



Γενική Περιγραφή του Έργου



Αρχική Σχεδίαση του Έργου

Ένα καλά σχεδιασμένο δημόσιο πάρκο οφείλει να εξασφαλίζει την ικανοποίηση των αναγκών όλων των επισκεπτών. Θα πρέπει να προσφέρει ξεχωριστούς χώρους, όπου κάποιοι επισκέπτες μπορούν να απολαύσουν δυνατή μουσική και έντονες δραστηριότητες, ενώ άλλοι να χαλαρώνουν και να ηρεμούν. Θα σχεδιάσουμε ένα Δημόσιο Πάρκο σε ζώνες για να διασφαλίσουμε ότι επισκέπτες με διαφορετικές και αντικρουόμενες επιθυμίες θα μπορούν να συνυπάρχουν αρμονικά στο δημόσιο πάρκο.

Το Δημόσιο Πάρκο έχει διάφορες περιοχές που, ανάλογα με την διάθεσή σου, μπορείς να πας. Ένας "Συναισθηματικός Σύμβουλος" – "Θυρωρός" βρίσκεται στην είσοδο και σε βοηθά να βρεις το σημείο του πάρκου που ανταποκρίνεται στη διάθεσή σου. Από κει και πέρα αναλαμβάνει ένας "Ταξιθέτης" να σε πάει στην κατάλληλη περιοχή.

Οι μαθητές θα σχεδιάσουν το πάρκο και τις λειτουργίες του. Επίσης θα σχεδιάσουν τους "Επισκέπτες" - 3D φιγούρες που θα απεικονίζουν ένα συγκεκριμένο συναίσθημα. Μια κάμερα στην είσοδο λειτουργεί ως "Θυρωρός". Η κάμερα θα "διαβάζει" τη φιγούρα και ένα ρομπότ-



αυτοκινητάκι "Ταξιθέτης" θα αναλαμβάνει να την πάει στο κατάλληλο μέρος του πάρκου.

Η ακριβής τοποθεσία της κατάλληλης περιοχής που επέλεξε ο "Θυρωρός" θα ανιχνεύεται με RFID - NFC tag και ο "Ταξιθέτης" θα σταματά, θα αφήνει τον επισκέπτη και θα επιστρέφει στην αφετηρία του πάρκου για τον επόμενο.

Θα χρησιμοποιήσουμε micro:bit, huskylens camera, maqueen αυτοκίνητο, RFID reader. Ένα Tablet θα λαμβάνει τα δεδομένα από το micro:bit και θα προσομοιώνει την κάθε περιοχή πχ "παιδική χαρά", "συναυλιακός χώρος", "λίμνες", "γυμναστήριο". Σε μια οθόνη θα απεικονίζεται η περιοχή και θα ακούγεται ο ανάλογος ήχος. Οι προσομοιώσεις των περιοχών και οι ήχοι θα παραχθούν με εργαλεία τεχνητής νοημοσύνης, με κατάλληλες περιγραφές από τους μαθητές.







Ανάπτυξη υπολογιστικής σκέψης: (Προγραμματισμός micro:bit , MIT App Inventor)

Κριτική Σκέψη: Πως πρέπει να σχεδιάσουμε ένα πάρκο, ώστε να μην υπάρχει όχληση από αντικρουόμενες δραστηριότητες; Ποιά συναισθήματα ταιριάζουν σε ποιο χώρο; Πως θα περιγράψουμε κατάλληλα έναν χώρο για να μπορεί να τον σχεδιάσει και να τον "ντύσει" ηχητικά ένα εργαλείο Τεχνητής Νοημοσύνης;







Ήπιες δεξιότητες: Σχεδιασμός της Φιγούρας του "Επισκέπτη" (Δημιουργικότητα), Συνεργασία και Ομαδικότητα.



Βασικά Εξαρτήματα στο Έργο







 <p>micro:bit</p>	 <p>RFID Reader RFID-MFRC522</p>	 <p>Maqueen Lite Car</p>
 <p>HuskyLens Camera</p>	 <p>Android Tablet + MIT App Inventor Application</p>	 <p>NFC tags</p>

Βασικά Εργαλεία Λογισμικού (& Online)

 <p>ChatGPT</p>	 <p>Coding with the micro:bit and MakeCode</p>	 <p>MIT APP INVENTOR</p>
 <p>GIMP</p>	 <p>AUTODESK® TINKERCAD®</p>	 <p>Ultimaker cura.</p>



Media: Λογότυπο Έργου & Θέσεις Πάρκου

	 Playground.jpg 🎵 Playground.mp3	 Concert.jpg 🎵 Concert.mp3
 Gym.jpg 🎵 Gym.mp3	 Lake.jpg 🎵 Lake.mp3	 Silence.jpg 🎵 Silence.mp3

Σχήματα και Κωδικοί για Tokens (Μάρκες)

1	Playground	★
2	Concert	○
3	Lake	⬡
4	Gym	□
5	Silence	♥



Τελική Μορφή του Έργου

Πρωταγωνιστές του Έργου

Η αρχική σχεδίαση του έργου είχε λιγότερους «Πρωταγωνιστές»

- Τον «Ταξιθέτη» (maqueen car + micro:bit ελέγχου)
- Τον «Θυρωρό» (HuskyLens Camera + micro:bit ελέγχου)

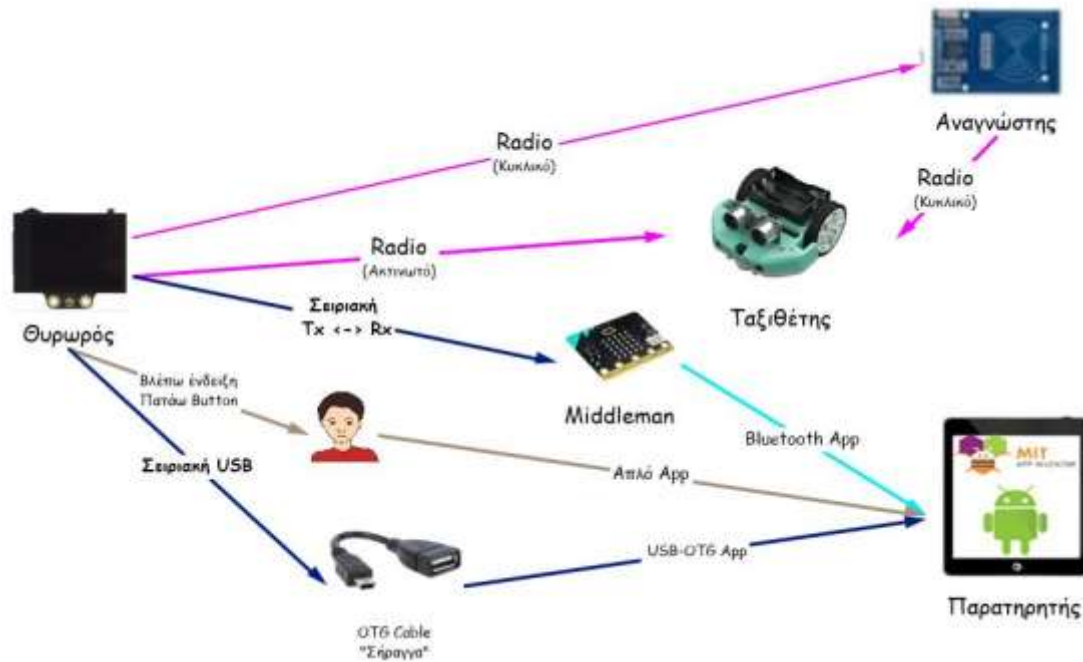
Καταλήξαμε να έχουμε 4 micro:bits και ένα Tablet για απεικόνιση των θέσεων στο Πάρκο – για «ζωντάνεμα» του Πάρκου.

