο micro:bit: Το επιταχυνσιόμετρο. Θα δούμε πως μπορούμε να χρησιμοποιήσουμε αυτό τον αισθητήρα για να ανιχνεύσουμε τη σχετική θέση του micro:bit και επίσης πως το micro:bit χρησιμοποιεί το επιταχυνσιόμετρο για να ανιχνεύσει κινήσεις όπως το να κουνήσουμε το micro:bit. Θα χρησιμοποιήσουμε επίσης τη σύνδεση USB ως ένα μέσο για να μεταφέρουμε κάποια δεδομένα μεταξύ του micro:bit και του υπολογιστή μας..

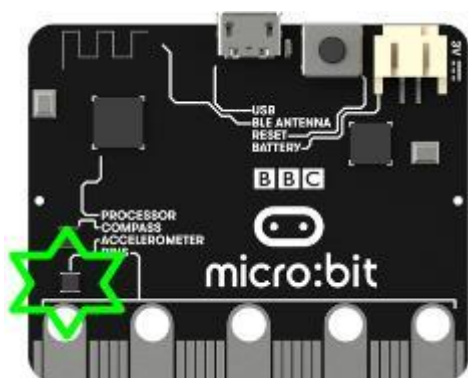
Στόχοι:

- Να κατανοήσουμε πως δουλεύει το επιταχυνσιόμετρο.
- Να χρησιμοποιήσουμε το επιταχυνσιόμετρο ως συσκευή εισόδου.
- Να χρησιμοποιήσουμε τις κινήσεις (gestures) που υποστηρίζει το επιταχυνσιόμετρο έτσι ώστε να φτιάξουμε μια προσομοίωση ζαριού που θα μας δείχνει έναν αριθμό στην οθόνη του micro:bit όταν το ταρακουνάμε.
- Να χρησιμοποιήσουμε τη σειριακή σύνδεση USB για να στείλουμε δεδομένα στον υπολογιστή

Απαιτούμενο υλικό

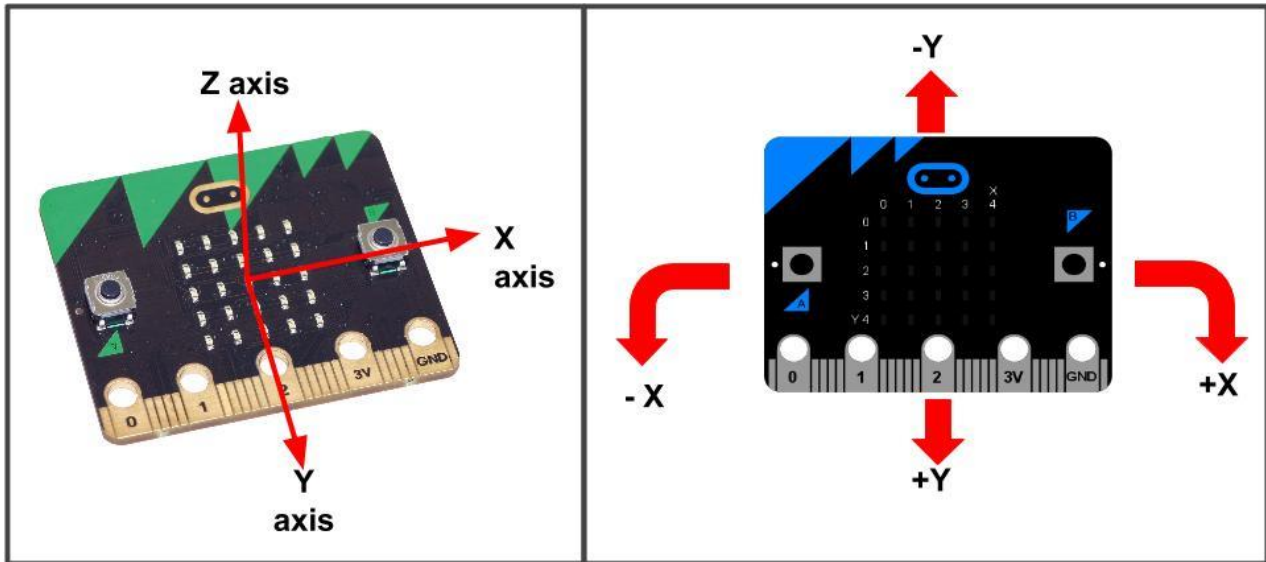
Εξαρτήματα	
1	Micro:bit board
2	Καλώδιο Micro USB (για να συνδέσουμε το micro:bit στον υπολογιστή)

Το επιταχυνσιόμετρο στο micro:bit μετράει επιτάχυνση, πράγμα που δηλώνει και το όνομά του. Το επιταχυνσιόμετρο μπορεί να μετρήσει επιταχύνσεις μέχρι +/-8g. (g είναι η επιτάχυνση της βαρύτητας με μια τυπική μέση τιμή 9,80665 m/sec²)



Το micro:bit ανιχνεύει κίνηση σε τρεις άξονες:

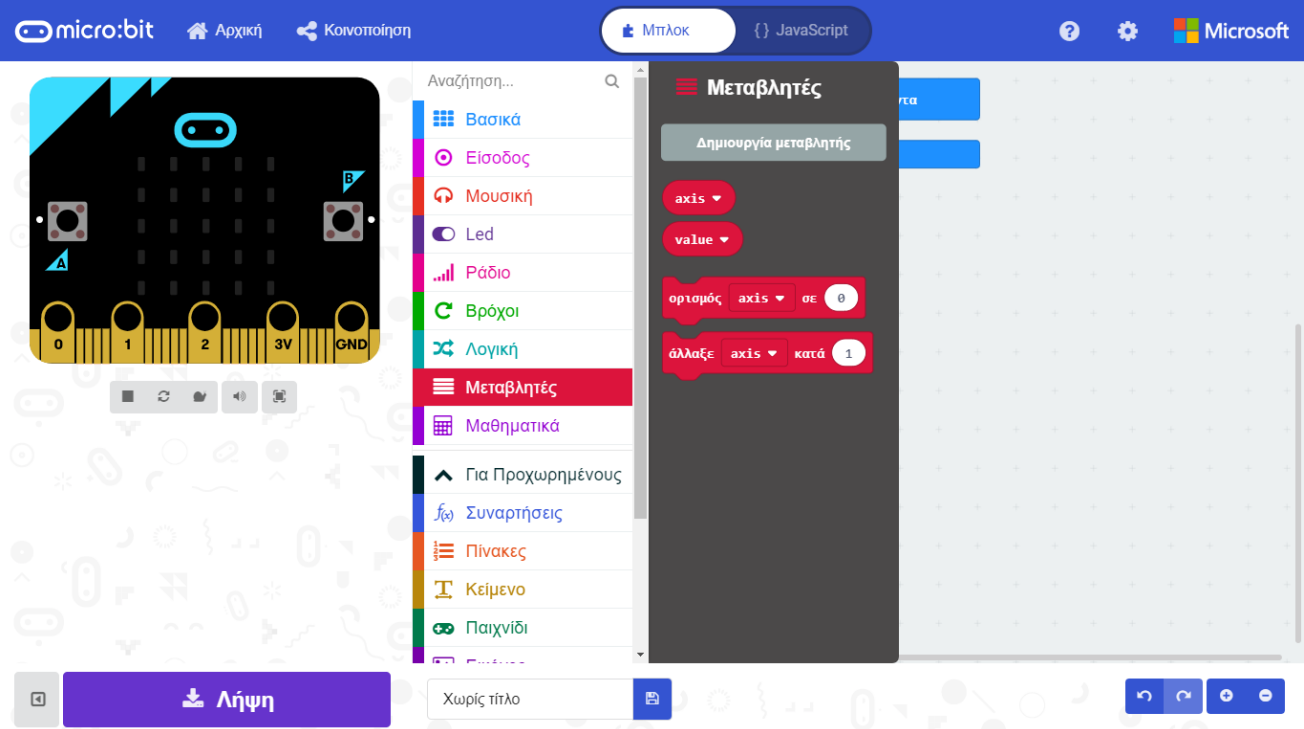
- X – Γέριμο αριστερά ή δεξιά.
- Y – Γέριμο μπροστά ή πίσω.
- Z – Μετακίνηση πάνω ή κάτω.



Χρήσιμη πληροφορία :Η προηγούμενη έκδοση του περιβάλλοντος προγραμματισμού υποστήριζε μόνο ακεραίους αριθμούς. Η τρέχουσα έκδοση όμως υποστηρίζει πραγματικούς αριθμούς (κινητής υποδιαστολής) και έχουν προστεθεί στην ενότητα μαθηματικά πολλές καινούριες συναρτήσεις.

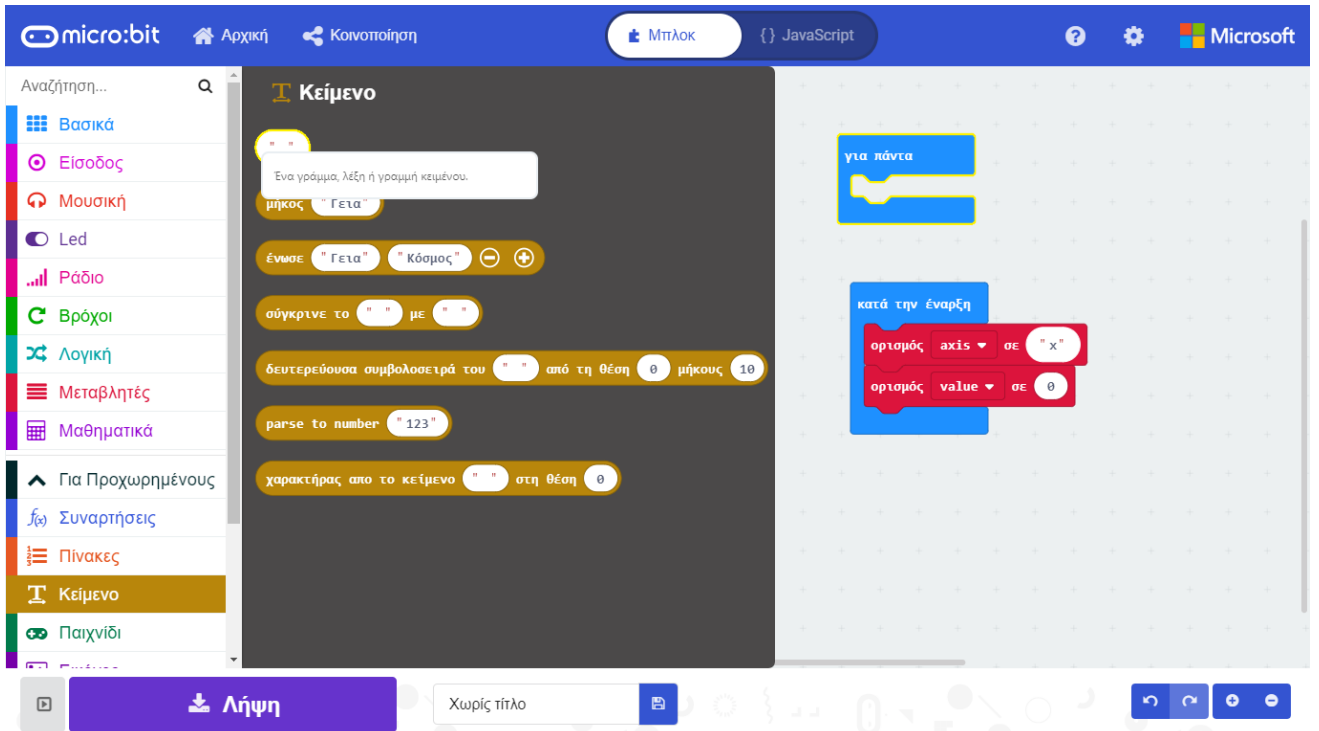
Βήματα της δραστηριότητας

1. Πρώτα ας δημιουργήσουμε ένα καινούριο Έργο. Θα χρειαστούμε 2 μεταβλητές για το πρόγραμμά μας. Μια μεταβλητή που θα δείχνει ποιον άξονα του επιταχυνσιόμετρου θα μετράμε (ας την πούμε «axis»), και μια μεταβλητή που θα αποθηκεύει την τιμή που διαβάζουμε από το επιταχυνσιόμετρο (ας την πούμε «value»). Τις μεταβλητές τις δημιουργούμε κάνοντας κλικ στο «Δημιουργία μεταβλητής» που βρίσκεται στην ενότητα μπλοκ «Μεταβλητές».




Ας χρησιμοποιήσουμε το μπλοκ «κατά την έναρξη» για να αρχικοποιήσουμε τις μεταβλητές σε


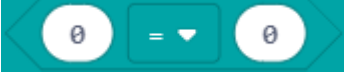


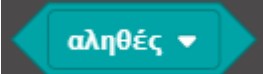
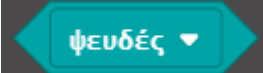
κάποιες λογικές τιμές.



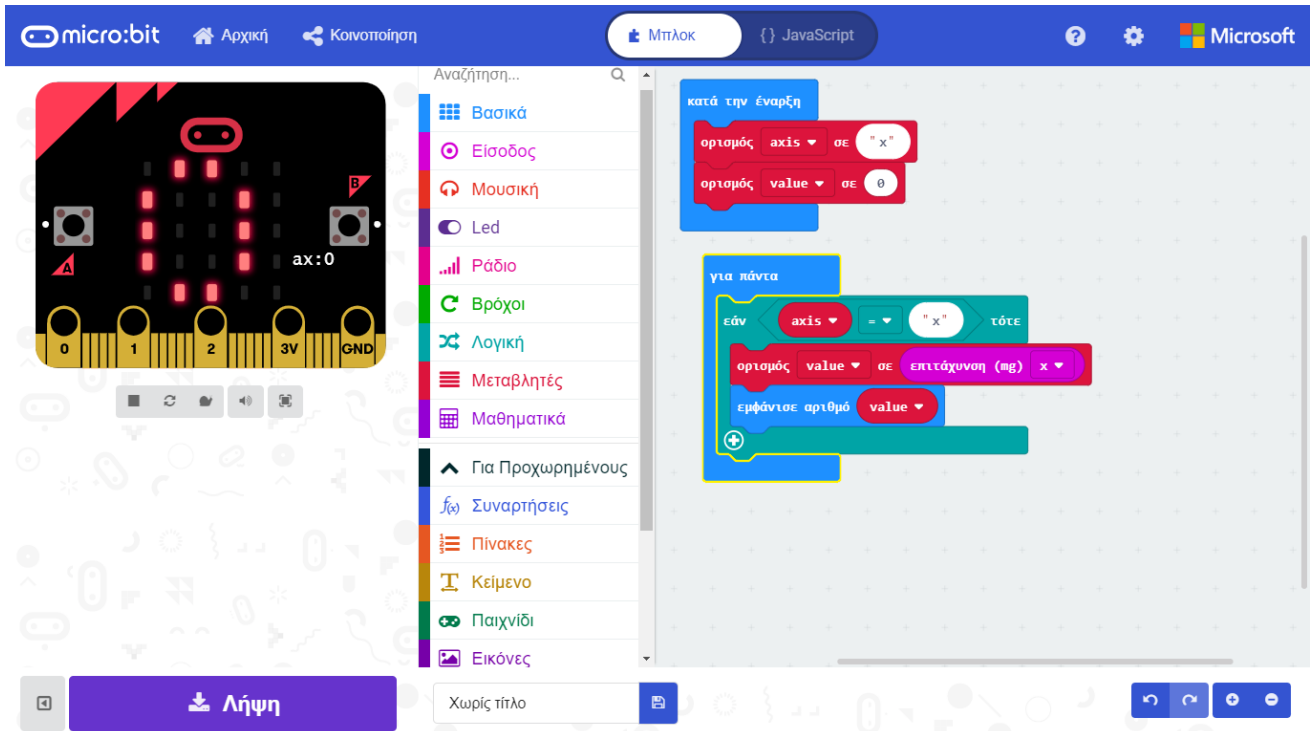
Για να αρχικοποιήσουμε τη μεταβλητή `axis` στην τιμή κειμένου «x», πρέπει να πάμε στην ενότητα «Για προχωρημένους» και στα μπλοκ «Κείμενο» να επιλέξουμε το μπλοκ με το τετραγωνάκι που περικλείεται από τα δυο εισαγωγικά και να το βάλουμε να κουμπώσει όπως στην προηγούμενη εικόνα:

2. Τώρα θα αρχίσουμε να μετράμε τις τιμές της επιτάχυνσης! Για αυτό θα χρειαστούμε κάποια καινούρια μπλοκ από την ενότητα «Λογική». Μέσα στην ενότητα «Λογική» μπορούμε να βρούμε μπλοκ που μας επιτρέπουν να διαλέξουμε διαφορετικά μονοπάτια στον κώδικά μας (τα μπλοκ «εάν») και να κάνουμε συγκρίσεις (να ελέγξουμε για ισότητα, να ελέγξουμε αν μια μεταβλητή έχει τιμή μεγαλύτερη, μικρότερη ή ίση από μια τιμή ή από μια άλλη μεταβλητή) και να δημιουργήσουμε σύνθετες λογικές προτάσεις. Μπορείτε να δείτε στον επόμενο πίνακα μια σύνοψη.

Όνομα Μπλοκ	Εμφάνιση Μπλοκ	Περιγραφή
Εάν .. τότε		Κάνε μια ενέργεια αν η έκφραση μετά το εάν είναι αληθής.

<p>Εάν ... τότε Αλλιώς εάν ...αλλιώς</p>		<p>Κάνε την ενέργεια μετά το «τότε» αν η έκφραση μετά το «εάν» είναι αληθής. Αν δεν είναι αληθής, έλεγξε την έκφραση που βρίσκεται στο αλλιώς εάν και εκτέλεσε αυτό που υπάρχει μετά το τότε. Αλλιώς, εκτέλεσε αυτό που υπάρχει μετά το αλλιώς. Κάνοντας κλικ στα σύμβολα (+) και (-) προσθέτουμε και αφαιρούμε κλάδους. Μπορούμε να έχουμε πολλά αλλιώς εάν, αν έχουμε πολλές διαφορετικές περιπτώσεις που πρέπει να ελέγξουμε.</p>
<p>Τελεστές σύγκρισης</p>		<p>Σύγκρινε δυο τιμές (μεταβλητές ή σταθερές). Χρησιμοποιήστε το μενού για να επιλέξετε τον τελεστή που θέλετε να χρησιμοποιήσετε. Οι διαθέσιμοι τελεστές σύγκρισης είναι: = (ίσο) ≠ (όχι ίσο) < (μικρότερο) ≤ (μικρότερο ή ίσο) > (μεγαλύτερο) ≥ (μεγαλύτερο ή ίσο)</p>
<p>Λογικό Και/Λογικό Ή</p>		<p>Χρησιμοποιήστε το κυλιόμενο μενού για να επιλέξετε μεταξύ λογικού Και (and) και λογικού Ή (or). Το αποτέλεσμα του λογικού Και είναι αληθές μόνο αν και οι δυο εκφράσεις είναι αληθείς. Το αποτέλεσμα του λογικού Ή είναι αληθές αν έστω μια από τις δυο εκφράσεις είναι αληθής.</p>
<p>Λογικό Όχι</p>		<p>Αντιστρέφει την τιμή. Αν αυτή ήταν αληθής γίνεται ψευδής και αντίστροφα.</p>
<p>Αληθές</p>		
<p>Ψευδές</p>		

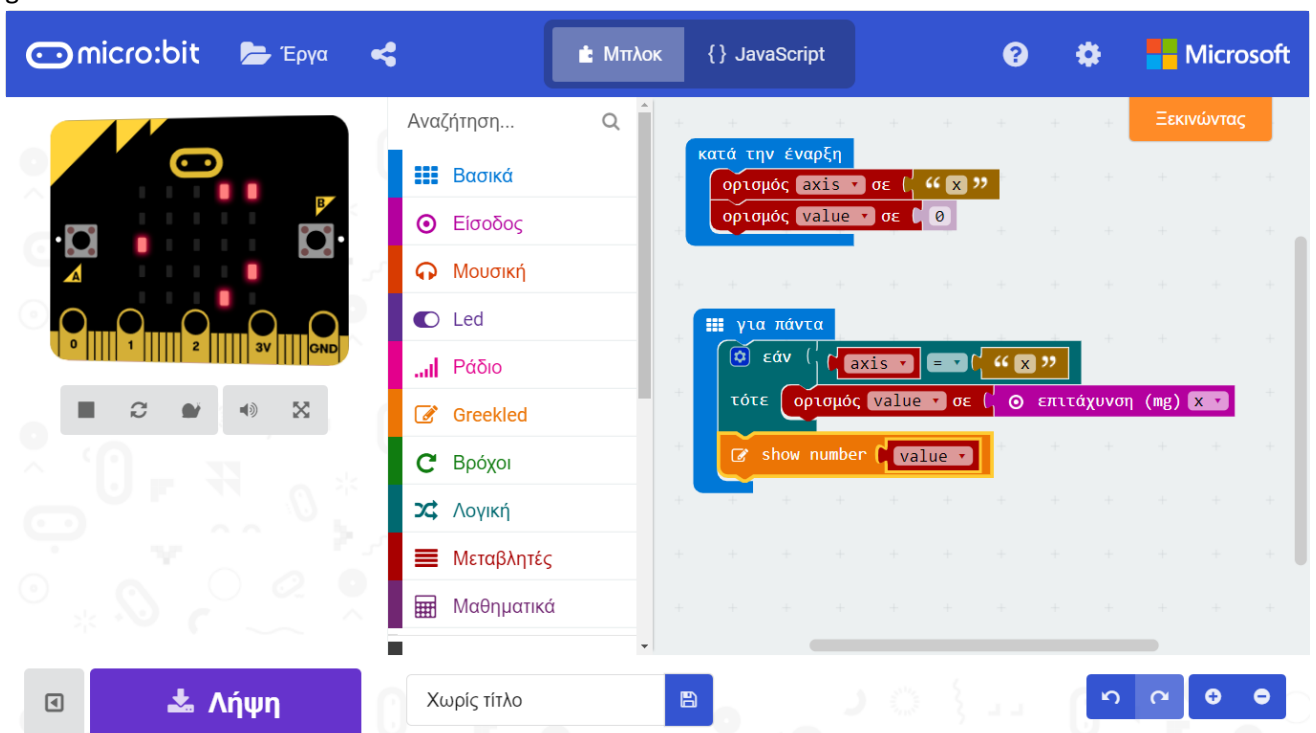
3. Τώρα ας διαβάσουμε το επιταχυνσιόμετρο. Το επιταχυνσιόμετρο μπορεί να διαβαστεί χρησιμοποιώντας το μπλοκ «επιτάχυνση (mg)» από τα μπλοκ της κατηγορίας «Είσοδος» Πρώτα θα φτιάξουμε ένα μπλοκ που διαβάζει τον x άξονα του επιταχυνσιόμετρου.



4. Ύστερα, απεικονίζουμε την τιμή της επιτάχυνσης του άξονα x.

Η τιμή της επιτάχυνσης κυλάει και δεν είναι τόσο εύκολο να διαβαστεί. Έτσι ίσως θα ήταν καλή ιδέα να χρησιμοποιήσουμε το πακέτο rxt-greekled-beta για να δείξουμε την τιμή στην οθόνη χωρίς να κυλάει.

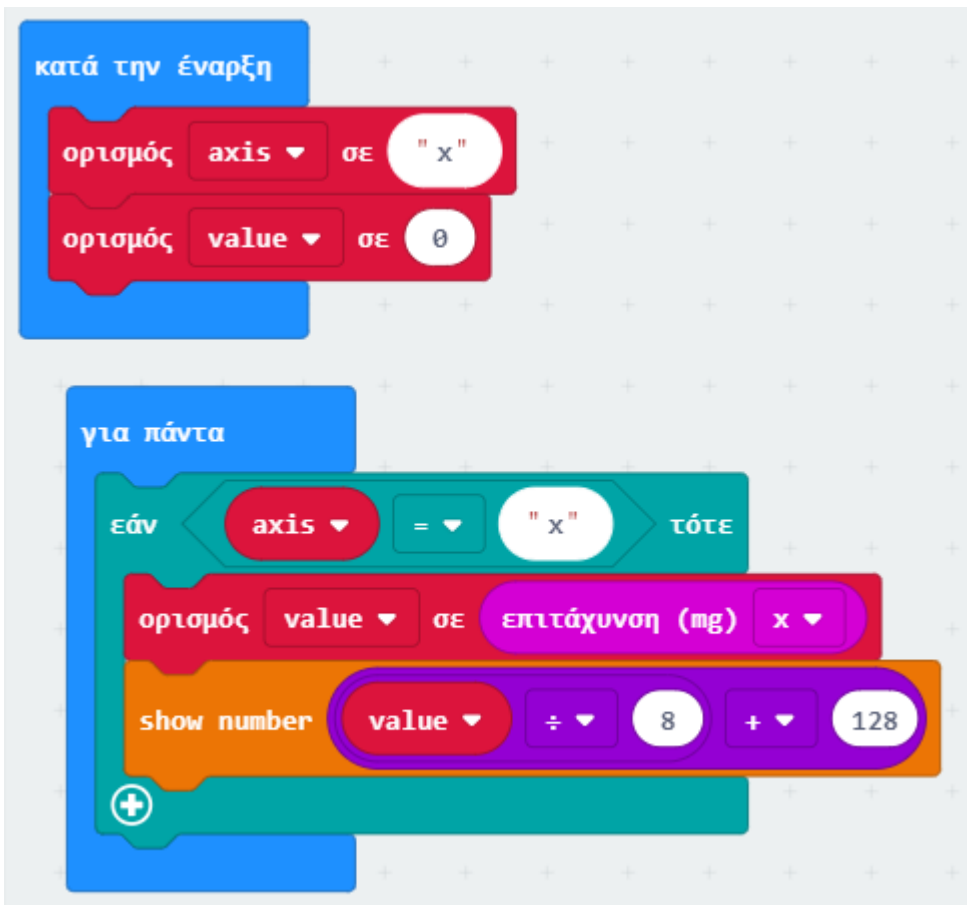
5. Έτσι, αντικαθιστούμε το μπλοκ «εμφάνισε αριθμό» με το μπλοκ «show number» του πακέτου «rxt-greekled-beta».



Παρόλα αυτά μερικοί αριθμοί είναι πολύ μεγάλοι για να τους εμφανίσουμε χωρίς να κυλίσει η οθόνη. Έτσι είναι καλό να κανονικοποιήσουμε τις τιμές πριν τις απεικονίσουμε. Πως μπορούμε να το κάνουμε αυτό; Οι τιμές που επιστρέφει το επιταχυνσιόμετρο βρίσκονται μεταξύ -1024 και 1024.

Έτσι αν διαιρέσουμε την τιμή αυτή με το 8 το εύρος τιμών γίνεται από -128 έως 128. Αν προσθέσουμε 128 στο αποτέλεσμα προκειμένου να αποφύγουμε το αρνητικό πρόσημο, το εύρος γίνεται 0 έως 255. Έτσι ας ενημερώσουμε τον κώδικα για να δούμε πως συμπεριφέρεται.

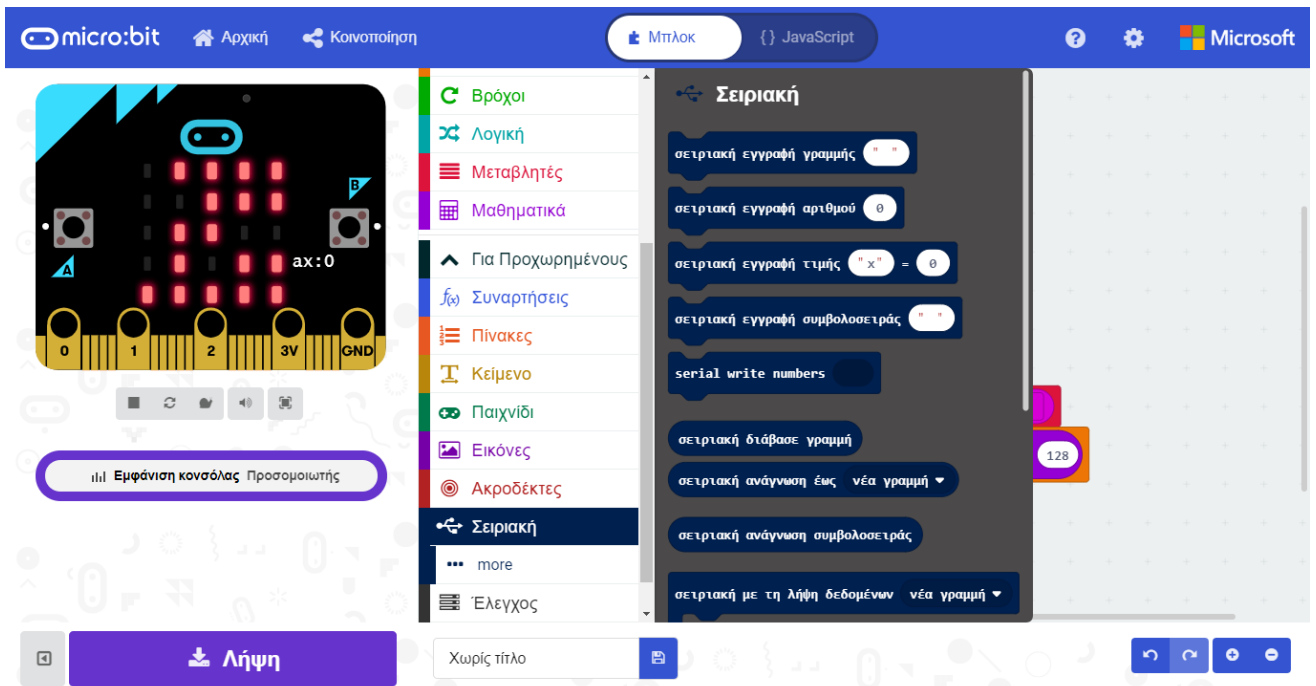
6. Έτσι ο κώδικας γίνεται:



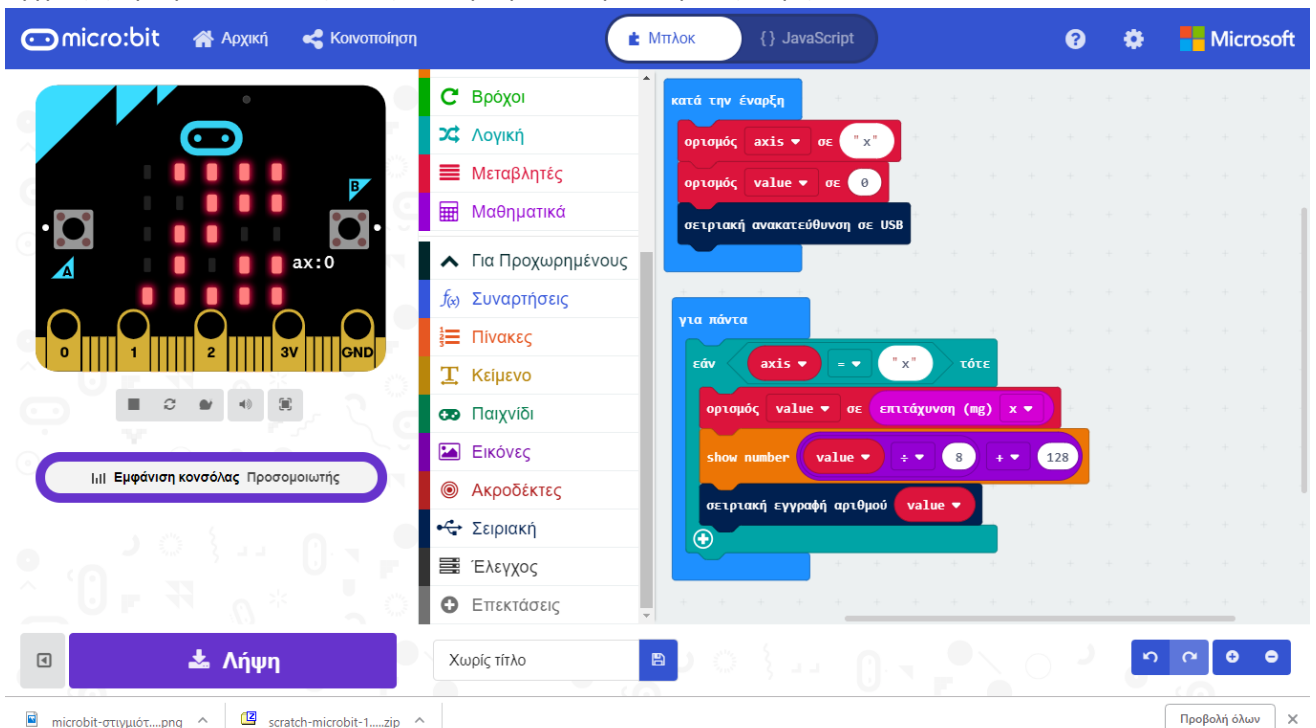
Και όπως μπορούμε να δούμε στον προσομοιωτή τώρα απεικονίζονται οι τιμές σωστά.

7. Ένας άλλος τρόπος να απεικονίσουμε τις τιμές είναι να τις στείλουμε στον υπολογιστή χρησιμοποιώντας την σειριακή θύρα. Όταν το micro:bit συνδέεται με έναν υπολογιστή με τη θύρα USB, εμφανίζεται στον υπολογιστή ως μια σειριακή θύρα. Έτσι το micro:bit μπορεί να στείλει δεδομένα στον υπολογιστή μέσω της σειριακής θύρας. Για να στείλουμε δεδομένα στη σειριακή πόρτα πρώτα επιλέγουμε την κατηγορία «Σειριακή» και προσθέτουμε ένα μπλοκ «σειριακή ανακατεύθυνση σε USB» μέσα στο μπλοκ «Κατά την έναρξη».