Έχουμε λοιπόν:

1. Τον «**Ταξιθέτη**» (maqueen car + micro:bit ελέγχου)
2. Τον «**Αναγνώστη**» (RFID + micro:bit ελέγχου)
3. Τον «**Θυρωρό**» (HuskyLens Camera + micro:bit ελέγχου)
4. Τον «**Μεσολαβητή - Middleman**»
(micro:bit ανάμεσα HuskyLens + Tablet με **Σειριακή + Bluetooth**)
*Update: Χρησιμοποιήσαμε και άλλη διάταξη, με καλώδιο OTG
(**Σειριακή USB**) –«**Σήραγγα**».*
5. Τον «**Παρατηρητή**» (Android Tablet + MIT App Inventor)

Σχεδιάγραμμα Συστήματος του Έργου

Το τελικό σύστημα του Έργου, όπως τελικά διαμορφώθηκε φαίνεται στο παρακάτω διάγραμμα. Το έργο δεν λειτουργεί με μία μόνο διάταξη, αλλά μπορεί να υλοποιηθεί με διαφορετικούς συνδυασμούς.



«Ταξιθέτης», «Αναγνώστης» και «Θυρωρός» συνδέονται με **Radio Commands**

Οι «Ταξιθέτης» + «Αναγνώστης» συνεργάζονται μεταξύ τους. Ο «Αναγνώστης» διαβάζει το RFID tag και το επικοινωνεί στον «Ταξιθέτη» μέσω **Radio Commands**.

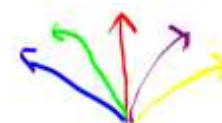
Οι «Θυρωρός» + «Middleman» συνδέονται με **Σειριακή**. Ο «Middleman» χρειάζεται γιατί τα υπόλοιπα micro:bit που χρησιμοποιούμε επικοινωνούν μεταξύ τους με **Radio Commands**, ένα πρωτόκολλο που δεν είναι συμβατό με **Bluetooth**. Οι «Middleman» + «Παρατηρητής» επικοινωνούν με **Bluetooth**.

Σημείωση: Στην πορεία του έργου διαπιστώσαμε ότι η **Σειριακή Tx-Rx** με το **Bluetooth** (το «Middleman») αργεί να ανταποκριθεί. Έτσι, υλοποιήσαμε άλλη μια εφαρμογή για τον Παρατηρητή, με χρήση **Σειριακής USB** και χρήση καλωδίου OTG «Σήραγγα».



Tokens

Τα Tokens (Μάρκες) είναι βασικό στοιχείο στο έργο. Διαβάζονται από τον Θυρωρό (Huskylens Camera) και κατευθύνουν τον Ταξιθέτη στον κατάλληλο προορισμό. Η επιλογή του Σχεδίου και του Χρώματος του μονοπατιού τους δεν είναι τυχαία, αλλά έχει να κάνει με το τι αντιπροσωπεύουν.



Προορισμός	Κωδικός	Σχέδιο	Κατεύθυνση Ακτινωτό		Χρώμα Μονοπατιού
Playground	1	★	→	Δεξιά	Κίτρινο
Concert	2	○	↗	Λοξά Δεξιά	Μοβ
Gym	3	□	↑	Ευθεία	Κόκκινο
Lake	4	⬡	↖	Λοξά Αριστερά	Πράσινο
Silence	5	♥	←	Αριστερά	Μπλε



Tokens - Σχεδίαση



AUTODESK®
TINKERCAD®

Οδηγίες για σχεδίαση Tokens στο Tinkercad

(Βλέπε και βίντεο TokenTinkercad.mp4)

1. Τοποθετώ σχήμα σε Workplane
2. Θέτω ύψος σχήματος σε 5mm
3. Αντιγραφή σχήματος (Copy)
4. Επικόλληση σχήματος (Paste)
5. Σηκώνω κατά 2mm το 2ο σχήμα
6. Μικραίνω κατά 4mm σε μήκος + πλάτος το 2ο σχήμα π.χ. 20mmx16mm
7. «Κουφώνω» (Hole) το 2ο σχήμα
8. Στοιχίζω (Align) και τα δύο σχήματα
9. Ομαδοποιώ (Group) τα σχήματα
10. Χρωματίζω καφέ το σχήμα (Προαιρετικό)

3D Tokens





Είδη Εφαρμογών με MIT App Inventor

Ο Παρατηρητής λειτουργεί με τρεις εκδοχές:





- Απλή εφαρμογή που την χειριζόμαστε μη-αυτόματα, δηλαδή βλέπουμε την ένδειξη στον Ταξιθέτη και πατάμε το αντίστοιχο Button.
- Εφαρμογή με **Bluetooth** - Αυτόματα συνδέει Θυρωρό + Παρατηρητή, με ενδιάμεσο Middleman, λόγω ασυμβατότητας **Radio** + **Bluetooth**.
- Εφαρμογή με **Σειριακή USB**. Αυτόματα συνδέει Θυρωρό και Παρατηρητή μέσω Καλωδίου OTG «Σήραγγα».

Είδος Εφαρμογής	Στοιχεία	Επιπλέον Εξοπλισμός	Μορφή Εφαρμογής
Απλή Εφαρμογή	Simple 		
Εφαρμογή με Ενδιάμεσο micro:bit, Σειριακή-Tx- Rx, Bluetooth https://community.appinventor.mit.edu/t/app-inventor-micro-bit-and-uart/44876	Middleman 	  	
Εφαρμογή με Σειριακή USB rkl099/Appinventor-SerialOTG https://github.com/rkl099/Appinventor-SerialOTG/blob/main/README.pdf	“Σήραγγα” 	Καλώδιο OTG 	



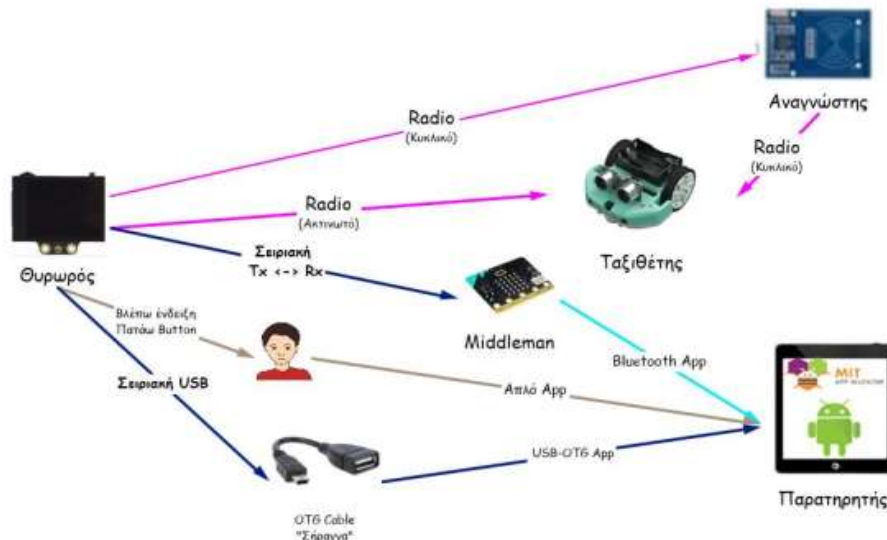
10

Σύνοψη

 <p>Παρατηρητής</p>	 <p>Απλή Εφαρμογή με Buttons που πατά ο χρήστης («χειροκίνητη»)</p>	 <p>Πιο Σύνθετη Εφαρμογή, που επικοινωνεί με micro:bit μέσω Bluetooth (Αυτόματη, Απομακρυσμένη σύνδεση)</p>	 <p>Χρησιμοποιεί Σειριακή USB Καλώδιο OTG (Αυτόματη, Φυσική σύνδεση)</p>
--	--	--	--



Πιθανοί Συνδυασμοί



1. Θυρωρός: Αναγνωρίζει Token, στέλνει Κωδικό (1...5) **Radio** στον Ταξιθέτη που «φορά» τα «Μάτια» (Πίστα Ακτινωτό με Εμπόδια). Ο Μαθητής βλέπει την ένδειξη του Κωδικού στον Ταξιθέτη (1..5) και πατάει το αντίστοιχο Button στην Απλή Εφαρμογή που έχουμε στον Παρατηρητή.
2. Θυρωρός: Αναγνωρίζει Token, στέλνει Κωδικό (1...5) **Radio** στον Αναγνώστη, στέλνει Κωδικό (1...5) **Σειριακή** στο Middleman. Ο Middleman στέλνει μέσω **Bluetooth** σε Παρατηρητή. Ο Αναγνώστης στέλνει Κωδικό (1..5) **Radio** στον Ταξιθέτη, ο οποίος κινείται στη διάταξη του Κυκλικού Πάρκου. Στον Παρατηρητή εμφανίζεται η θέση του Πάρκου που οδεύει ο Ταξιθέτης.
3. Θυρωρός: Αναγνωρίζει Token, στέλνει Κωδικό (1...5) **Radio** στον Ταξιθέτη που ΔΕΝ «φορά» τα «Μάτια» (Πίστα Ακτινωτό Με χρονισμό). Ο Θυρωρός στέλνει Κωδικό (1...5) **Σειριακή** στο Middleman. Ο Middleman στέλνει μέσω **Bluetooth** σε Παρατηρητή. Στον Παρατηρητή εμφανίζεται η θέση του Πάρκου που οδεύει ο Ταξιθέτης.

Όλοι αυτοί οι Συνδυασμοί θέλουν λίγο διαφοροποιημένο πρόγραμμα που «τρέχει» στα micro:bit, ειδικά στον Ταξιθέτη.



Τεχνολογίες που χρησιμοποιήθηκαν

NFC

NFC = Near Field Communication =Επικοινωνία Κοντινού Πεδίου, είναι μία τεχνολογία συνδεσιμότητας ηλεκτρονικών συσκευών.

- Έχει μικρή εμβέλεια
- Λειτουργεί ασύρματα στη συχνότητα των 13,56 MHz
- Ρυθμός μεταφοράς δεδομένων έως και 424 kbps
- Πολύ συνηθισμένη τελευταία:
 - ανέπαφες συναλλαγές πληρωμής
 - «ηλεκτρονικό κλειδί»
 - «ηλεκτρονικό εισιτήριο»

Πώς λειτουργεί;

Υπάρχει η Κάρτα (chip NFC), το οποίο διαβάζει ένας αισθητήρας (Αναγνώστης).

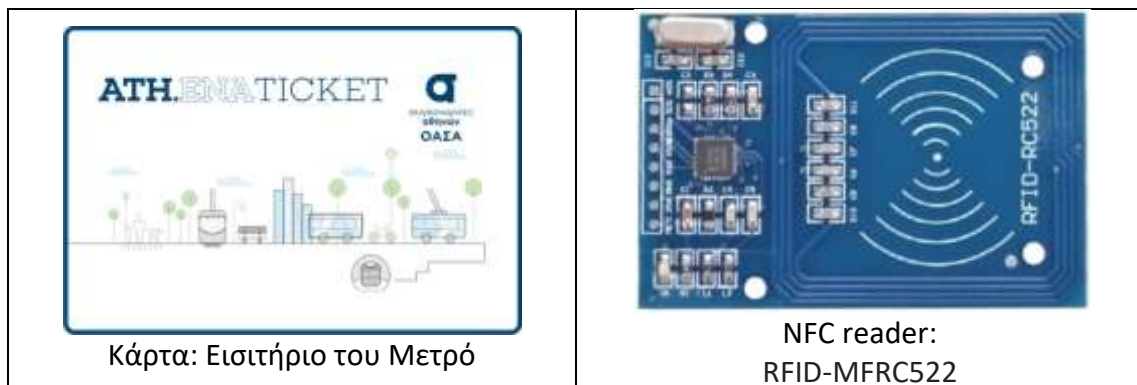


Πλησιάζουμε την κάρτα $\approx 4\text{cm}$ στον αναγνώστη και επιτυγχάνεται η επικοινωνία



Υπάρχουν διαφορετικοί τύποι καρτών. Στο έργο χρησιμοποιήθηκαν κάρτες **Mifare Classic 1,4Kbytes**. Μεταξύ άλλων οι κάρτες αυτές χρησιμοποιούνται και στα εισιτήρια των Αστικών Συγκοινωνιών της Αθήνας, οπότε χρησιμοποιήσαμε *ακυρωμένα* εισιτήρια του Μετρό.

Ο Αναγνώστης μας ήταν το RFID-MFRC522.



(Με πληροφορίες από τη Wikipedia)

AI camera

Μια AI camera (κάμερα τεχνητής νοημοσύνης) είναι μια κάμερα εξοπλισμένη με δυνατότητες τεχνητής νοημοσύνης (AI) που της επιτρέπουν να αναλύει, να αναγνωρίζει και να ερμηνεύει οπτικές πληροφορίες σε πραγματικό χρόνο.

Χρησιμοποιεί αλγόριθμους μηχανικής μάθησης (machine learning) και βαθιάς μάθησης (deep learning) για να αναγνωρίζει αντικείμενα, πρόσωπα, συναισθήματα, πινακίδες, χειρονομίες ή ακόμα και να βελτιστοποιεί τις ρυθμίσεις της ανάλογα με το περιβάλλον.



Στο Έργο χρησιμοποιήσαμε την Huskylens AI Camera και συγκεκριμένα τη λειτουργία της **Object Classification** (Ταξινόμηση Αντικειμένων).



Bluetooth



Το **Bluetooth** είναι μια ασύρματη τεχνολογία που επιτρέπει σε συσκευές να επικοινωνούν μεταξύ τους σε μικρές αποστάσεις (≈ 10 μέτρα), χωρίς τη χρήση καλωδίων. Χρησιμοποιείται για τη μεταφορά δεδομένων, όπως ήχος, αρχεία και εντολές, μεταξύ κινητών, υπολογιστών και πολλών άλλων ηλεκτρονικών συσκευών. Χρησιμοποιήσαμε το **Bluetooth** για να στέλνουμε δεδομένα στο Android Tablet.

Σειριακή Επικοινωνία

Η Σειριακή Επικοινωνία είναι ένας τρόπος ανταλλαγής δεδομένων μεταξύ συσκευών, όπου οι πληροφορίες αποστέλλονται διαδοχικά, bit προς bit, μέσω μιας γραμμής μεταφοράς δεδομένων. Αυτός ο τρόπος επικοινωνίας χρησιμοποιείται για απλή και αξιόπιστη μεταφορά δεδομένων.

Χρησιμοποιήσαμε αυτή την επικοινωνία στο micro:bit “Middleman”, επειδή θέλαμε να μεταφέρουμε πληροφορίες από την κάμερα στο Android Tablet μέσω **Bluetooth**. Τα υπόλοιπα micro:bit που χρησιμοποιούμε επικοινωνούν μεταξύ τους με **Radio Commands**, ένα πρωτόκολλο που δεν είναι συμβατό με **Bluetooth**, οπότε μεταξύ κάμερας και Tablet παρεμβάλλεται ο Μεσολαβητής – Middleman.

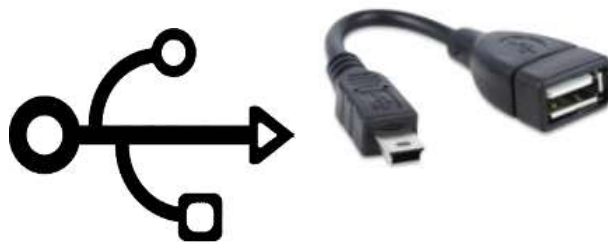
- Ταχύτητα μετάδοσης 115200 bps (bits per second)
- Tx (Transmit) είναι η γραμμή που στέλνει δεδομένα.
- Rx (Receive) είναι η γραμμή που λαμβάνει δεδομένα.

Σημαντικό: Το GND του micro:bit της κάμερας πρέπει να συνδεθεί με το GND του micro:bit του Middleman.



Σειριακή Επικοινωνία μέσω USB

Σειριακή Επικοινωνία μέσω USB με καλώδιο OTG είναι ένας τρόπος επικοινωνίας ανάμεσα σε μια συσκευή (π.χ. κινητό ή tablet) και μια άλλη συσκευή (όπως ένας μικροελεγκτής ή ένας αισθητήρας), χρησιμοποιώντας ένα **καλώδιο OTG (On-The-Go)** και το **σειριακό πρωτόκολλο** μέσω USB. Το καλώδιο OTG επιτρέπει στη φορητή συσκευή να λειτουργήσει ως **"host"** (σαν υπολογιστής), ώστε να μπορεί να ελέγχει ή να διαβάζει δεδομένα από άλλες USB συσκευές.