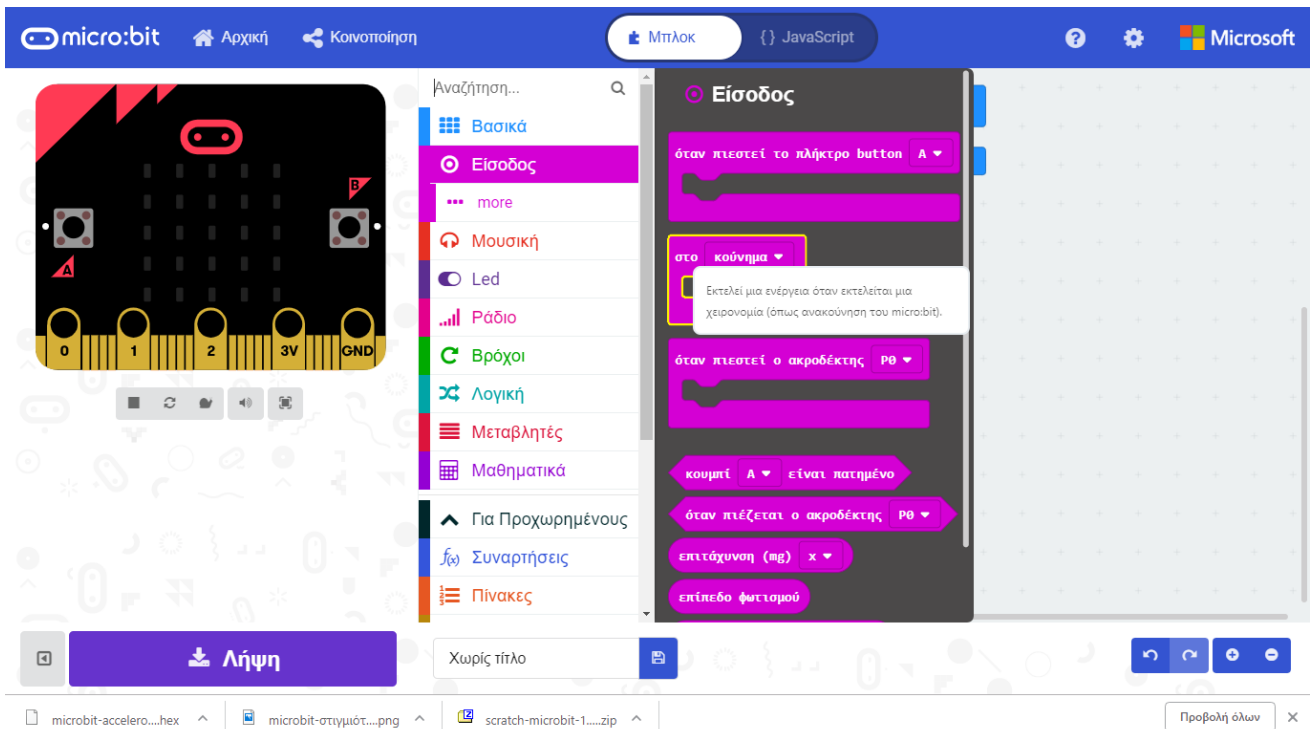
Χρήσιμη πληροφορία: Οι δομές «Εάν ... τότε» είναι βασικοί πυλώνες της υπολογιστικής σκέψης. Η δυνατότητα να ακολουθεί κανείς διαφορετικές λογικές διαδρομές ανάλογα με την τιμή μιας έκφρασης είναι αυτό που επιτρέπει στους υπολογιστές να πετύχουν υπέροχα πράγματα, καθώς αν δεν είχαν αυτή τη δυνατότητα δε θα διέφεραν από μια αριθμομηχανή. Η πρόοδος στην αρχιτεκτονική των υπολογιστών όπως για παράδειγμα η πρόβλεψη κλάδων (branch prediction) επιτρέπει στους σημερινούς μικροεπεξεργαστές να πετύχουν πολύ μεγάλες ταχύτητες επεξεργασίας δεδομένων και κάνουν εφικτές εφαρμογές Τεχνητής Νοημοσύνης.



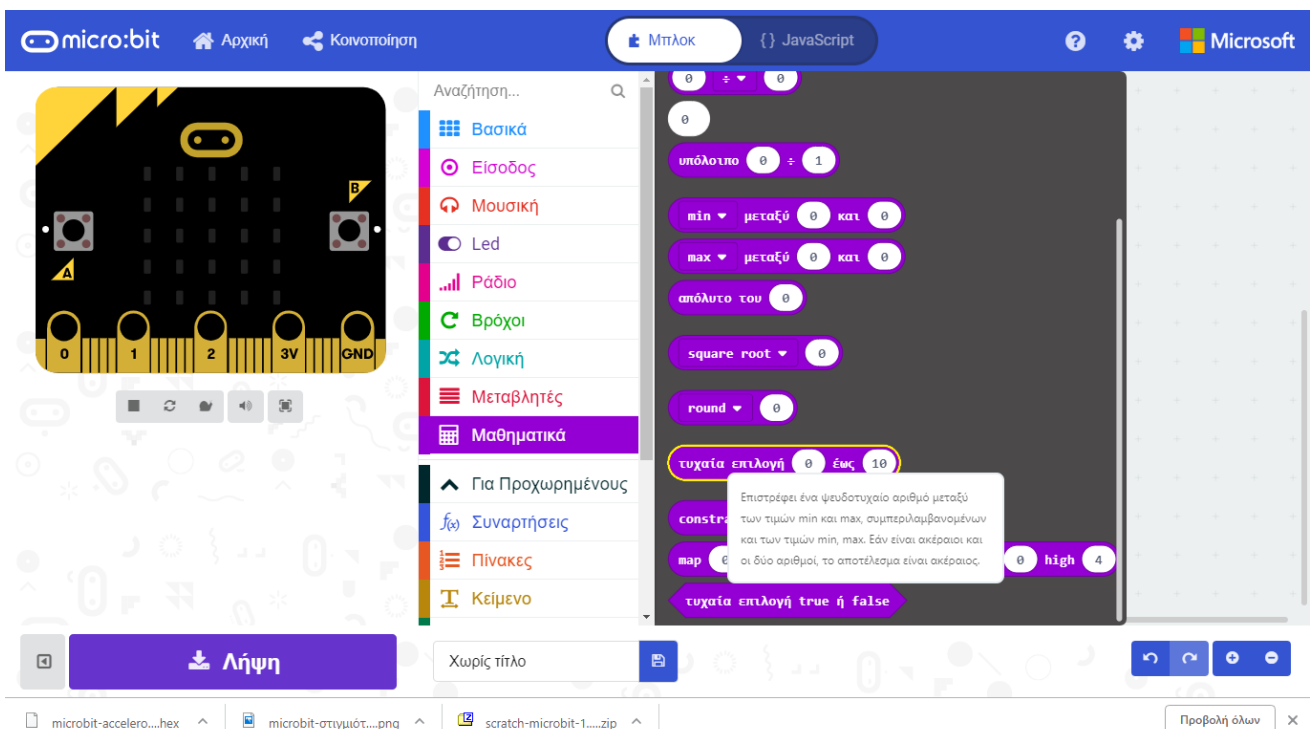
Για να στείλουμε τα δεδομένα στη σειριακή θύρα αρκεί να προσθέσουμε ένα μπλοκ «σειριακή εγγραφή αριθμού» και στη θέση του αριθμού θα βάλουμε την τιμή value.



Στον προσομοιωτή του micro:bit μπορούμε να δούμε τώρα τη γραφική παράσταση των τιμών:



Χρησιμοποιείτε την χειρονομία «στο κούνημα» που βρίσκεται στα μπλοκ εισόδου σε συνδυασμό με την «τυχαία επιλογή από το 0 έως ...» που βρίσκεται στα μπλοκ των μαθηματικών για να φτιάξετε ένα ηλεκτρονικό ζάρι που να δείχνει ένα τυχαίο αριθμό από το 1 έως το 6 όταν κουνάμε το micro:bit. Αν είστε τελειομανείς μπορείτε να κάνετε το micro:bit να δείχνει τους αριθμούς με τελίτσες, όπως στο πραγματικό ζάρι.



Links	
1	https://en.wikipedia.org/wiki/Conditional_(computer_programming)#If%E2%80%93then%E2%80%93else εάν ... τότε... αλλιώς.
2	https://en.wikipedia.org/wiki/Accelerometer Πως δουλεύουν τα επιταχυνσιόμετρα.

Δραστηριότητα Α6 – Ο αισθητήρας φωτός και η πυξίδα του micro:bit

Περίληψη: Σε αυτή τη δραστηριότητα θα δούμε δυο ακόμη αισθητήρες του micro:bit. Έναν αισθητήρα φωτός και μια ψηφιακή πυξίδα. Θα δούμε πως μπορούμε να δημιουργήσουμε ένα πρόγραμμα που μπορεί να βρει τη κατεύθυνση που έχει το περισσότερο φως και να τη δείξει σε μοίρες στην οθόνη.

Στόχοι:

- Κατανόηση του τρόπου λειτουργίας ενός αισθητήρα φωτός.
- Χρήση του αισθητήρα φωτός ως συσκευής εισόδου.
- Κατανόηση του τρόπου λειτουργίας της ψηφιακής πυξίδας.
- Χρήση της ψηφιακής πυξίδας ως συσκευής εισόδου.
- Χρήση μεταβλητών και δομών επιλογής (Εάν ... Τότε ... Αλλιώς).

Απαιτούμενο Υλικό

Εξαρτήματα	
1	Micro:bit board
2	Καλώδιο Micro USB (Για να συνδέσουμε το micro:bit στον υπολογιστή)

Η οθόνη από 5x5 led που έχει το micro:bit είναι επίσης και ένας αισθητήρας φωτός που μπορεί να ανιχνεύσει 256 διαφορετικά επίπεδα φωτισμού από 0 (λιγότερο φως) έως 255 (περισσότερο φως).

Το micro:bit είναι επίσης εξοπλισμένο με μια ψηφιακή πυξίδα (στην πραγματικότητα είναι ένα μαγνητόμετρο που μπορεί να μετρήσει την ένταση και τη κατεύθυνση των μαγνητικών πεδίων. Το μαγνητόμετρο μπορεί να επιστρέψει μια κατεύθυνση σε μοίρες που να αντιστοιχεί σε αυτή του μαγνητικού Βορρά, όπως μια πραγματική πυξίδα, ή μπορεί να επιστρέψει την ένταση του μαγνητικού πεδίου στους τρεις άξονες x, y και z, καθώς και το μέτρο της έντασης του μαγνητικού πεδίου. Η ακρίβεια της πυξίδας δεν είναι όμως πολύ καλή.

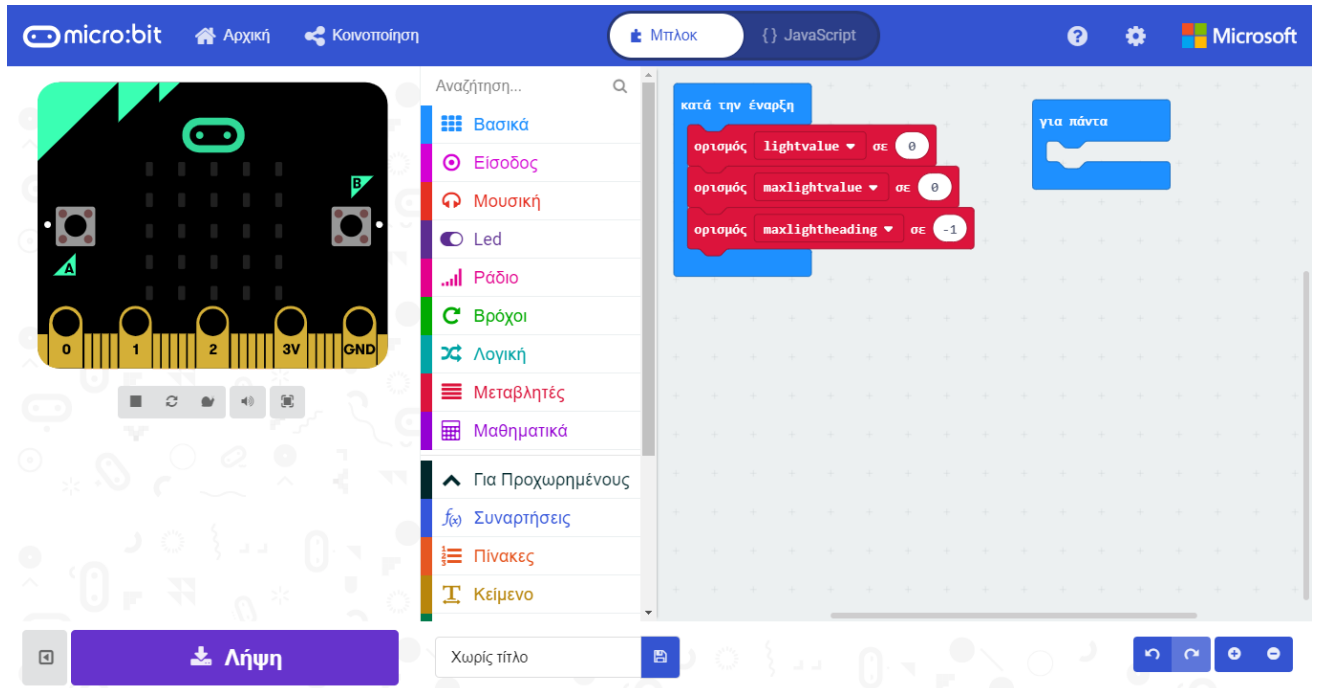
Θα χρησιμοποιήσουμε τον αισθητήρα φωτός και το μαγνητόμετρο για να φτιάξουμε ένα πρόγραμμα που μπορεί να μας δείξει την κατεύθυνση στην οποία το micro:bit λαμβάνει περισσότερο φως.

Σημείωση: Η ψηφιακή πυξίδα χρειάζεται βαθμονόμηση. Έτσι όταν ένα πρόγραμμα που χρησιμοποιεί την πυξίδα τρέχει για πρώτη φορά στο micro:bit, εμφανίζεται μια οθόνη βαθμονόμησης, στην οποία πρέπει να ακολουθήσουμε τις οδηγίες που γράφει η οθόνη του micro:bit.

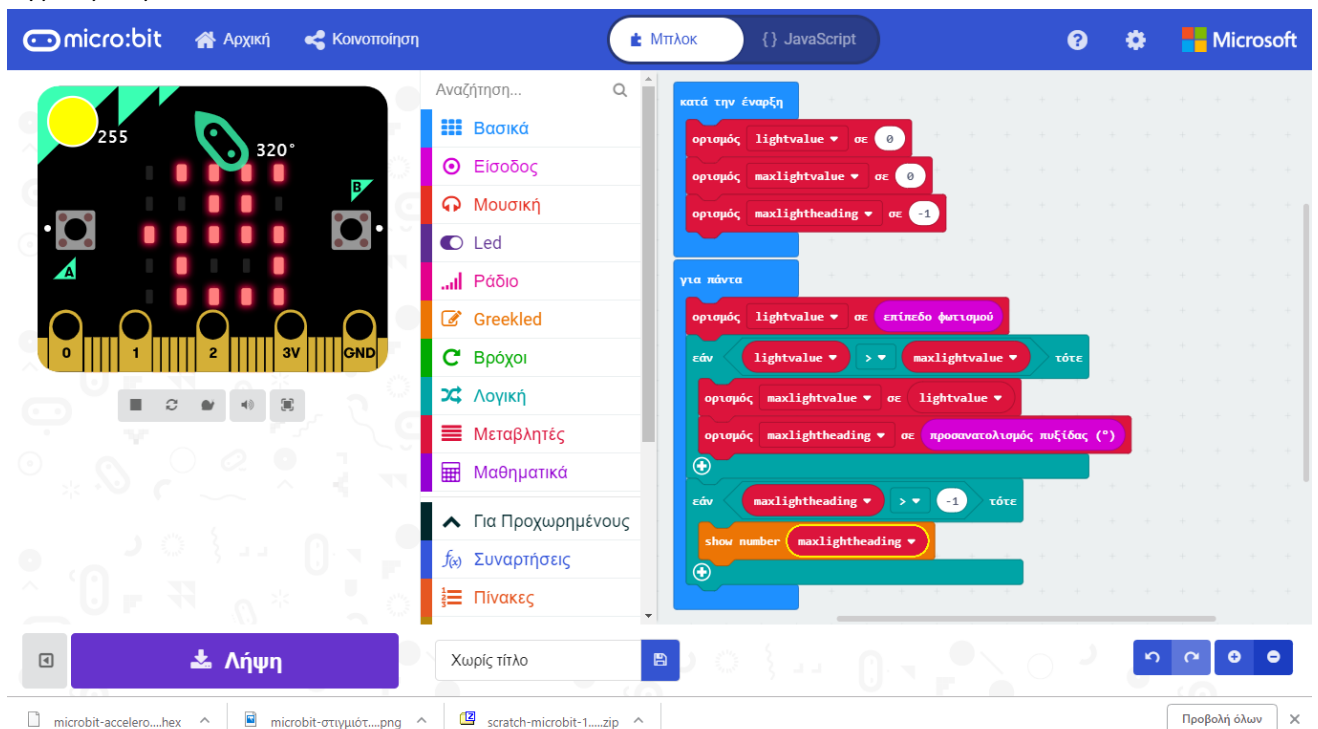
Βήματα της δραστηριότητας

1. Για αυτή τη δραστηριότητα θα χρησιμοποιήσουμε τρεις μεταβλητές, την `lightvalue` (κρατάει την τρέχουσα τιμή του φωτός που μετρήσαμε με τον αισθητήρα και την αρχικοποιούμε σε 0), `maxlightvalue` (κρατάει τη μέγιστη τιμή του φωτός που έχει μετρήσει ο αισθητήρας και αρχικοποιείται σε 0) και `maxlightheading` (Η κατεύθυνση στην οποία το φως έχει μεγαλύτερη τιμή,

αρχικοποιημένη σε -1).



2. Στη συνέχεια, στο «για πάντα» διαβάζουμε την τιμή του φωτός, την συγκρίνουμε με τη μέγιστη τιμή μέχρι στιγμής και εάν είναι μεγαλύτερη από τη μέγιστη τιμή, ενημερώνουμε την μέγιστη τιμή και διαβάζουμε την τιμή της κατεύθυνσης και ενημερώνουμε την τιμή του maxlighthreading με την τιμή της πυξίδας.



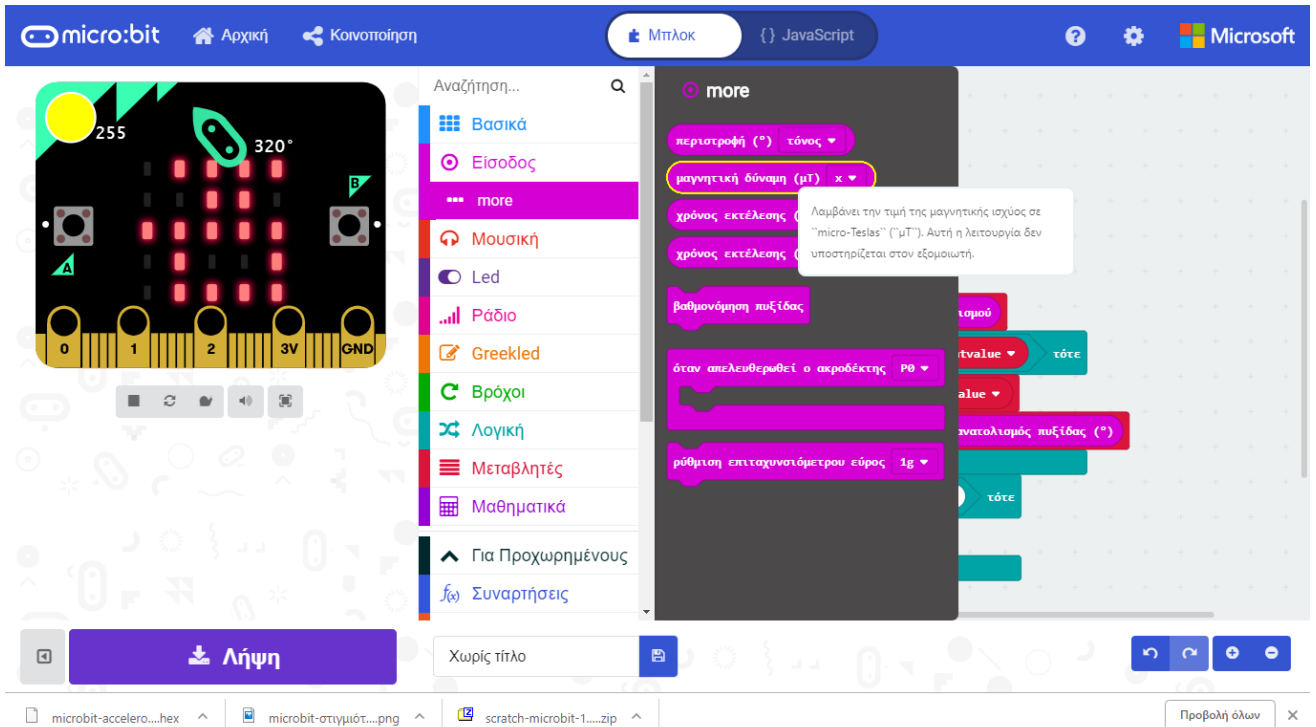
Για να δείξουμε την κατεύθυνση, θα χρησιμοποιήσουμε το μπλοκ «show number» (Πρέπει να προσθέσουμε το πακέτο GreekLed για να το κάνουμε αυτό.).

Ασκήσεις:

1. Ανεβάστε το πρόγραμμα σε ένα πραγματικό micro:bit. Δείτε πως δουλεύει η διαδικασία βαθμονόμησης και δοκιμάστε την εφαρμογή. Πως συμπεριφέρεται σε σύγκριση με τον

προσομοιωτή; Αν αλλάξετε στη σύγκριση lightvalue και maxlightvalue τον τελεστή σύγκρισης από μεγαλύτερο (>) σε μεγαλύτερο ίσο (≥) τι παρατηρείτε;

Για αυτούς που έχουν ... όρεξη: Δημιουργήστε έναν ανιχνευτή μετάλλων χρησιμοποιώντας το μπλοκ «μαγνητική δύναμη (μΤ) (δύναμη)» που βρίσκεται στα μπλοκ Εισόδου, στην επιλογή «Περισσότερα». Μπορείτε να απεικονίσετε τη δύναμη είτε χρησιμοποιώντας κατάλληλα την «show Number» του πακέτου GreekLed είτε χρησιμοποιώντας το μπλοκ «plot bar graph» που βρίσκεται στην περιοχή «Led». Η δεύτερη επιλογή είναι μάλλον ευκολότερη.



Σύνδεσμοι	
1	https://www.kitronik.co.uk/blog/how-an-ldr-light-dependent-resistor-works/ Πως δουλεύει μια φωτοαντίσταση.
2	https://makezine.com/projects/make-36-boards/how-to-use-leds-to-detect-light/ Χρησιμοποιώντας μια φωτοδίοδο ως ανιχνευτή φωτός. (Έτσι δουλεύει ο αισθητήρας φωτός του micro:bit)

Δραστηριότητα A7 – Χρησιμοποιούμε τη λειτουργία Ράδιο του micro:bit για να ανταλλάξουμε μηνύματα!

Περίληψη: Σε αυτή τη δραστηριότητα θα δούμε πως μπορούμε να χρησιμοποιήσουμε τη λειτουργία Ράδιο του micro:bit για να ανταλλάξουμε μηνύματα μεταξύ 2 ή περισσότερων micro:bit

Στόχοι:

- Κατανόηση του πως λειτουργεί η λειτουργία Ράδιο
- Χρήση των κουμπιών A και B για να γράψουμε και να στείλουμε μηνύματα σε άλλα micro:bit.

Απαιτούμενος εξοπλισμός

Components	
1	Micro:bit board (x2)
2	Καλώδιο Micro USB (για σύνδεση του Micro:bit με τον υπολογιστή)

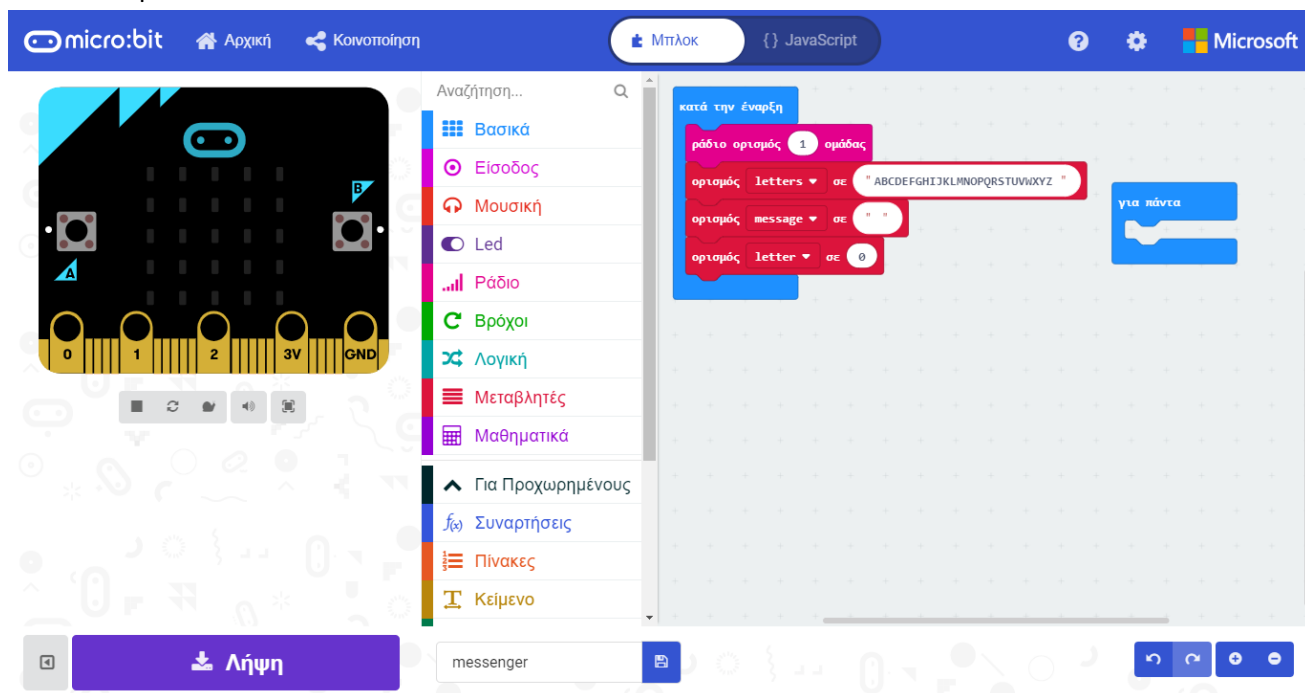
Το micro:bit είναι εξοπλισμένο με την λεγόμενη λειτουργία «Ράδιο». Η λειτουργία «Ράδιο» είναι ένας τρόπος ασύρματης ανταλλαγής μηνυμάτων μεταξύ 2 ή περισσότερων micro:bit

Σημείωση: Το micro:bit υποστηρίζει επίσης επικοινωνία Bluetooth Low Energy (BLE). Όμως δεν μπορεί να χρησιμοποιηθεί ταυτόχρονα με τη λειτουργία «Ράδιο». Επίσης το Bluetooth χρειάζεται κάπως περίπλοκες ρυθμίσεις, γι' αυτό και για απλή επικοινωνία η λειτουργία «Ράδιο» είναι μάλλον η καλύτερη επιλογή.

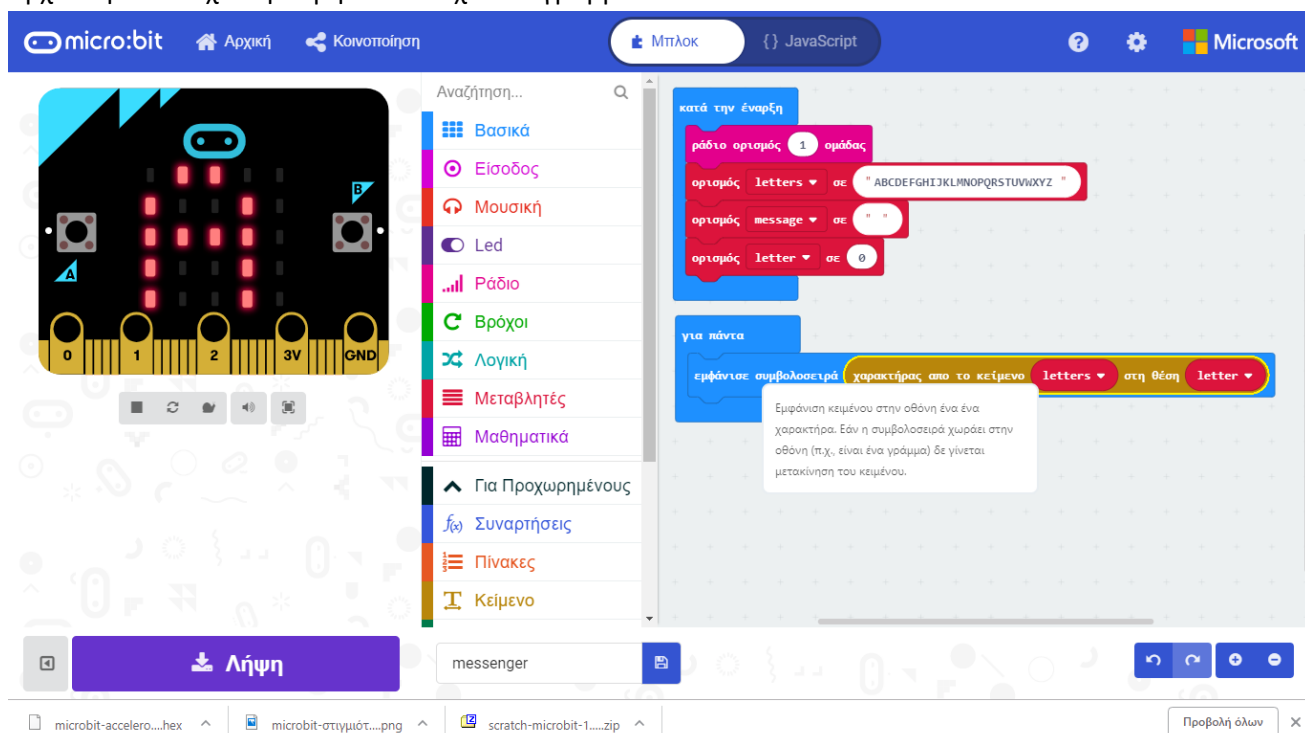
Βήματα της δραστηριότητας

1. Για αυτή τη δραστηριότητα θα χρειαστούμε 3 μεταβλητές: Η «letters» θα κρατάει τα σύμβολα που θα χρησιμοποιήσουμε για να ανταλλάξουμε μηνύματα. Αρχικά θα χρησιμοποιήσουμε Αγγλικά κεφαλαία γράμματα και το κενό, έτσι η μεταβλητή θα αρχικοποιηθεί ως "ABCDEFGHIJKLMNOPQRSTUVWXYZ ". Η «message» θα περιέχει το μήνυμα που θα στείλουμε. Αρχικά θα είναι κενό. Τέλος η μεταβλητή «letter» δείχνει ποιο σύμβολο έχουμε επιλέξει για να προσθέσουμε στο μήνυμά μας, ως δείκτη στην μεταβλητή «letters». Αρχικά θα έχει την τιμή 0 που αντιστοιχεί στον χαρακτήρα A της μεταβλητής «letters». Αν είχε την τιμή 1 θα αντιστοιχούσε στο B , το 25 στο Z και το 26 στο κενό " ". Έτσι ξεκινάμε αρχικοποιώντας της μεταβλητές στο μπλοκ «κατά την έναρξη». Επίσης θα αρχικοποιήσουμε τη λειτουργία radio χρησιμοποιώντας το μπλοκ «ράδιο ορισμός 1». Ο αριθμός αυτός μας επιτρέπει να φτιάξουμε ένα γκρουπ, μια ομάδα από micro:bit που μπορούν να επικοινωνήσουν μεταξύ τους καθώς όλα τα micro:bit που ανήκουν στο ίδιο γκρουπ λαμβάνουν τα μηνύματα που στέλνονται στο γκρουπ. Αν 2 micro:bit ανήκουν σε διαφορετικά γκρουπ, πχ. Το ένα στο γκρουπ 1 και το άλλο στο γκρουπ 5, δεν μπορούν να

επικοινωνήσουν.

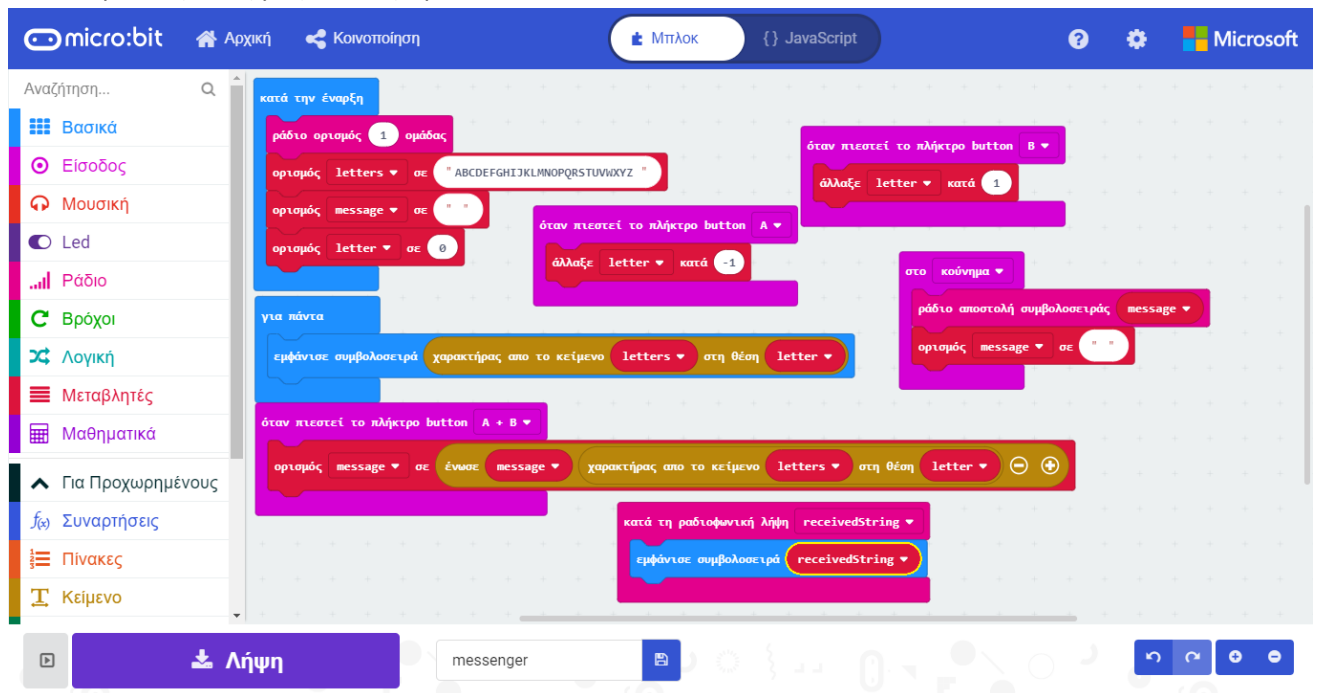


- 2. Στο μπλοκ «για πάντα» δείχνουμε στην οθόνη το επιλεγμένο γράμμα που δείχνει η μεταβλητή «letter». Αρχικά η letter έχει την τιμή 0 και δείχνει το γράμμα «Α».



- 3. Θέλουμε να μπορούμε να αλλάζουμε το επιλεγμένο γράμμα χρησιμοποιώντας τα πλήκτρα «Α» και «Β». Και θέλουμε να μπορούμε να προσθέσουμε το επιλεγμένο γράμμα στο μήνυμα που θέλουμε να

στείλουμε πιέζοντας μαζί τα πλήκτρα «Α» και «Β».