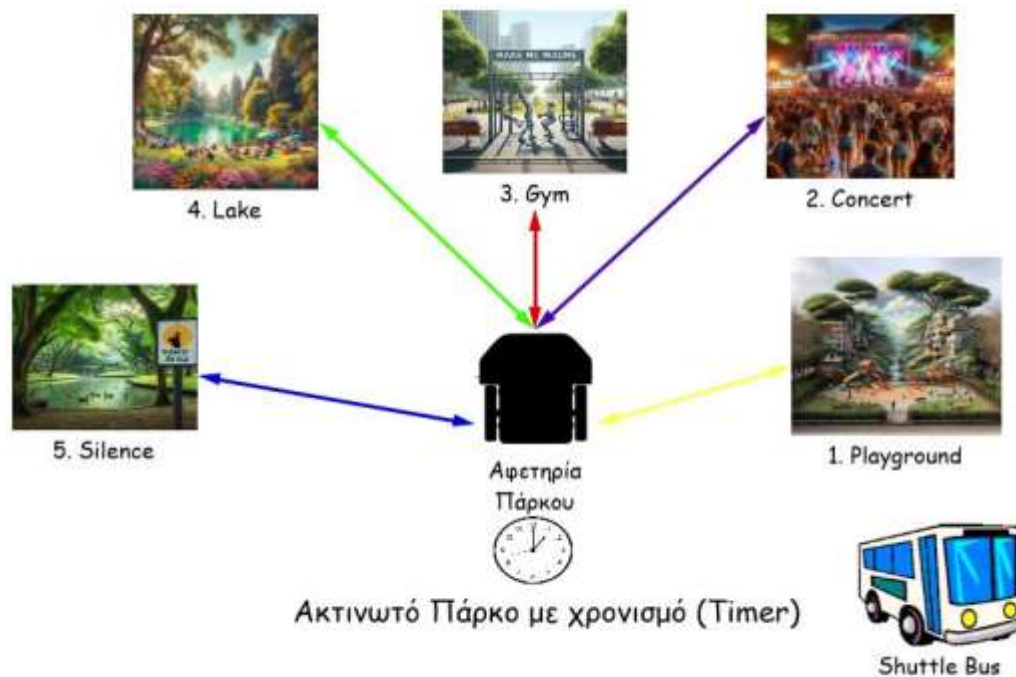




Διατάξεις Πάρκου

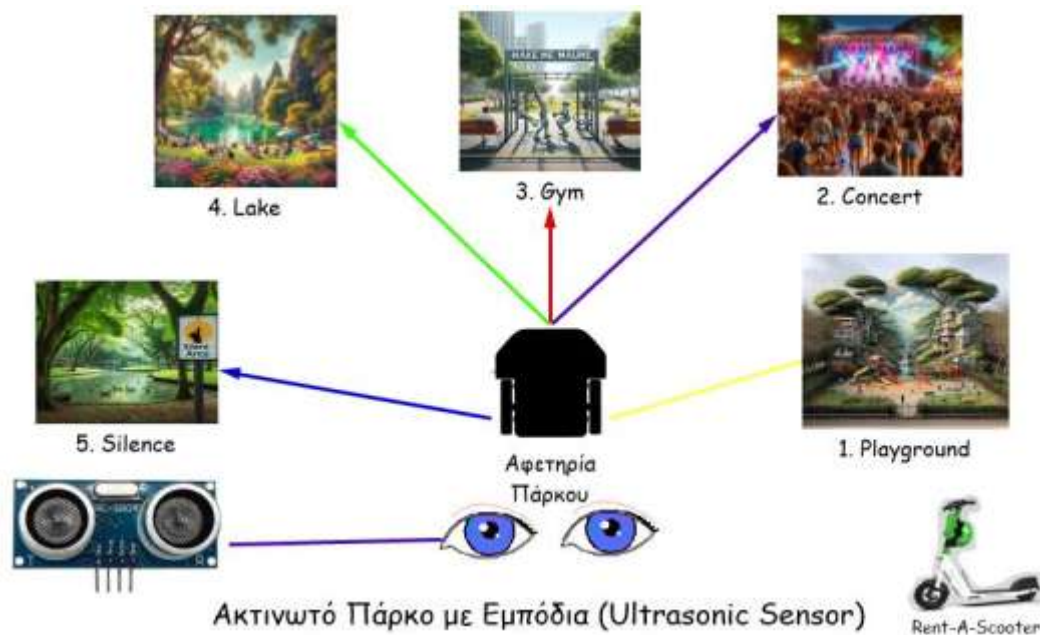
Όσο αφορά τη σχεδίαση του πάρκου, υπάρχουν τρεις διαφορετικές διατάξεις:

1. Ακτινωτό Πάρκο με Timer



Ο Ταξιθέτης δεν κινείται πάνω σε γραμμή (line follower), αλλά κινείται ελεύθερα προς συγκεκριμένη κατεύθυνση $\leftarrow \nearrow \nearrow \rightarrow$, ανάλογα με τον Προορισμό. Με χρήση timer ο Ταξιθέτης πηγαίνει *εμπρός* και γυρίζει *πίσω*. Μοιάζει με ένα Shuttle Bus, που σε πηγαίνει και σε φέρνει.

2. Ακτινωτό Πάρκο με «Μάτια» + Εμπόδια



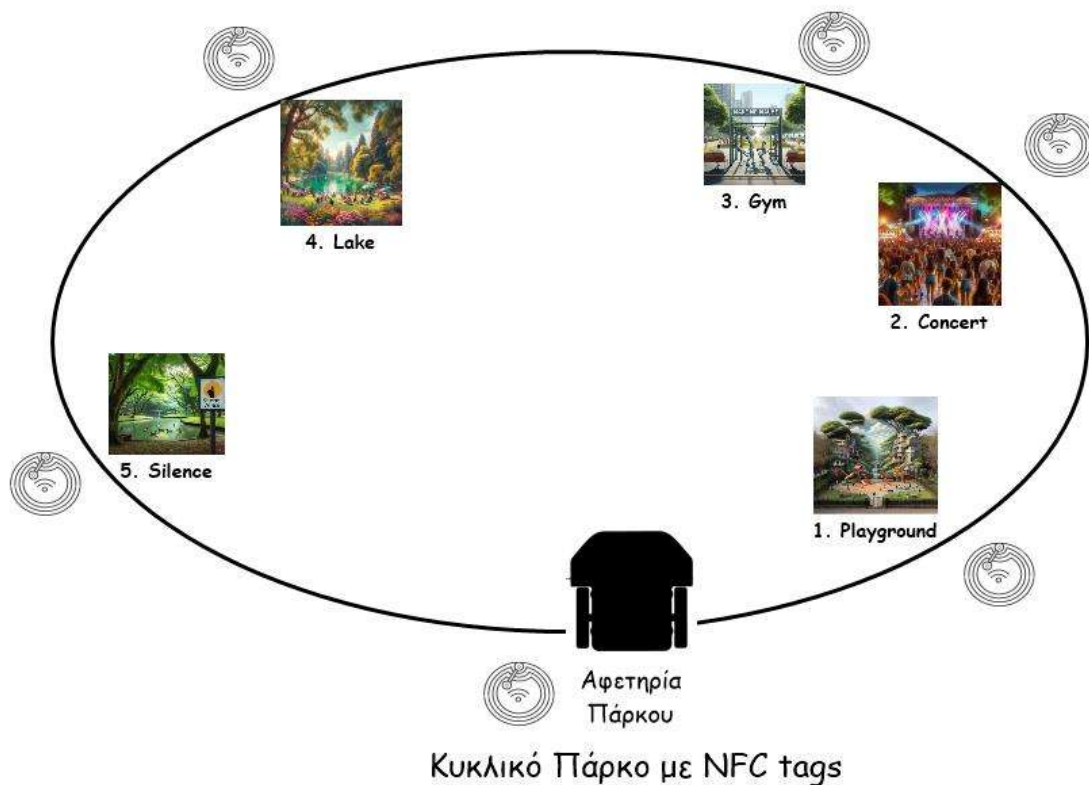
Ο Ταξιθέτης δεν κινείται πάνω σε γραμμή (line follower), αλλά κινείται ελεύθερα προς συγκεκριμένη κατεύθυνση $\leftarrow \nearrow \uparrow \searrow \rightarrow$, ανάλογα με τον Προορισμό. Χρησιμοποιείται ο Αισθητήρας Υπερήχων (Ultrasonic Sensor) – τα «Μάτια». Ο Ταξιθέτης πηγαίνει μόνο *εμπρός* και σταματά όπου βρει εμπόδιο, δηλαδή στον προορισμό. Μοιάζει με ένα ενοικιαζόμενο πατίνι, που το παίρνει κάποιος και το αφήνει. Το μαζεύει η εταιρεία, δηλαδή εμείς επαναφέρουμε χειροκίνητα τον Ταξιθέτη στην Αφετηρία.



3. Κυκλικό Πάρκο με χρήση NFC Reader

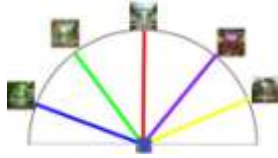

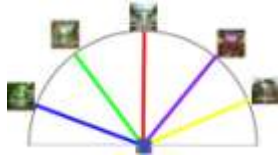





Ο Ταξιθέτης κινείται πάνω σε γραμμή (line follower). Τα «Μάτια» αφαιρούνται και στη θέση τους μπαίνει ο Αναγνώστης (RFID Reader). Ο Ταξιθέτης προχωρά πάνω στο σχεδιασμένο μονοπάτι και όταν ο Αναγνώστης αναγνωρίσει Tag που ταιριάζει με τον προορισμό, σταματά.





Είδη Πάρκου: Σύνοψη

Είδος Πάρκου	Λειτουργεί σαν	Βασίζεται σε	Κίνηση και Έλεγχος
<p>Ακτινωτό Πάρκο</p> 	<p>Shuttle Bus</p> 	<p>Χρονισμό (Timer)</p>	<p>Ελεύθερη κίνηση. Έλεγχος ταχύτητας αριστερού και δεξί τροχού.</p>
<p>Ακτινωτό Πάρκο</p> 	<p>Rent a Scooter</p> 	<p>Εμπόδια</p>	<p>Ελεύθερη κίνηση. Έλεγχος ταχύτητας αριστερού και δεξί τροχού, Ultrasonic Sensor ("Μάτια")</p>
<p>Κυκλικό Πάρκο</p> 	<p>Τρενάκι</p> 	<p>Tracking Sensors, NFC</p>	<p>Line Follower. Στάση στον προορισμό NFC reader & Tags</p>



20

Δοκιμές και Πειραματισμοί





Maqueen Lite

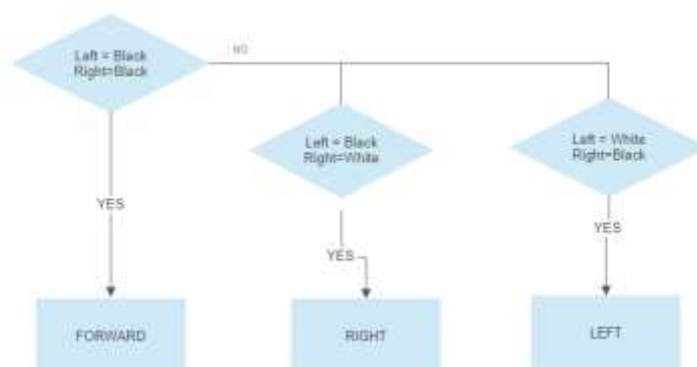
Line Tracking Maqueen

Πως προγραμματίζουμε το αυτοκινητάκι maqueen ώστε να κινείται διαρκώς πάνω σε ένα μονοπάτι



- Αν και οι δύο αισθητήρες βλέπουν Μαύρο, πηγαίνει ευθεία.
- Αλλιώς
 - Αν δεξιός αισθητήρας βλέπει λευκό **ΚΑΙ** ο αριστερός μαύρο, ενεργοποίησε δεξί κινητήρα
 - Αν αριστερός αισθητήρας βλέπει λευκό **ΚΑΙ** ο δεξιός μαύρο, ενεργοποίησε αριστερό κινητήρα

Διάγραμμα Ροής





Πρόγραμμα σε makecode

Βάζουμε επέκταση maqueen





Έλεγχος της κίνησης του Maqueen στο μονοπάτι με χρήση flag και micro:bit-ελεγκτή

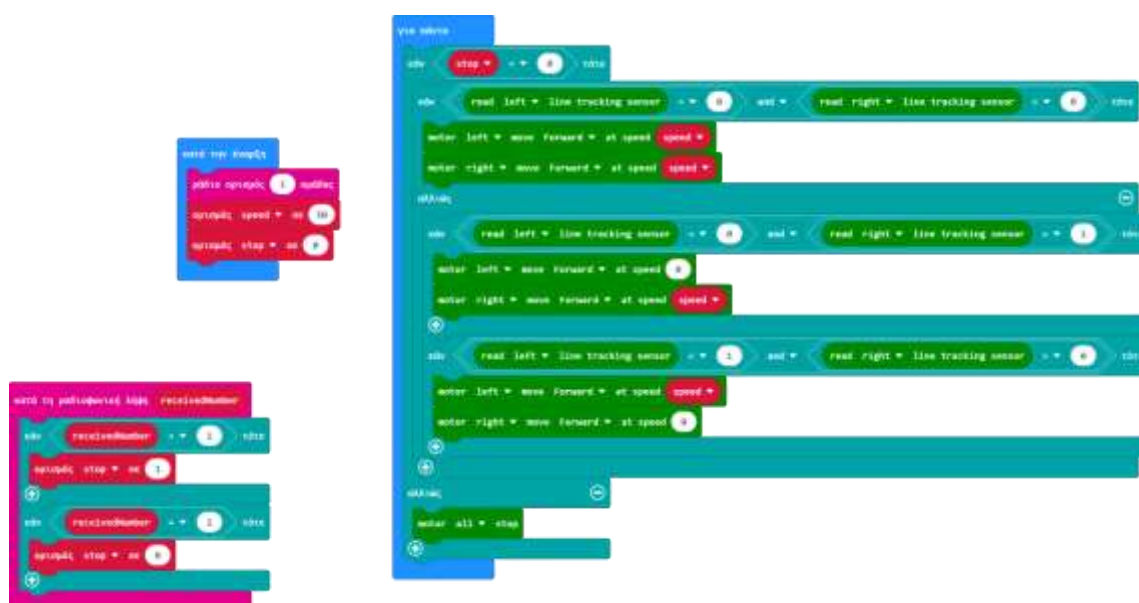
Θα χρειαστούμε και ένα δεύτερο micro:bit που θα παίζει το ρόλο του ελεγκτή. Πατώντας στον ελεγκτή το κουμπί A θα σταματάμε το Maqueen και πατώντας το κουμπί B θα ενεργοποιούμε πάλι το Maqueen. Ενεργοποιούμε και τις εντολές **Radio**.



Πρόγραμμα Ελεγκτή



Πρόγραμμα για το maqueen Η **μεταβλητή** stop παίζει το ρόλο σημαίας (flag) //Προχώρα: stop=0 // Στάματα: stop=1





Απλή Εφαρμογή Απεικόνισης Θέσεων Πάρκου (Simple App)



24

Για τα πολυμέσα (Media) της εφαρμογής χρησιμοποιήσαμε εργαλεία A.I.

- Χρησιμοποιήσαμε το <https://www.design.com/maker/logo> για να δημιουργήσουμε το εικονίδιο της εφαρμογής (ταυτόχρονα και logo του Project) -Πρώτη εικόνα αριστερά.