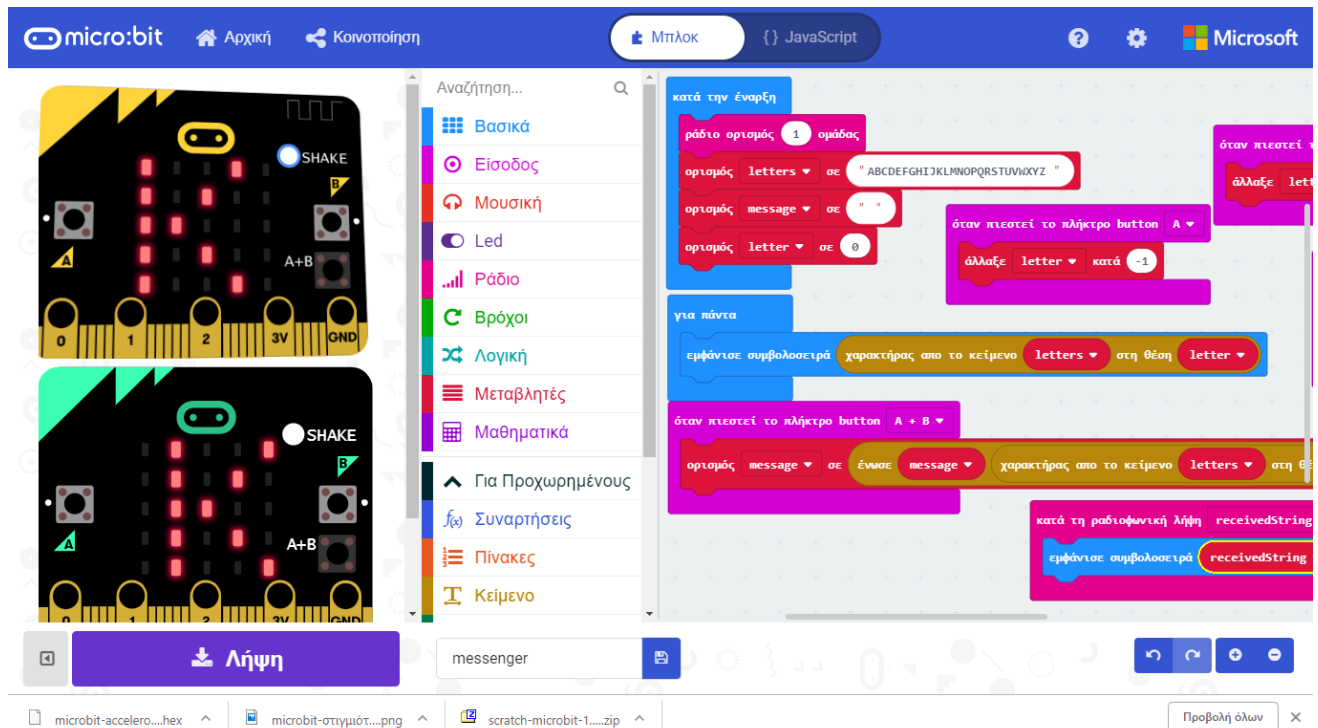


4. Για να στείλουμε το μήνυμα χρησιμοποιούμε την χειρονομία «κούνημα»:

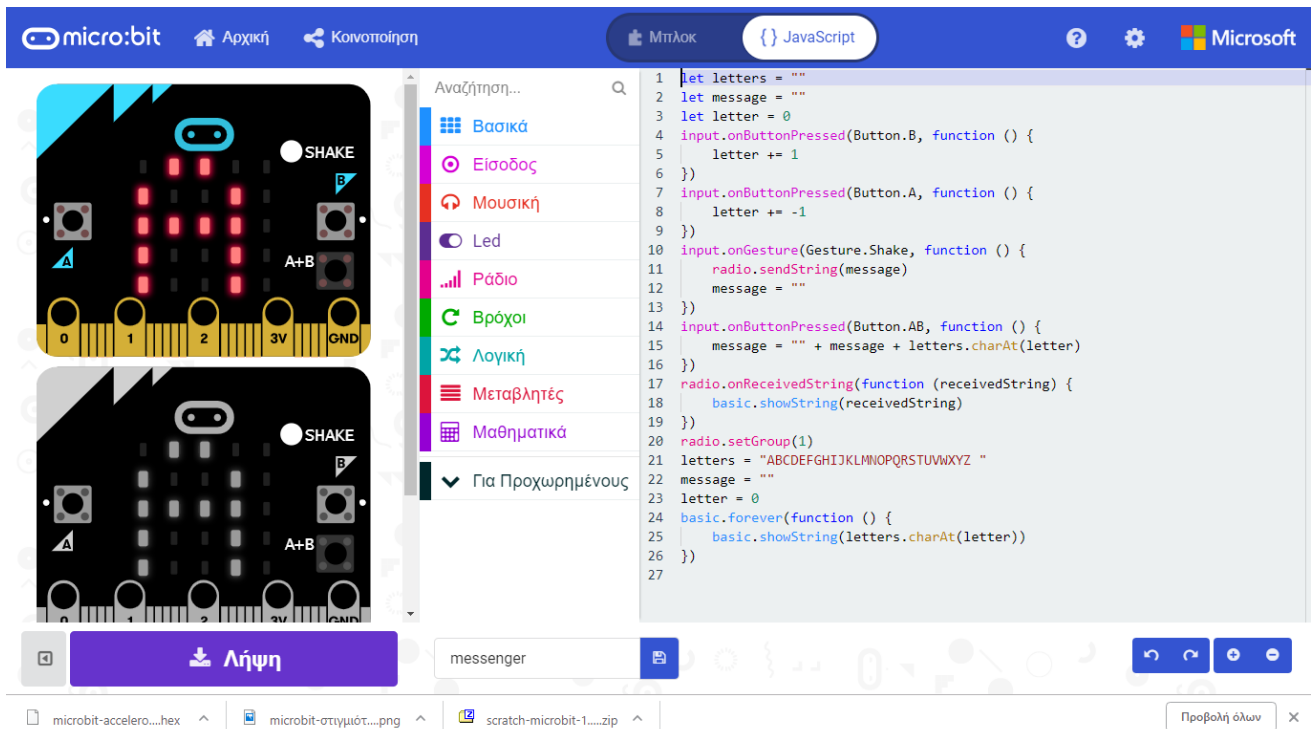
Σημειώστε ότι αφού στείλουμε το μήνυμα το κάνουμε πάλι κενό και ξεκινάμε πάλι δείχνοντας στο «Α».

5. Για να λάβουμε το μήνυμα, χρησιμοποιούμε το μπλοκ «κατά τη ραδιοφωνική λήψη receivedString»:

6. Τώρα το πρόγραμμά μας είναι ολοκληρωμένο και μπορούμε να το δοκιμάσουμε. Σημειώστε ότι ο προσομοιωτής υποστηρίζει τη λειτουργικότητα «Ράδιο», έτσι μας δείχνει και ένα δεύτερο micro:bit ώστε να μπορέσουμε να δούμε πως δουλεύει η επικοινωνία.



7. Αν κάνετε κλικ στο κουμπί «{ } Javascript» θα δείτε πως το πρόγραμμα είναι δομημένο σε μορφή κειμένου.



Είναι περίπου 26 γραμμές κώδικα javascript. Σημειώστε ότι μπορείτε να μετακινηθείτε μεταξύ μπλοκ και javascript απλά κάνοντας κλικ στο αντίστοιχο κουμπί. Όσο τα προγράμματά μας μεγαλώνουν σε πολυπλοκότητα, το περιβάλλον προγραμματισμού με μπλοκ γίνεται λιγότερο ελκυστικό και η χρήση κώδικα σε μορφή κειμένου αποκτά προβάδισμα.

Ασκήσεις:

1. Ανεβάστε τον κώδικα σε δύο micro:bit και στείλτε μηνύματα από το ένα στο άλλο!
2. Προσθέστε έναν έλεγχο όταν πατάτε τα πλήκτρα «Α» και «Β» έτσι ώστε το letter να μη μπορεί να παίρνει τιμές μικρότερες από το 0 και μεγαλύτερες από το 26. Ανεβάστε τον τροποποιημένο κώδικα στο micro:bit.
3. Τροποποιήστε τον προηγούμενο κώδικα έτσι ώστε αν πιέσουμε το πλήκτρο «Α» όταν έχουμε επιλέξει τον πρώτο χαρακτήρα να πηγαίνει στον τελευταίο και όταν πατάμε το πλήκτρο «Β» όταν έχουμε επιλέξει τον τελευταίο χαρακτήρα, να πηγαίνουμε στον πρώτο.
4. Τι θα μπορούσατε να προσθέσετε ώστε να κάνετε το πρόγραμμα πιο εύχρηστο;
5. Μετατρέψτε τον κώδικα ώστε να μπορεί να στέλνει ελληνικούς χαρακτήρες. Τι αλλαγές πρέπει να κάνετε;

Για αυτούς που έχουν ... όρεξη: Προσθέστε τη δυνατότητα να μπορεί ο χρήστης να σβήσει έναν χαρακτήρα από το μήνυμα, όπως όταν γράφουμε στον υπολογιστή και πατάμε το πλήκτρο backspace για να σβήσουμε έναν χαρακτήρα. Πως θα το πετύχετε;

Σύνδεσμοι			
1	https://en.wikipedia.org/wiki/Communication_protocol	Υλοποιήσαμε	ένα
	τηλεπικοινωνιακό πρωτόκολλο!		

Δραστηριότητα A8 – Συνδέοντας έναν υπερηχητικό αισθητήρα απόστασης

Περίληψη: Σε αυτή τη δραστηριότητα θα δούμε πως μπορούμε να χρησιμοποιήσουμε έναν υπερηχητικό αισθητήρα απόστασης για να μετρήσουμε απόσταση από ένα εμπόδιο και να απεικονίσουμε την απόσταση στην οθόνη του micro:bit. Θα δούμε πως μπορούμε να συνδέσουμε τον αισθητήρα και να χρησιμοποιήσουμε κάποια απλά ηλεκτρονικά εξαρτήματα όπως η πλακέτα πρωτοτυποποίησης (solderless breadboard), οι αντιστάσεις, τα καλώδια και η εξωτερική τροφοδοσία.

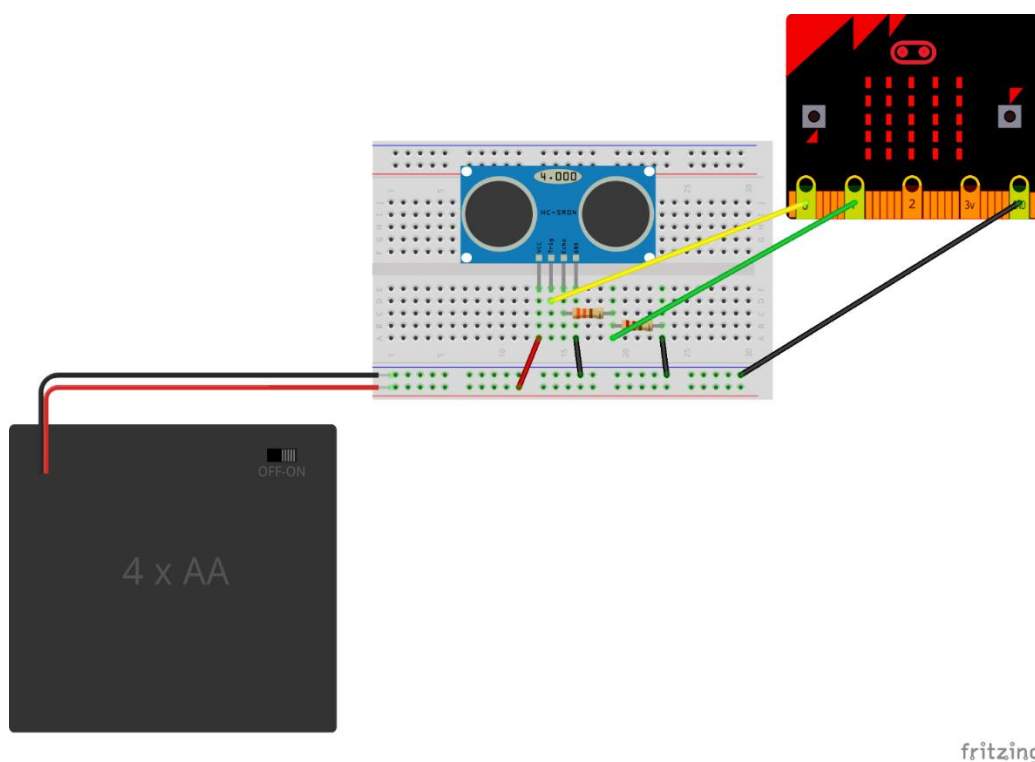
Στόχοι:

- Εξοικείωση με κάποια βασικά ηλεκτρονικά εξαρτήματα και εργαλεία όπως οι αντιστάσεις και οι πλακέτες πρωτοτυποποίησης.
- Προσαρμογή εξαρτημάτων στο micro:bit.
- Πως λειτουργεί η μέτρηση απόστασης με έναν υπερηχητικό αισθητήρα και πως μπορούμε να χρησιμοποιήσουμε μια έτοιμη βιβλιοθήκη για να μετρήσουμε την απόσταση χρησιμοποιώντας τις ιδιότητες του ήχου.
- Απεικόνιση των δεδομένων που διαβάστηκαν από τον υπερηχητικό αισθητήρα απόστασης σε πραγματικό χρόνο στην οθόνη του micro:bit.

Απαιτούμενο Υλικό

Εξαρτήματα	
1	Micro:bit board
2	Καλώδιο Micro USB (για να συνδέσουμε το Micro:bit με τον υπολογιστή)
3	HC-SR04 Ultrasonic sensor
4	Πλακέτα πρωτοτυποποίησης
5	Αντιστάσεις (1x330 Ohms, 1x470 Ohms)
6	Καλωδιάρια
7	Μπαταριοθήκη με έξοδο 4.8 Volts

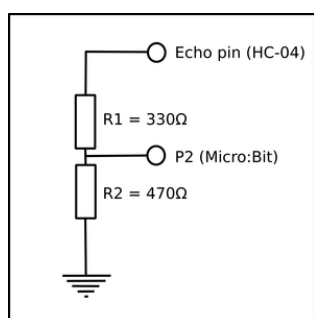
Στο επόμενο σχεδιάγραμμα βλέπουμε πως είναι συνδεδεμένα τα εξαρτήματα στο micro:bit



Το σχεδιάγραμμα μπορεί να μοιάζει κάπως πολύπλοκο για έναν αρχάριο στα ηλεκτρονικά μα είναι αρκετά απλό. Ας το δούμε κομμάτι - κομμάτι.

1. Το micro:bit δουλεύει στα 3,3 Volts, τόσο στην τροφοδοσία του, όσο και στις εισόδους και εξόδους των ακροδεκτών του. Αυτό σημαίνει ότι αν του στείλουμε περισσότερα από 3,3Volts σε κάποιο από τους ακροδέκτες του μπορεί να πάθει ζημιά. Επίσης, το micro:bit δεν μπορεί να τροφοδοτήσει τον αισθητήρα υπερήχων με τα περίπου 5 Volts που χρειάζεται για τροφοδοσία..
2. Ο αισθητήρας HC-SR04 είναι πολύ συνηθισμένος. Έχει 4 ακροδέκτες που συνήθως έχουν τις ενδείξεις VCC (τροφοδοσία), ring ή trig (σκανδαλισμός) , echo (ηχώ) and GND (γείωση). Όμως, οι περισσότερες εκδόσεις του HC-SR04 χρειάζονται τουλάχιστον 4.8Volts που είναι πολύ περισσότερα από τα 3 έως 3,3 Volts που το micro:bit μπορεί να παρέχει γι' αυτό του παρέχουμε τροφοδοσία με μπαταριοθήκη. Χρησιμοποιούμε επαναφορτιζόμενες μπαταρίες NiMH (μεγέθους AA ή AAA) που παρέχουν 1,2 Volts η καθεμία, έτσι πρέπει να χρησιμοποιήσουμε μια μπαταριοθήκη που παίρνει 4 μπαταρίες αφού $4 \times 1.2 \text{Volts} = 4.8 \text{Volts}$. Η γείωση (ο αρνητικός ακροδέκτης που είναι σημειωμένος με -) της μπαταριοθήκης πρέπει να είναι πάντα συνδεδεμένη στη γείωση (GND) του micro:bit. Ένας αισθητήρας απόστασης υπερήχων δουλεύει στέλνοντας έναν σύντομο υπερηχητικό παλμό όταν ο ακροδέκτης trig ενεργοποιείται θέτοντάς τον σε κατάσταση ΥΨΗΛΟ. Αν ο παλμός συναντήσει ένα εμπόδιο, αντανακλάται πίσω και εντοπίζεται από τον αισθητήρα και ενεργοποιεί τον ακροδέκτη echo θέτοντάς τον σε κατάσταση ΥΨΗΛΟ. Μετρώντας το χρόνο που μεσολάβησε μεταξύ της αποστολής του παλμού με την ενεργοποίηση του trig ακροδέκτη και της αλλαγής της κατάστασης του ακροδέκτη echo σε ΥΨΗΛΟ μπορούμε να προσδιορίσουμε την απόσταση από το εμπόδιο. Έτσι μπορούμε να κάνουμε τους ακόλουθους υπολογισμούς:

$2 * S(\text{απόσταση από το εμπόδιο}) = \text{χρόνος} \times \text{ταχύτητα του ήχου}$. Έτσι προκύπτει ότι η απόσταση από το



εμπόδιο είναι $S = (\text{χρόνος} \times \text{ταχύτητα του ήχου στον αέρα}) / 2$. Αν για παράδειγμα μετρούσαμε το χρόνο μεταξύ την αποστολή του παλμού στον ακροδέκτη trig και τη λήψη της απάντησης στον ακροδέκτη echo ως 0.005 δευτερόλεπτα (5 milliseconds) τότε η απόσταση ως το εμπόδιο είναι $(0.005 \text{ sec} \times 343 \text{ m/sec}) / 2 = 0.8575 \text{ m} = \text{περίπου } 85 \text{ εκατοστά}$. Φυσικά χρησιμοποιούμε την παραδοχή ότι η ταχύτητα του ήχου είναι σταθερή και ίση με 343 m/sec. Στην πραγματικότητα η ταχύτητα του ήχου στον αέρα μεταβάλλεται με τη θερμοκρασία την ατμοσφαιρική πίεση κλπ., μα για τις

περισσότερες περιπτώσεις τα αποτελέσματα της παραδοχής αυτής δε μας δημιουργούν σφάλμα μεγαλύτερο από 1 με 2 εκατοστά.