Για να δημιουργήσουμε εικόνες που απεικονίζουν τις θέσεις του πάρκου χρησιμοποιήσαμε το [Bing](https://bing.com/images/create)

prompt:

make me an image 1024 x 1024 pixels showing a playground in a public park

- Για να δημιουργήσουμε ήχους χρησιμοποιήσαμε το <https://elevenlabs.io/app/sound-effects/generate>

prompts:

Noise from Public Park Recreation Area with lake and picnic







Noise from Playground area

Noise from rock concert in open area

Noise from public outdoor gym outside

serene lake with ducks

quiet lake with ducks

	 Playground.jpg 🎵 Playground.mp3	 Concert.jpg 🎵 Concert.mp3
 Gym.jpg 🎵 Gym.mp3	 Lake.jpg 🎵 Lake.mp3	 Silence.jpg 🎵 Silence.mp3

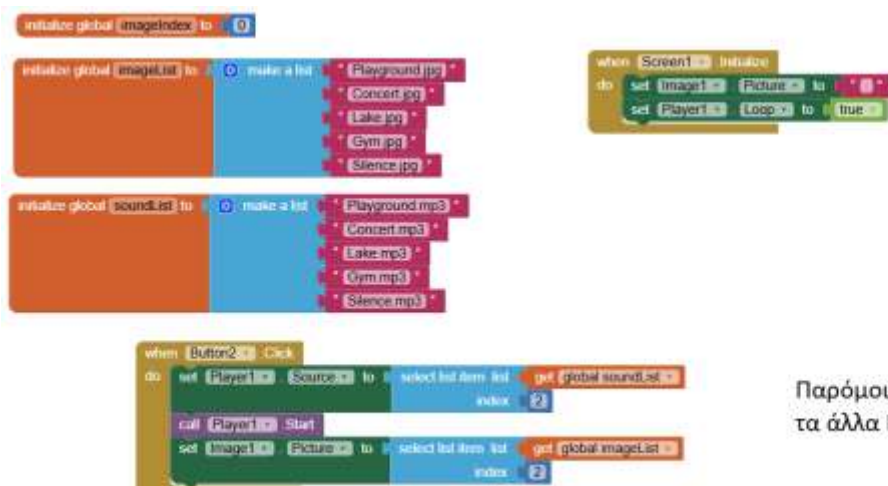


Πατώντας τα κουμπιά 1.. 5 η εφαρμογή μάς δείχνει την εικόνα στο αντίστοιχο μέρος του πάρκου και να παίζει τον ήχο που αναμένεται να ακούγεται εκεί.

Στο **Designer** της εφαρμογής έχουμε 5 Buttons, Image, Player και δύο πίνακες με τους ήχους και τις εικόνες που δημιουργήσαμε.



Blocks της εφαρμογής:



Παρόμοια και για
τα άλλα Buttons



Εφαρμογή Bluetooth Απεικόνισης Θέσεων Πάρκου (Bluetooth App)



26



Σκοπός μας είναι να κατασκευάσουμε μία εφαρμογή σε **MIT App Inventor**, έτσι ώστε να επικοινωνεί ένα **Android Tablet** με ένα **micro:bit** μέσω **Bluetooth**. Η εφαρμογή θα απεικονίζει τη θέση του πάρκου που πηγαίνει ο «Ταξιθέτης». Ταυτόχρονα, θα ακούγεται και ένας ήχος που θα προσομοιώνει αυτό που θα ακούγαμε στη θέση του πάρκου που πηγαίνουμε.

Η εφαρμογή αυτή αποδείχθηκε το πιο δύσκολο εγχείρημα στο Project αυτό.

Reference:

App Inventor, Micro:bit and UART

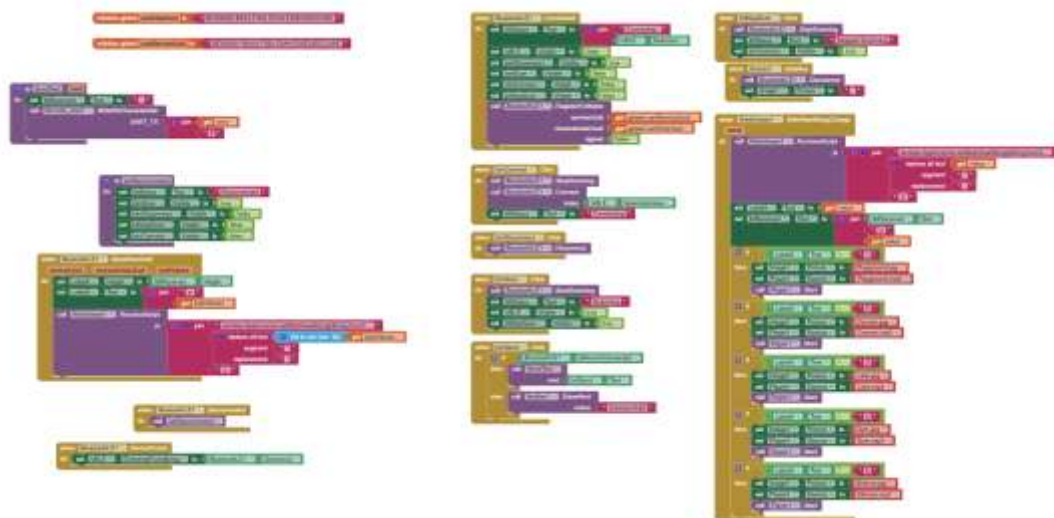
<https://community.appinventor.mit.edu/t/app-inventor-micro-bit-and-uart/44876>

Χρησιμοποιήσαμε το extension [edu.mit.appinventor.ble-20230223-beta.aix](https://community.appinventor.mit.edu/t/app-inventor-micro-bit-and-uart/44876) για το Bluetooth (BluetoothLE1).



27

Blocks της εφαρμογής στο MIT App Inventor



Designer της εφαρμογής





Εφαρμογή απεικόνισης με USB - καλώδιο OTG (USB OTG App)

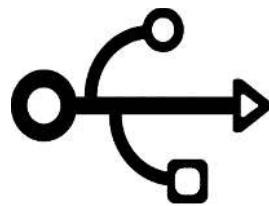


28

Η εφαρμογή απεικόνισης με το Middleman και το **Bluetooth** έχει ορισμένα μειονεκτήματα:

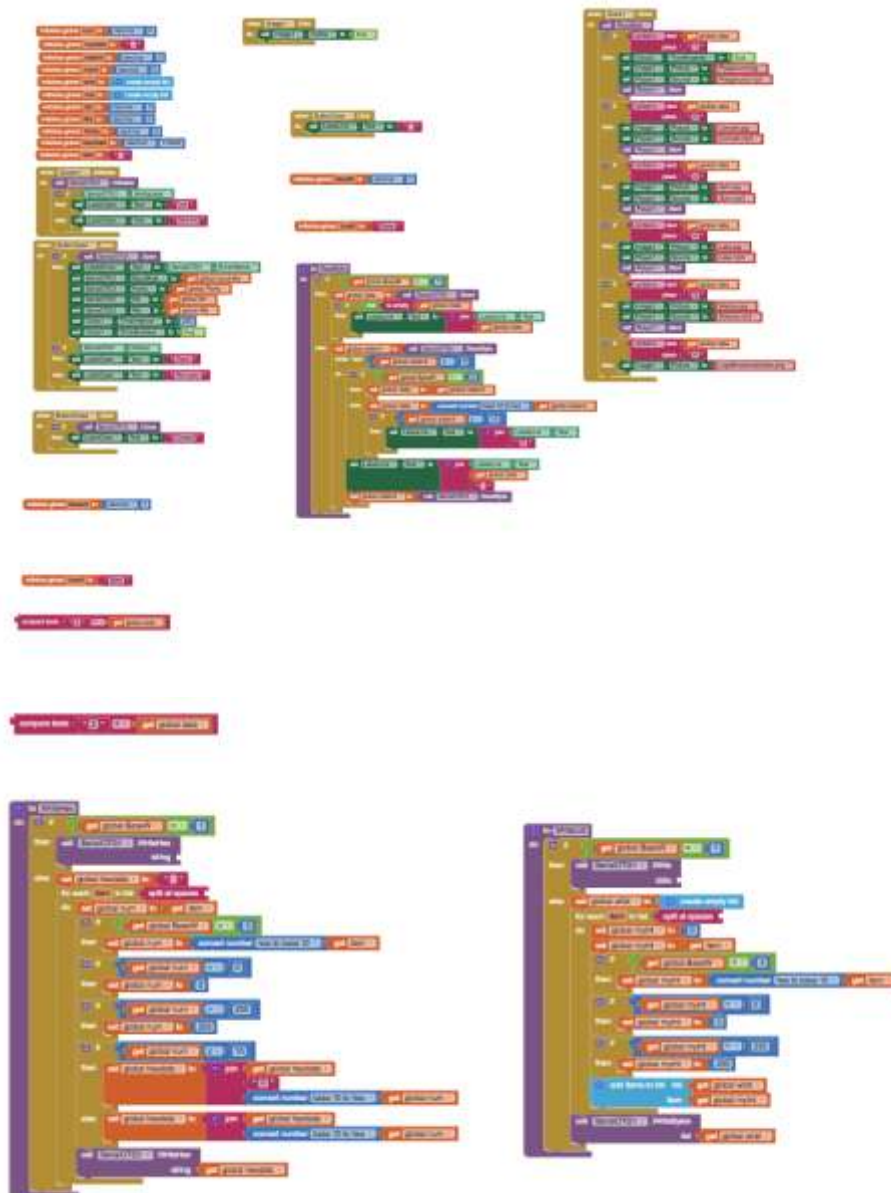
- Είναι πολύπλοκη
- Θέλει και επιπλέον micro:bit
- Έχει πολύ αργή απόκριση (το διαπιστώσαμε στην πορεία!)