Ο ακροδέκτης trig μπορεί να ενεργοποιηθεί με το micro:bit παρόλο που αυτό μπορεί να στείλει μόνο 3,3 Volts, γιατί στις περισσότερες ηλεκτρονικές διατάξεις αρκεί να στείλουμε σε έναν ακροδέκτη εισόδου μια τάση ίση με 2/3 την τιμή της τάσης τροφοδοσίας για να ενεργοποιηθούν. Όμως ο ακροδέκτης echo όταν ενεργοποιείται έχει τάση κοντά στα 4,8 Volts που θα μπορούσε να προκαλέσει ζημιά στον ακροδέκτη εισόδου του micro:bit. Για να το διορθώσουμε αυτό μπορούμε να χρησιμοποιήσουμε μια απλή διάταξη με αντιστάσεις που λέγεται διαιρέτης τάσης που χρησιμοποιεί απλές αντιστάσεις σε σειρά. Ένας διαιρέτης τάσης «διαιρεί» μια τάση σε 2 ή περισσότερες τάσεις που προσδιορίζονται από την αναλογία τιμών των αντιστάσεων. Κοιτάξτε το επόμενο σχεδιάγραμμα:

Οι τιμές των αντιστάσεων μπορούν να προσδιοριστούν χρησιμοποιώντας τον ακόλουθο μαθηματικό τύπο:

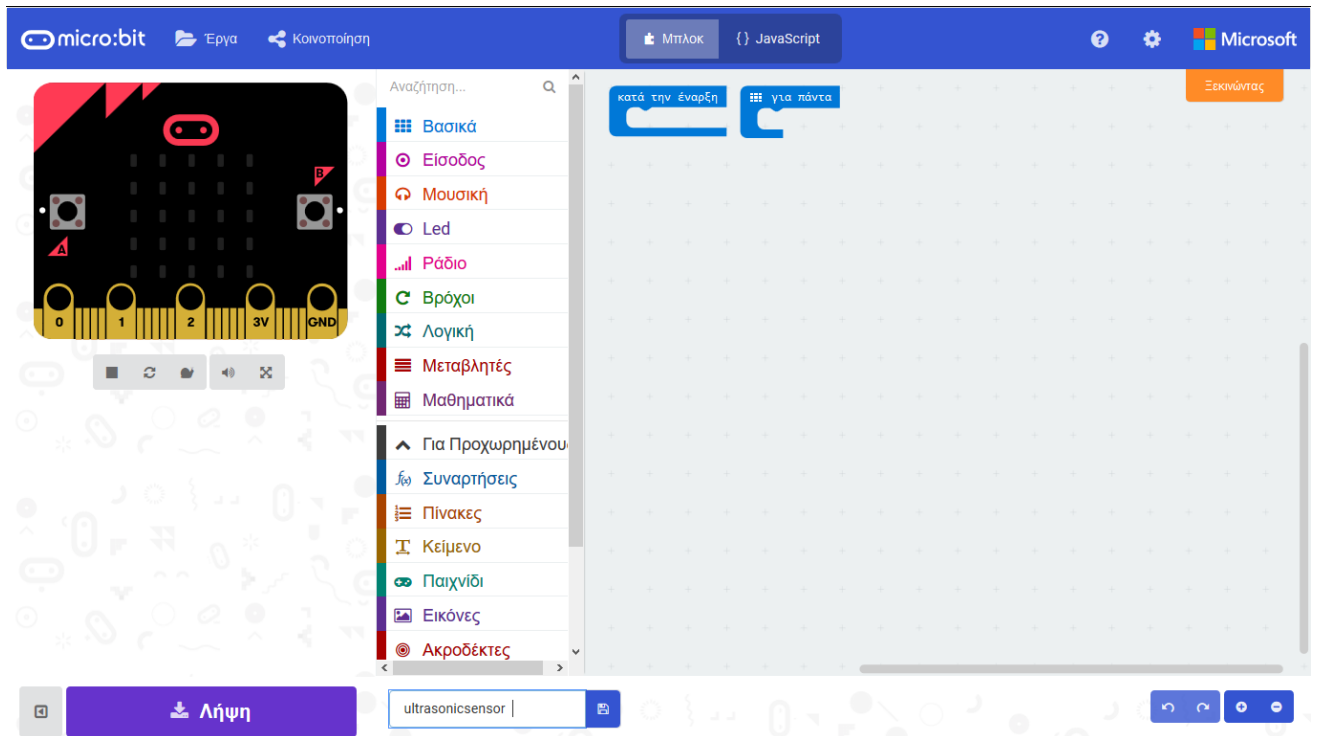
$$V_{out} = \frac{R_2}{R_1 + R_2} V_{in}$$

Όπου Vin είναι η τάση που επιστρέφει ο ακροδέκτης echo και Vout είναι η τάση που θέλουμε να στείλουμε στο micro:bit. Χρησιμοποιώντας τις συνηθισμένες τιμές αντιστάσεων 330 ohms και 470 ohms παίρνουμε για τάση εισόδου 4,8 Volts την τιμή 2,82 Volts έτσι το micro:bit δεν πρέπει να έχει πρόβλημα να ανιχνεύσει την τάση στον ακροδέκτη echo.

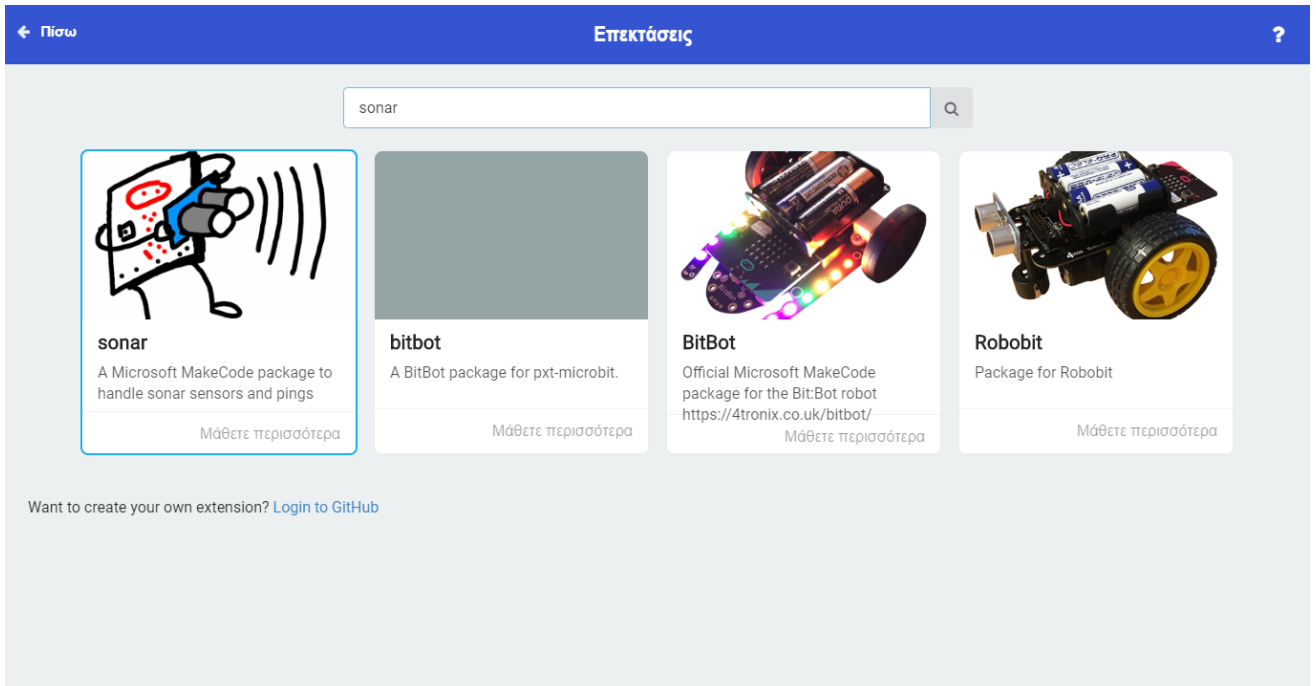
Τώρα ας δούμε πως μπορούμε να προγραμματίσουμε το micro:bit να μετράει την απόσταση με τον υπερηχητικό αισθητήρα απόστασης..

Βήματα της δραστηριότητας

1. Ξεκινήστε ένα καινούριο έργο στο MakeCode. Μπορείτε να το πείτε ultrasonicsensor.

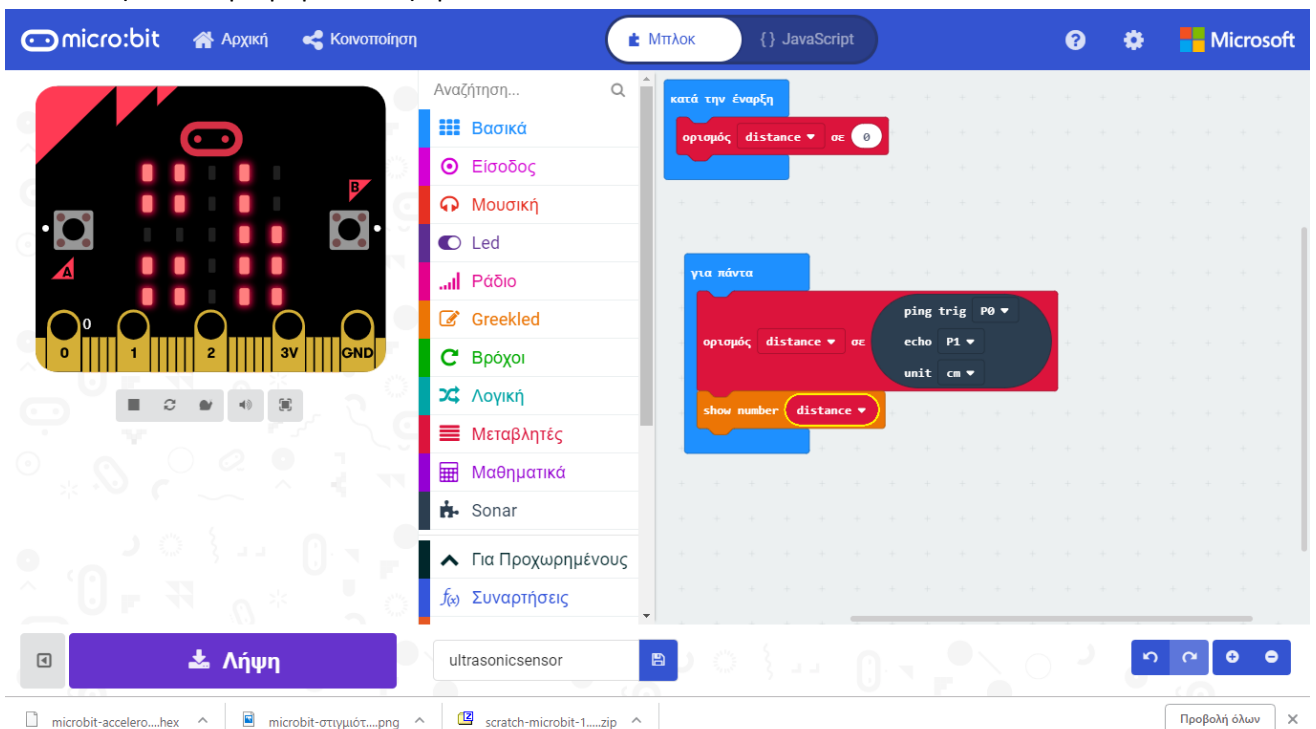


2. Ευτυχώς για μας, δεν χρειάζεται να κάνουμε κάποιο πολύπλοκο υπολογισμό. Υπάρχει μια έτοιμη επέκταση για το micro:bit που λέγεται sonar και μας επιτρέπει να χρησιμοποιήσουμε υπερηχητικούς αισθητήρες απόστασης οπότε ας το χρησιμοποιήσουμε.



Η επέκταση θα προσθέσει ένα μπλοκ Sonar στην περιοχή των μπλοκ, με ένα μοναδικό μπλοκ με την ονομασία ping. Είναι καλή ιδέα επίσης να προσθέσετε και το πακέτο greeklcd, έτσι ώστε να μπορείτε να απεικονίσετε την απόσταση που θα μετρήσει ο υπερηχητικός αισθητήρας απόστασης.

3. Θα χρειαστούμε και μια μεταβλητή που θα την ονομάσουμε «distance» που θα αποθηκεύει την απόσταση που θα μετράμε κάθε φορά.



Όπως μπορείτε να δείτε το μπλοκ ping παίρνει τρεις παραμέτρους. Τον ακροδέκτη trig, τον ακροδέκτη echo και την μονάδα που θα μας γυρίσει το αποτέλεσμα (cm (εκατοστά), in (ίντσες) ή msec (μικροδευτερόλεπτα). Έτσι με τη χρήση πολύ λίγων μπλοκ φτιάξαμε μια πολύ απλή αλλά χρήσιμη συσκευή μέτρησης με υπερήχους.

4. Δυστυχώς δεν γίνεται να δοκιμάσουμε τον υπερηχητικό αισθητήρα υπερήχων στον προσομοιωτή έτσι πρέπει να φτιάξουμε τη συνδεσμολογία και να κατεβάσουμε τον κώδικα στο micro:bit για να τον δοκιμάσουμε.

Ασκήσεις:

1. Συναρμολογήστε το κύκλωμα και δοκιμάστε τον κώδικα ανεβάζοντάς τον στο micro:bit.
2. Αντικαταστήστε την μονάδα μέτρησης (unit) στο μπλοκ ring. Τι υπολογισμούς θα πρέπει να κάνετε στη συνέχεια για να πάρετε το σωστό αποτέλεσμα σε cm;

Για αυτούς που έχουν ... όρεξη:

Προσπαθήστε να βάλετε ηχητική ανάδραση στον αισθητήρα υπερήχων. Συνδέστε ένα ηχείο ή έναν βομβητή (buzzer) στον ακροδέκτη 0 and αναδιατάξτε τους ακροδέκτες ring και echo στους ακροδέκτες 1 και 2 του micro:bit. Μετά χρησιμοποιήστε τον ακροδέκτη 0 για να παραγωγή τόνων όλο και πιο συχνά όσο η απόσταση μειώνεται (Κάπως έτσι δουλεύει και ένας αισθητήρας παρκαρίσματος)

Χρησιμοποιείτε τα μπλοκ που βρίσκονται στην κατηγορία «Μουσική». Σημειώστε ότι ο βομβητής ή το ηχείο συνδέονται μεταξύ του ακροδέκτη 0 και της γείωσης (GND).

Σύνδεσμοι	
1	https://cdn.sparkfun.com/assets/b/3/0/b/a/DGCH-RED_datasheet.pdf Φύλλο οδηγιών κατασκευαστή για τον υπερηχητικό αισθητήρα απόστασης.
2	https://en.wikipedia.org/wiki/Speed_of_sound Η ταχύτητα του ήχου μας βοηθά να υπολογίσουμε την απόσταση

Δραστηριότητα A9 – Χρησιμοποιώντας ένα κύκλωμα οδήγησης κινητήρα για να ελέγξουμε ένα κινητήρα με το micro:bit.

Περίληψη: Σε αυτό το έργο θα ελέγξουμε ένα μικρό κινητήρα συνεχούς ρεύματος (DC motor) χρησιμοποιώντας το micro:bit και ένα πρόσθετο κύκλωμα που λέγεται κύκλωμα οδήγησης κινητήρα (motor driver). Θα δούμε πως μπορούμε να σταματήσουμε και να ξεκινήσουμε το μοτέρ δίνοντας του απλές εντολές. Ο έλεγχος κινητήρων είναι κάτι βασικό που μπορούμε να προχωρήσουμε για να φτιάξουμε ένα ρομποτικό αυτοκινητάκι για παράδειγμα.

Στόχοι:

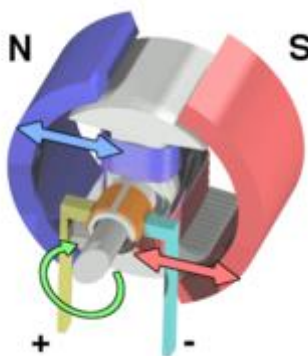
- Κατανόηση της αρχής λειτουργίας ενός κινητήρα συνεχούς ρεύματος.
- Γιατί χρειάζεται επιπλέον κύκλωμα για να λειτουργήσει το μοτέρ.
- Πως μπορούμε να ελέγξουμε τη λειτουργία του κινητήρα στέλνοντας εντολές στο κινητήρα με το micro:bit.