Γι' αυτό λοιπόν αναζητήσαμε μια άλλη λύση και καταλήξαμε σε **Σειριακή Σύνδεση με USB**. Συνδέουμε απευθείας το Tablet με το micro:bit με καλώδιο OTG.





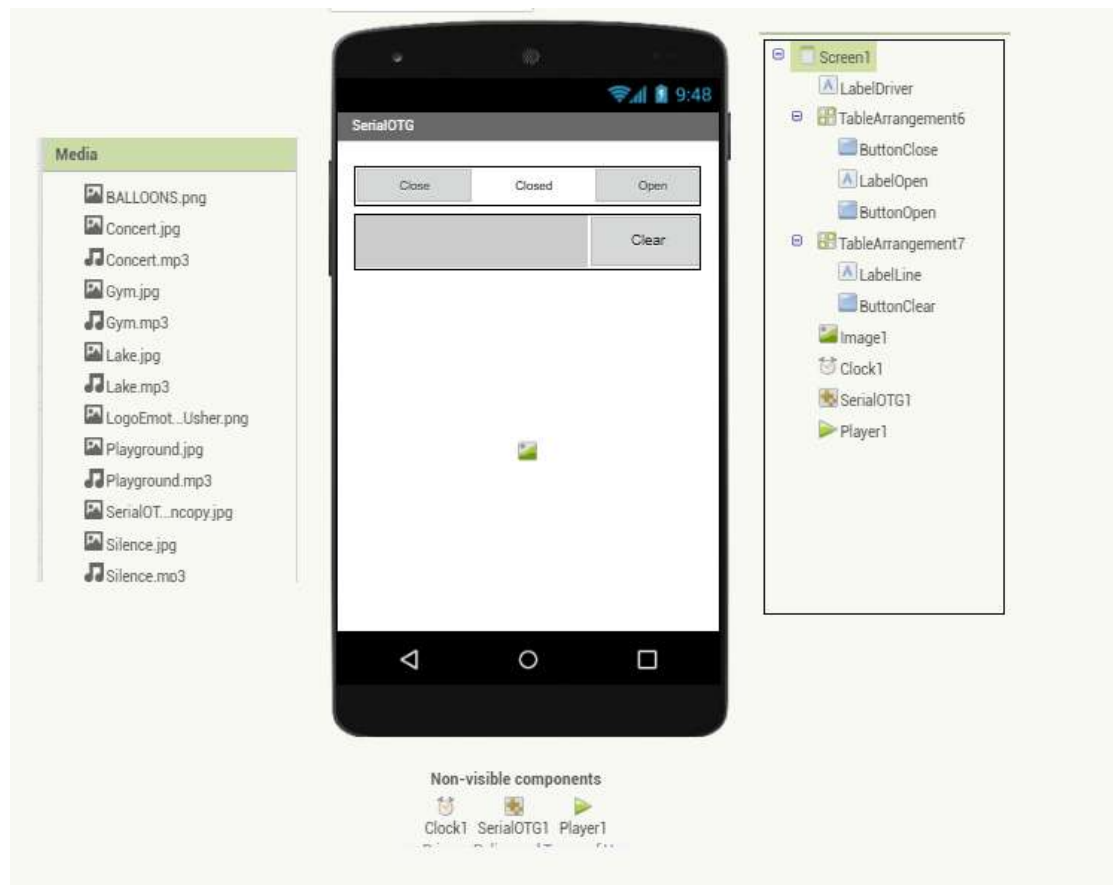
Blocks της εφαρμογής:





30

Designer της Εφαρμογής:



Reference:

[rkl099/Appinventor-SerialOTG](https://github.com/rkl099/Appinventor-SerialOTG)

<https://github.com/rkl099/Appinventor-SerialOTG/blob/main/README.pdf>



HuskyLens AI camera Εκμάθηση Tokens

Για το έργο μας χρησιμοποιήσαμε την λειτουργία της κάμερας **Object Classification**. Η κάμερα έχει:

Ροδέλα επιλογής **Function Button** στα αριστερά
(γυρίζει και μπορεί να πατηθεί)

Learning Button στα δεξιά
(μπορεί να πατηθεί)



Με το Function Button επιλέγουμε την λειτουργία **Object Classification**.

Τοποθετούμε το Token μπροστά στην κάμερα, πατάμε και κρατάμε πατημένο το Learning Button, το αφήνουμε και επαναλαμβάνουμε τη διαδικασία για όλα τα Tokens.