Απαιτούμενο υλικό

Εξαρτήματα	
1	Micro:bit board
2	Καλώδιο micro USB (για να συνδέσουμε το micro:bit στον υπολογιστή)
3	Kitronik Motor driver
4	Μπαταριοθήκη που παρέχει από 4.5 έως 6 Volts
5	Κινητήρας DC (DC motor)
6	Καλώδια σύνδεσης

Κινητήρες DC:

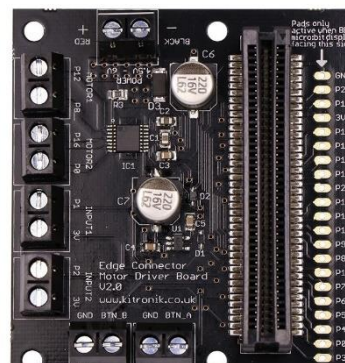
Ένας κινητήρας DC είναι μια συσκευή που χρησιμοποιεί συνεχές ρεύμα (Direct Current, DC) για να δημιουργήσει κίνηση γύρω από έναν άξονα.



How a DC motor works

Ανάγκες ρεύματος του κινητήρα και διαθέτει και ειδικά κυκλώματα που προστατεύουν το κύκλωμα από το back EMF. Στην περίπτωσή μας θα χρησιμοποιήσουμε το Kitronik Motor Driver, μια μικρή πλακέτα που έχει μια υποδοχή στην οποία μπορούμε να “κουμπώσουμε» το micro:bit. Επίσης διαθέτει 2 εξόδους και μπορεί να ελέγξει έως 2 κινητήρες ταυτόχρονα. Επειδή το κύκλωμα motor

Ένας κινητήρας DC χρησιμοποιεί μόνιμους μαγνήτες και πηνία. Όταν το ρεύμα περνάει μέσα από τα πηνία, δημιουργεί ένα μαγνητικό πεδίο που έλκει τους μαγνήτες και αυτό αναγκάζει το ρότορα (το κινούμενο μέρος του κινητήρα) να περιστραφεί γύρω από τον άξονά του. Αυτή είναι η βασική αρχή ενός ηλεκτρικού κινητήρα DC. Όμως το micro:bit δε μπορεί να βάλει σε λειτουργία απευθείας ένα κινητήρα DC γιατί δεν μπορεί να προσφέρει αρκετό ρεύμα έτσι ώστε να δημιουργηθεί το μαγνητικό πεδίο που χρειάζεται για να ξεκινήσει η περιστροφή του κινητήρα. Γι' αυτό χρειαζόμαστε ένα ειδικό κύκλωμα που λέγεται οδηγός κινητήρα (motor driver). Το κύκλωμα αυτό μπορεί να ανταπεξέλθει στις

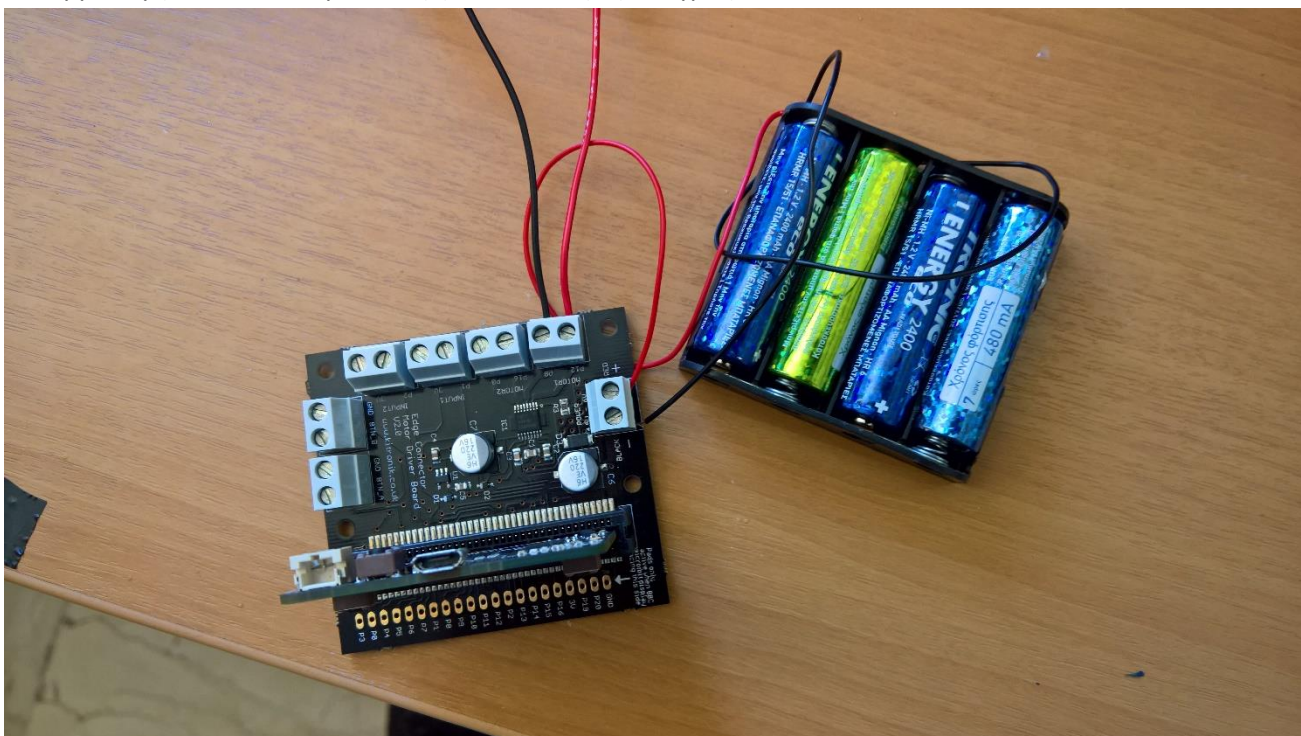


The kitronik motor driver board

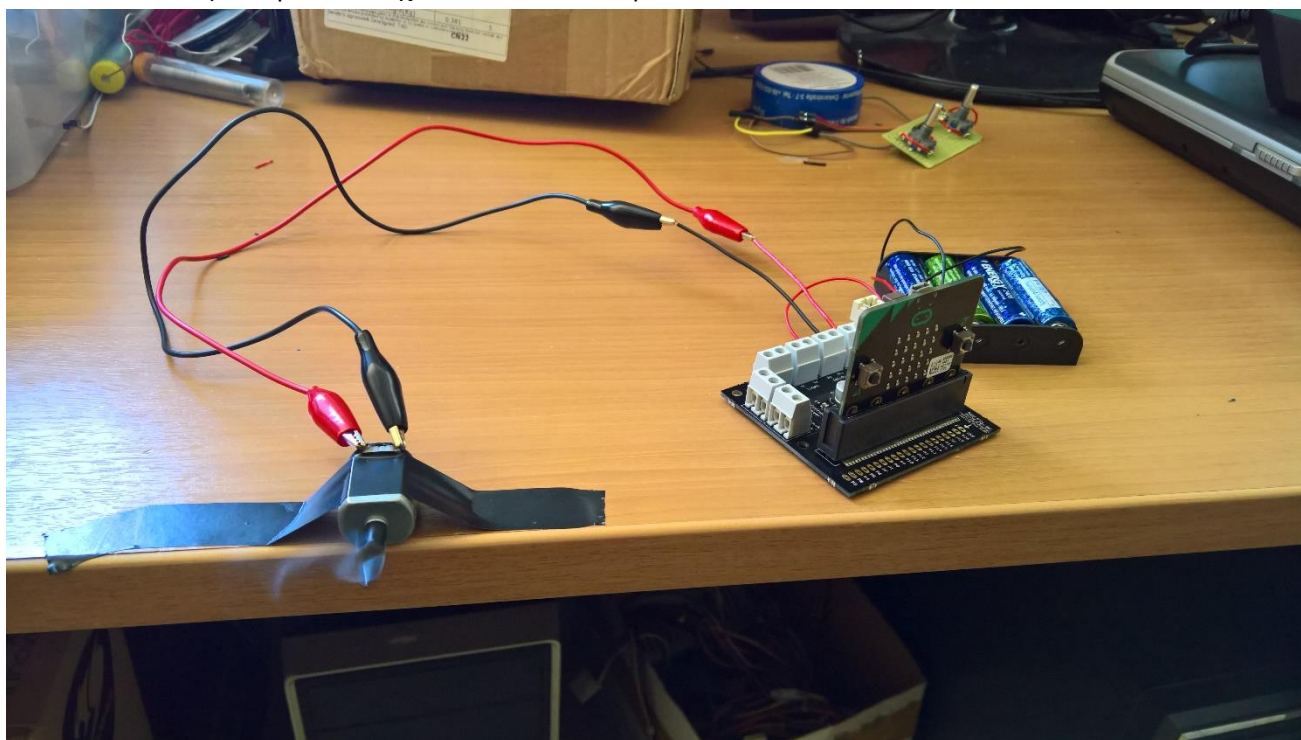
driver πρέπει να δώσει πολύ ρεύμα στο κινητήρα DC, χρειάζεται μια ξεχωριστή πηγή ρεύματος (για παράδειγμα μια μπαταριοθήκη).

Βήματα της δραστηριότητας

1. Συναρμολογήστε το κύκλωμα όπως φαίνεται στην φωτογραφία.:

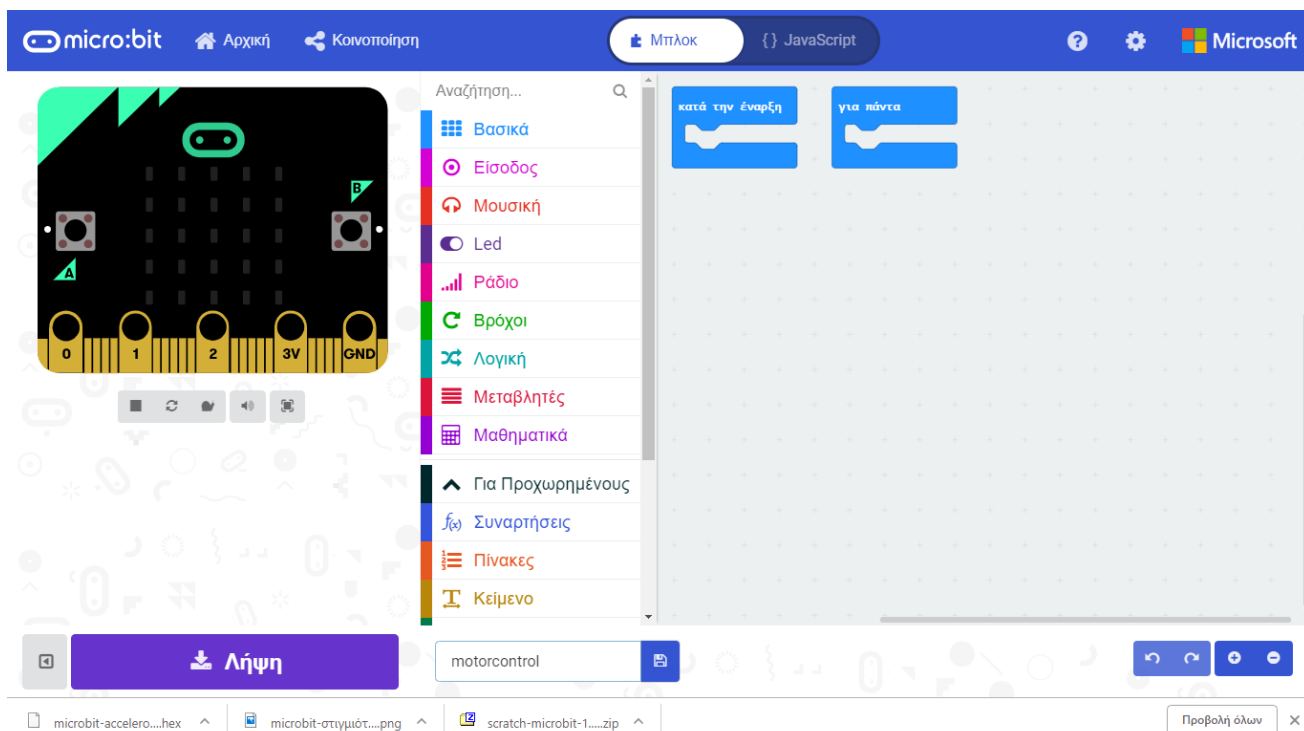


Πρέπει να συνδέσετε τη μπαταριοθήκη στο + και το - της πλακέτας kitronik motor driver board. Στο τέλος θα πρέπει να έχετε αυτό το κύκλωμα:

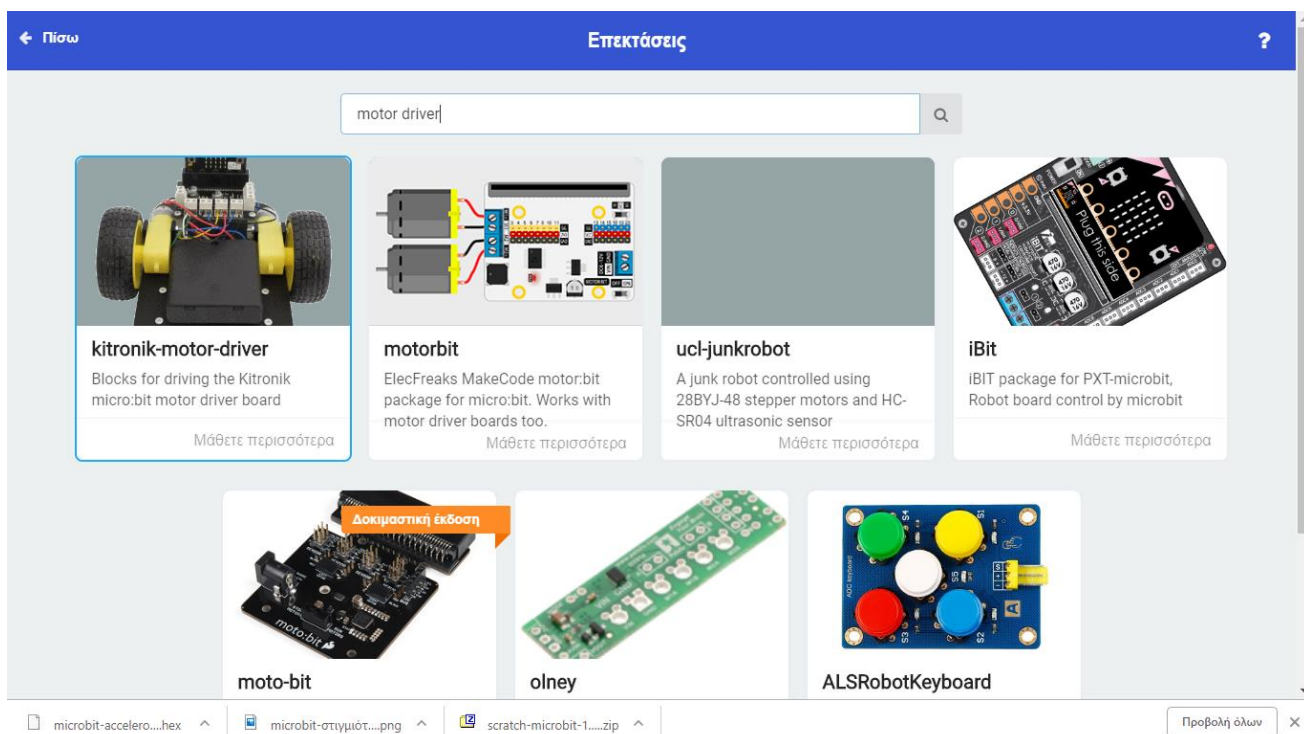


Όπως βλέπετε, το kitronik motor driver board έχει μια υποδοχή που κουμπώνουμε το micro:bit. Στηρίζουμε τον κινητήρα DC σε σταθερό σημείο με μονωτική ταινία. Ένα κομμάτι ταινίας είναι επίσης τυλιγμένο γύρω από τον άξονα του κινητήρα έτσι ώστε η κίνηση του κινητήρα να γίνεται ορατή.

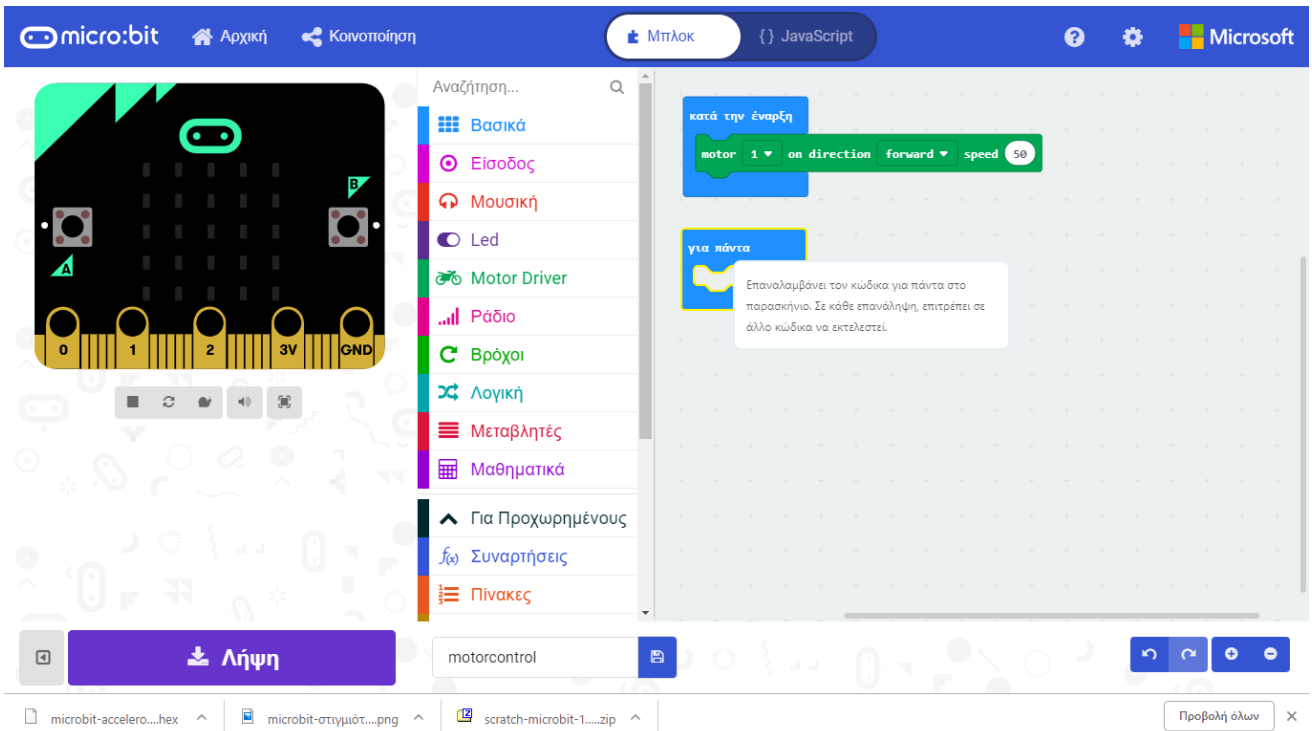
2. Τώρα είναι ώρα να φτιάξουμε το πρόγραμμά μας στο micro:bit και να το πούμε motorcontrol.



As προσθέσουμε το πακέτο kitronik motor driver χρησιμοποιώντας μια επέκταση. Ανοίξτε τις επεκτάσεις και ψάξτε για «motor driver».



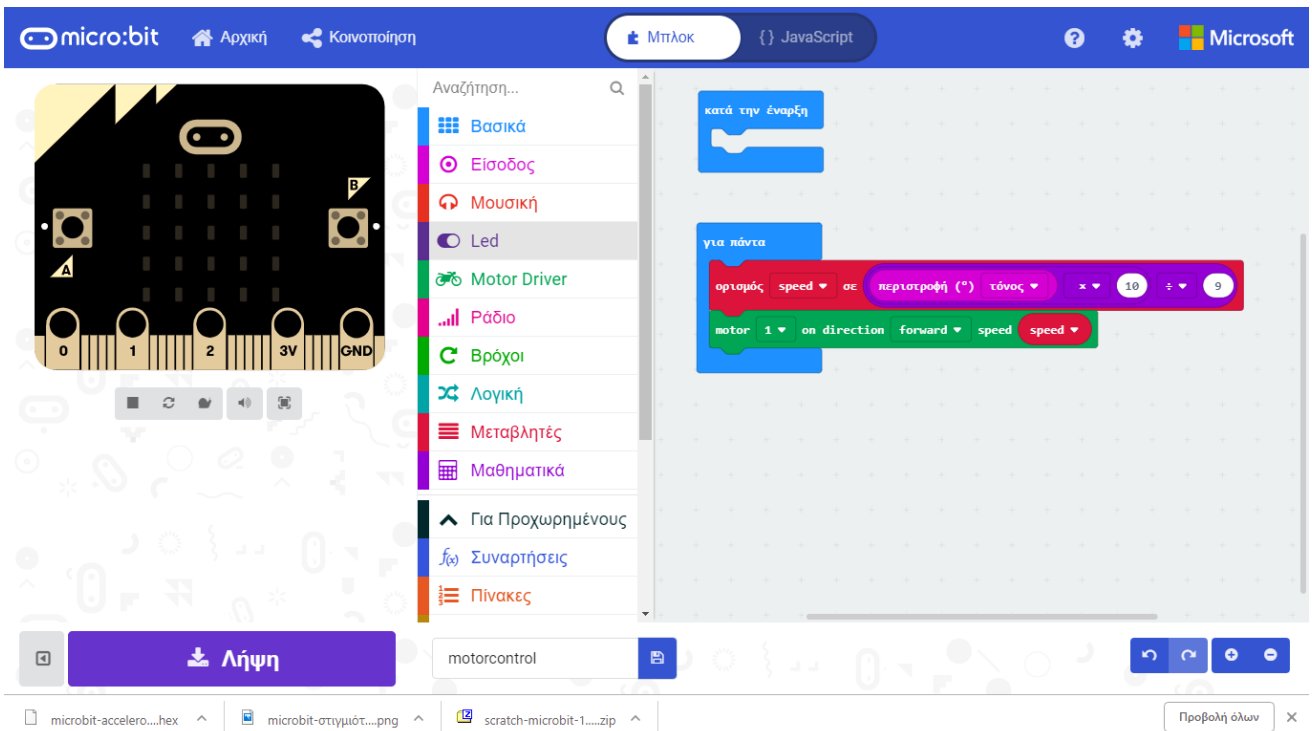
3. Επιλέξτε kitronik-motor-driver. Θα προστεθεί μια καινούρια κατηγορία μπλοκ που θα έχει 2 μπλοκ. «Motor ... on direction ... speed» και «turn off motor ...». Μπορούμε να επιλέξουμε ποιο κινητήρα θέλουμε να ελέγξουμε. Υποστηρίζονται 2 κινητήρες, ο κινητήρας 1 και ο κινητήρας 2, στους οποίους μπορούμε να καθορίσουμε την διεύθυνση (μπροστά ή πίσω) και την ταχύτητα από 0 (όχι κίνηση) ως το 100 (μέγιστη ταχύτητα).



Αν ανεβάσουμε το προηγούμενο πρόγραμμα στο micro:bit μπορούμε να δούμε το μοτέρ να γυρνάει.

Ασκήσεις:

1. Φτιάξτε το ακόλουθο πρόγραμμα και φορτώστε το στο micro:bit. Τι κάνει;



2. Απεικονίστε την ταχύτητα στην οθόνη του micro:bit.

3. Ενημερώστε το πρόγραμμα στο προηγούμενο βήμα 2 έτσι ώστε να μπορούμε να κινούμε το κινητήρα και προς τις δύο κατευθύνσεις.

Για όσους έχουν ... όρεξη:

Δημιουργήστε ένα πρόγραμμα που να επιτρέπει τον τηλεχειρισμό της ταχύτητας του μοτέρ χρησιμοποιώντας ένα δεύτερο micro:bit (το τηλεχειριστήριο). Με τα πλήκτρα «Α» και «Β» να αυξάνεται και να μειώνεται η ταχύτητα σε βήματα των 25. Έτσι αρχικά η ταχύτητα πρέπει να είναι 0, και όταν πατήσετε Β να γίνει 25. Αν πατήσετε πάλι το Α η ταχύτητα πρέπει να μειωθεί σε 0. Η μέγιστη ταχύτητα είναι 100 και η ελάχιστη 0. Το micro:bit τηλεχειριστήριο πρέπει να στέλνει μηνύματα στο άλλο χρησιμοποιώντας τη λειτουργία Ράδιο.

Σύνδεσμοι	
1	https://en.wikipedia.org/wiki/DC_motor DC motors
2	https://www.kitronik.co.uk/5620-motor-driver-board-for-the-bbc-microbit-v2.html The kitronik motor driver board

Δραστηριότητα A10 – Μετρητής βημάτων με το micro:bit.

Περίληψη: Σε αυτό το έργο θα φτιάξουμε έναν μετρητή βημάτων με το micro:bit. Ο μετρητής βημάτων μετράει τα βήματά μας και μπορεί να μας βοηθήσει να κάνουμε υπολογισμούς απόστασης, καύσης θερμίδων και να μας βοηθήσει να μην καθόμαστε πολύ ώρα ακίνητοι αλλά να μας παρακινεί να κινηθούμε για να ξεμουδιάσουμε..

Στόχοι:

- Κατανόηση του πως μπορούμε να μετρήσουμε βήματα με τη χρήση του επιταχυνσιόμετρου.
- Χρήση μαθηματικών πράξεων για υπολογισμό απόστασης και θερμίδων.
- Τρόποι μέτρησης του χρόνου με το micro:bit.

Απαιτούμενο υλικό

Εξαρτήματα	
1	Micro:bit board
2	Καλώδιο micro USB (για να συνδέσουμε το micro:bit στον υπολογιστή)
3	Μπαταριοθήκη για λειτουργία του micro:bit χωρίς να είναι συνδεδεμένο με καλώδιο USB

Μετρώντας τα βήματα με το επιταχυνσιόμετρο

Το επιταχυνσιόμετρο του micro:bit μπορεί να χρησιμοποιηθεί για να μετρήσει τα βήματα που κάνει κάποιος. Προϋπόθεση βέβαια είναι να έχει το micro:bit πάνω του! Τα διάφορα ρολόγια που βρίσκουμε στην αγορά ως fitness trackers διαθέτουν επιταχυνσιόμετρο με το οποίο είναι σε θέση να μετρήσουν τα βήματά μας. Εμείς για να μετρήσουμε τα βήματά μας θα χρησιμοποιήσουμε ένα μπλοκ που ανήκει στην κατηγορία Είσοδος. Το μπλοκ αυτό είναι το «στο κούνημα».

Βήματα της δραστηριότητας

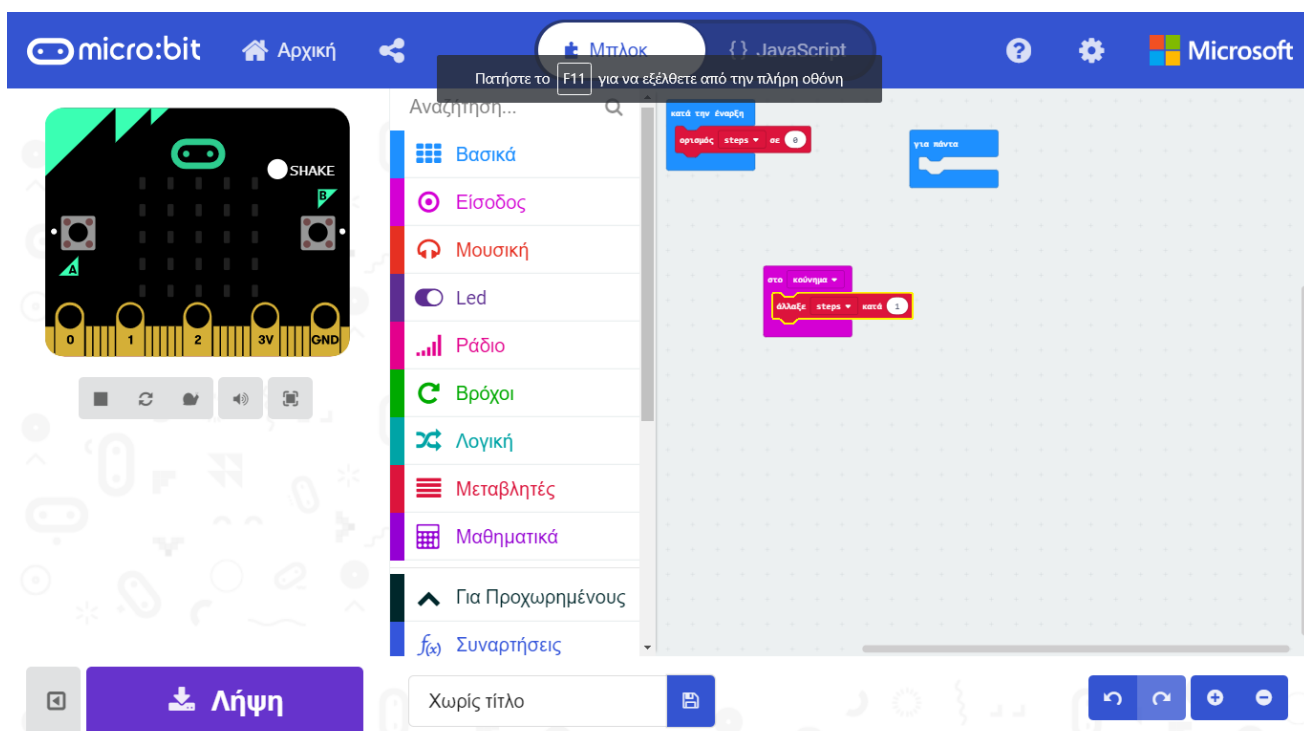
1. Δημιουργούμε μια μεταβλητή steps και την αρχικοποιούμε σε 0.

The screenshot shows the Microsoft MakeCode IDE for the micro:bit. The top bar includes the micro:bit logo, a home icon, a share icon, a block palette icon, the text 'Μπλοκ', a JavaScript icon, a help icon, a settings icon, and the Microsoft logo. The main workspace displays a script with the following code:

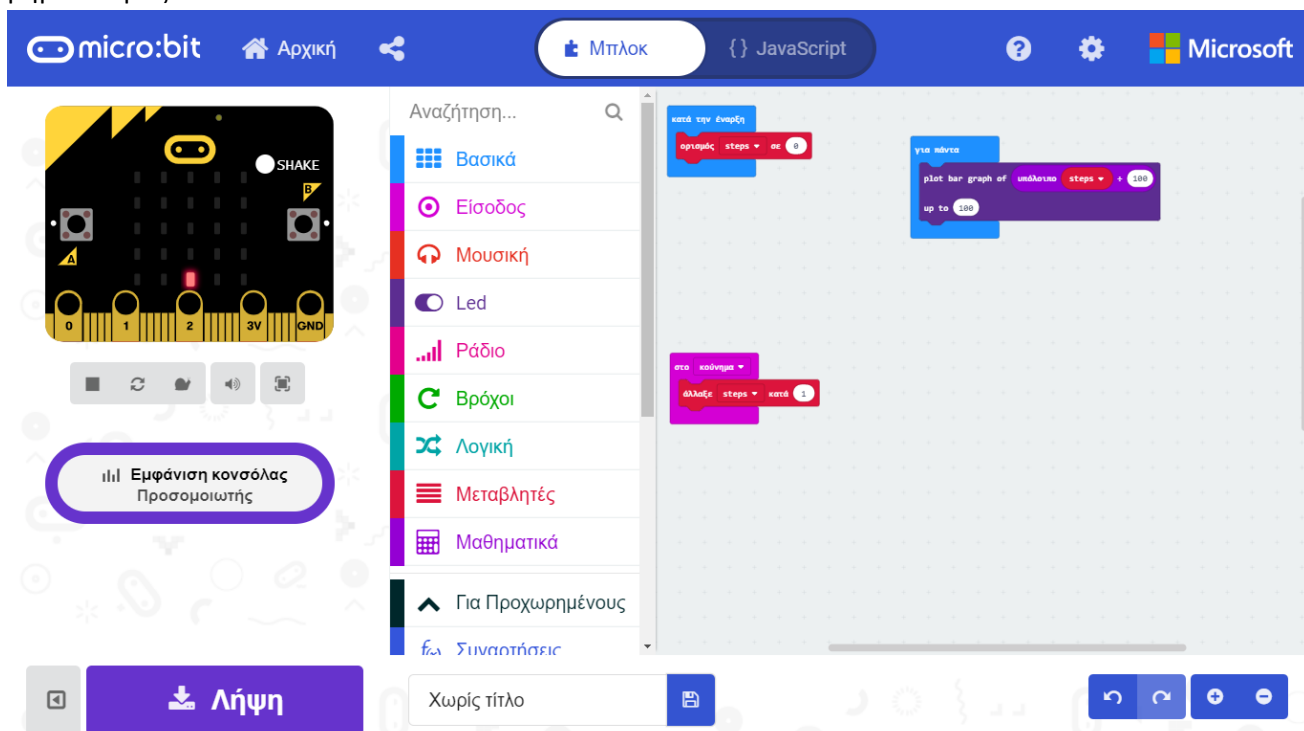
```
κατά την έναρξη  
ορισμός steps σε 0  
για πάντα
```

The 'FitnessTracker' block is visible in the bottom right corner of the workspace. The left sidebar shows a search bar and a list of categories: Βασικά, Είσοδος, Μουσική, Led, Ράδιο, Βρόχοι, Λογική, Μεταβλητές, Μαθηματικά, and Για Προχωρημένους. The 'Είσοδος' category is selected, showing the 'στο κούνημα' block.

2. Με τη βοήθεια αυτής της μεταβλητής θα μετράμε τα βήματα που κάνουμε. Στη συνέχεια μπορούμε να προσθέσουμε το «στο κούνημα» για να μετρήσουμε τα βήματα.



3. Στη συνέχεια θα μπορούσαμε να απεικονίσουμε τα βήματα χρησιμοποιώντας την “plot bar graph”. Αυτό που μπορούμε να κάνουμε για να φαίνεται ότι περπατάμε είναι να παίρνουμε το υπόλοιπο της διαίρεσης των βημάτων με το 100. Αυτό μας επιτρέπει να έχουμε μια οπτική εμφάνιση των βημάτων μας.



Ασκήσεις

1. Προσθέστε τη δυνατότητα να δείχνετε τα βήματα όταν πιέσουμε το πλήκτρο A.

2. Αν δεχτούμε ότι κατά μέσο όρο το βήμα μας έχει ένα συγκεκριμένο μήκος, π.χ. 0,5 μέτρα, δείξτε το μήκος που έχουμε διανύσει όταν πιέσουμε το πλήκτρο B.
3. Μέτρηση ταχύτητας (για προχωρημένους). Πως θα μπορούσαμε να μετρήσουμε την μέση ταχύτητά μας; Η ταχύτητα v ορίζεται ως το πηλίκο της απόστασης S που έχουμε κάνει σε χρόνο t . Δηλαδή $v = S/t$. Για να υπολογίσουμε την ταχύτητά μας σε χιλιόμετρα ανά ώρα θα πρέπει να μετατρέψουμε την απόσταση σε χιλιόμετρα και το χρόνο σε ώρες και να κάνουμε διαίρεση.