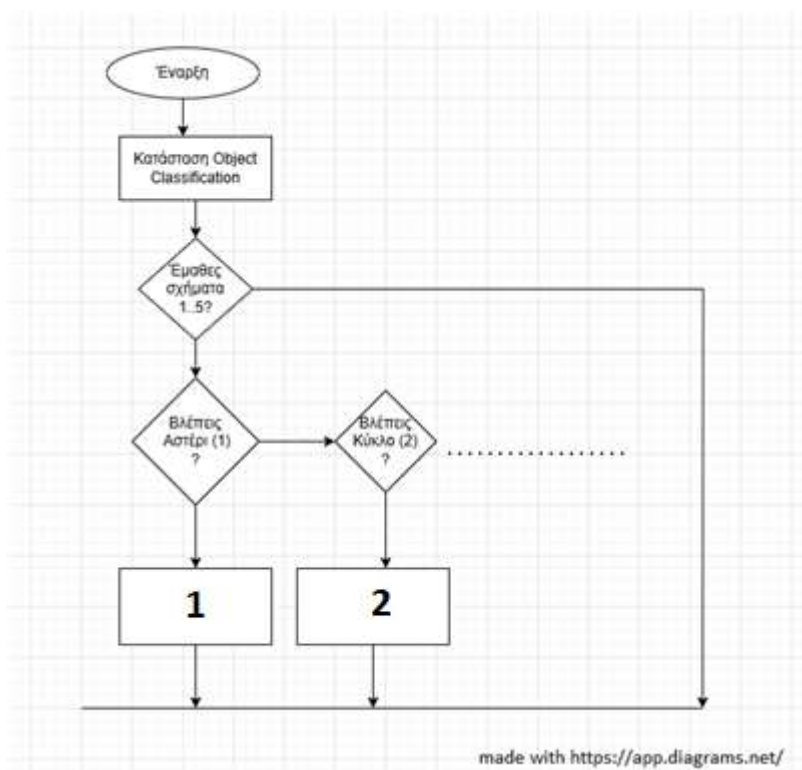


Σημαντικό: «Μαθαίνουμε» στην κάμερα και μια άδεια εικόνα, ID:6

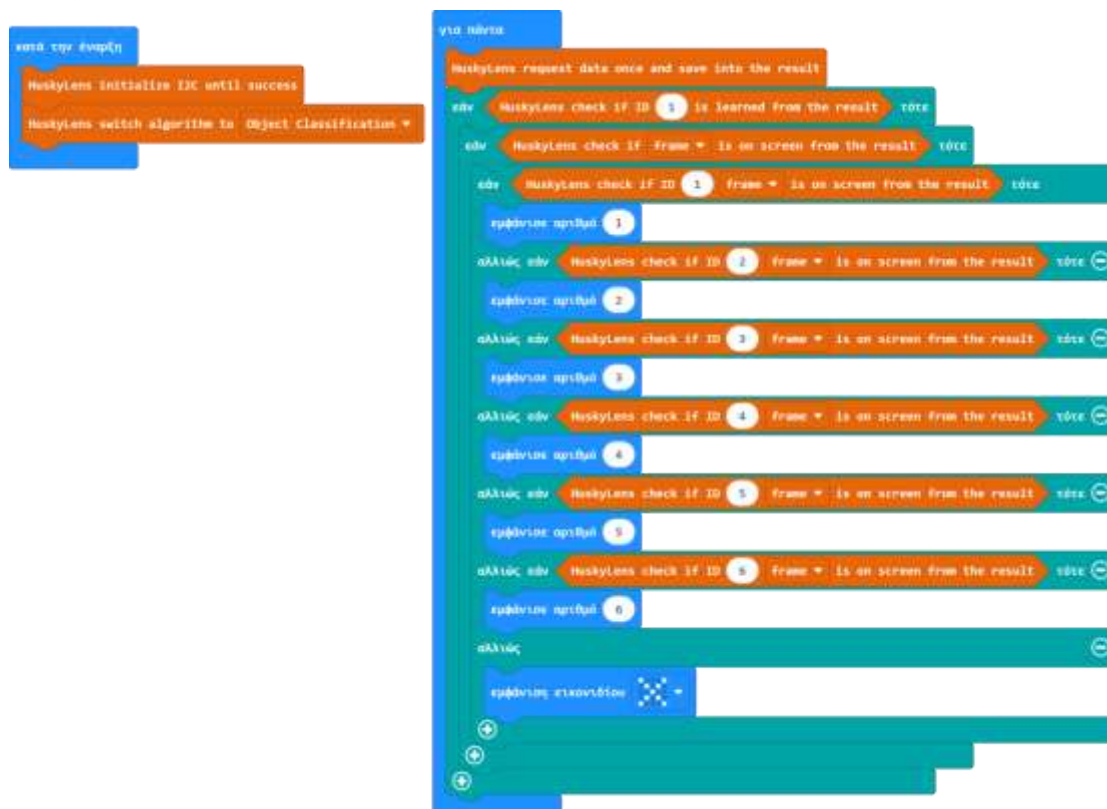
Στη συνέχεια, δοκιμάζουμε αν αναγνωρίζει τα Tokens.



Διάγραμμα Ροής:



Πρόγραμμα στο Makecode:





Huskylens Radio to micro:bit

Μπορούμε με **Radio Commands** να στείλουμε σε ένα άλλο micro:bit για να δοκιμάσουμε.

Playground	1	★
Concert	2	○
Gym	3	□
Lake	4	⬡
Silence	5	♥
Άδειο	6	-



Πρόγραμμα για Huskylens:



Πρόγραμμα για το άλλο micro:bit:





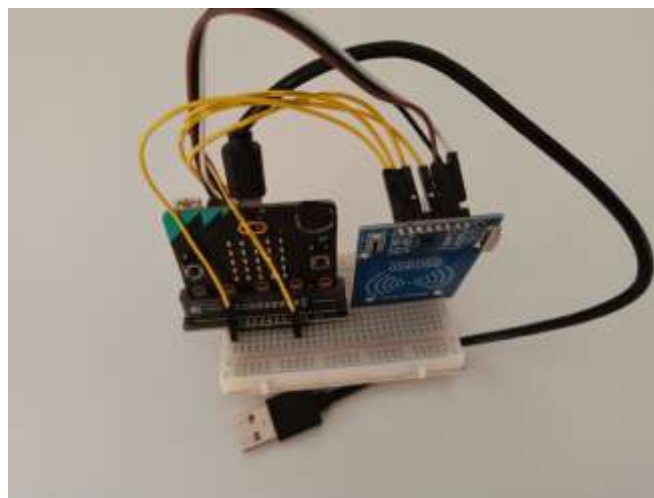
RFID-MFRC522



Συνδεσμολογία, Ανάγνωση ID Καρτών, Αντιστοίχιση σε θέσεις του πάρκου

Συνδεσμολογία

Συνδεσμολογία με το micro:bit



RC522 Pin	Micro:bit Pin	Function
SDA (NSS/SS)	P16	Chip Select (CS)
SCK	P13	SPI Clock (SCK)
MOSI	P15	SPI Data Out (MOSI)
MISO	P14	SPI Data In (MISO)
IRQ	(Not needed)	Interrupt (optional)
GND	GND	Ground
RST	P8	Reset
3.3V	3.3V	Power (⚠ Do NOT use 5V!)

Reference: <https://www.robotique.site/tutorial/read-rfid-card-data-using-microbit-board-and-rfid-rc522/>



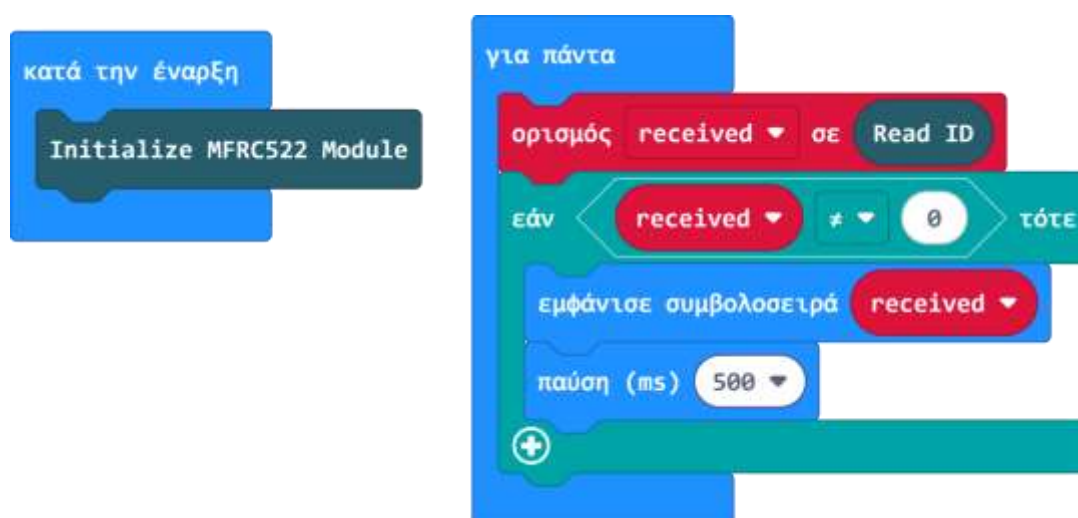
Ανάγνωση ID Καρτών

Τα tags που συνεργάζονται με το παραπάνω κύκλωμα είναι τα **ISO 14443A** RFID tags (13.56MHz MIFARE Classic 1K Smart). Μετά από πολλές δοκιμές με badge, κάρτες, τελικά καταλήξαμε να χρησιμοποιήσουμε *ακυρωμένα εισιτήρια του μετρό* για να διαβάσουμε το το RFID-MFRC522.

Έτσι ξαναχρησιμοποιούμε και δίνουμε μια δεύτερη ευκαιρία στα εισιτήρια!



Το πρόγραμμα για να μπορέσουμε να διαβάσουμε την ID κάθε κάρτας. Χρειάζεται να προσθέσουμε την **επέκταση** RFID-MRRC522:



Κάθε κάρτα έχει έναν μοναδικό δωδεκαψήφιο αριθμό ID. Καταγράφουμε εδώ.

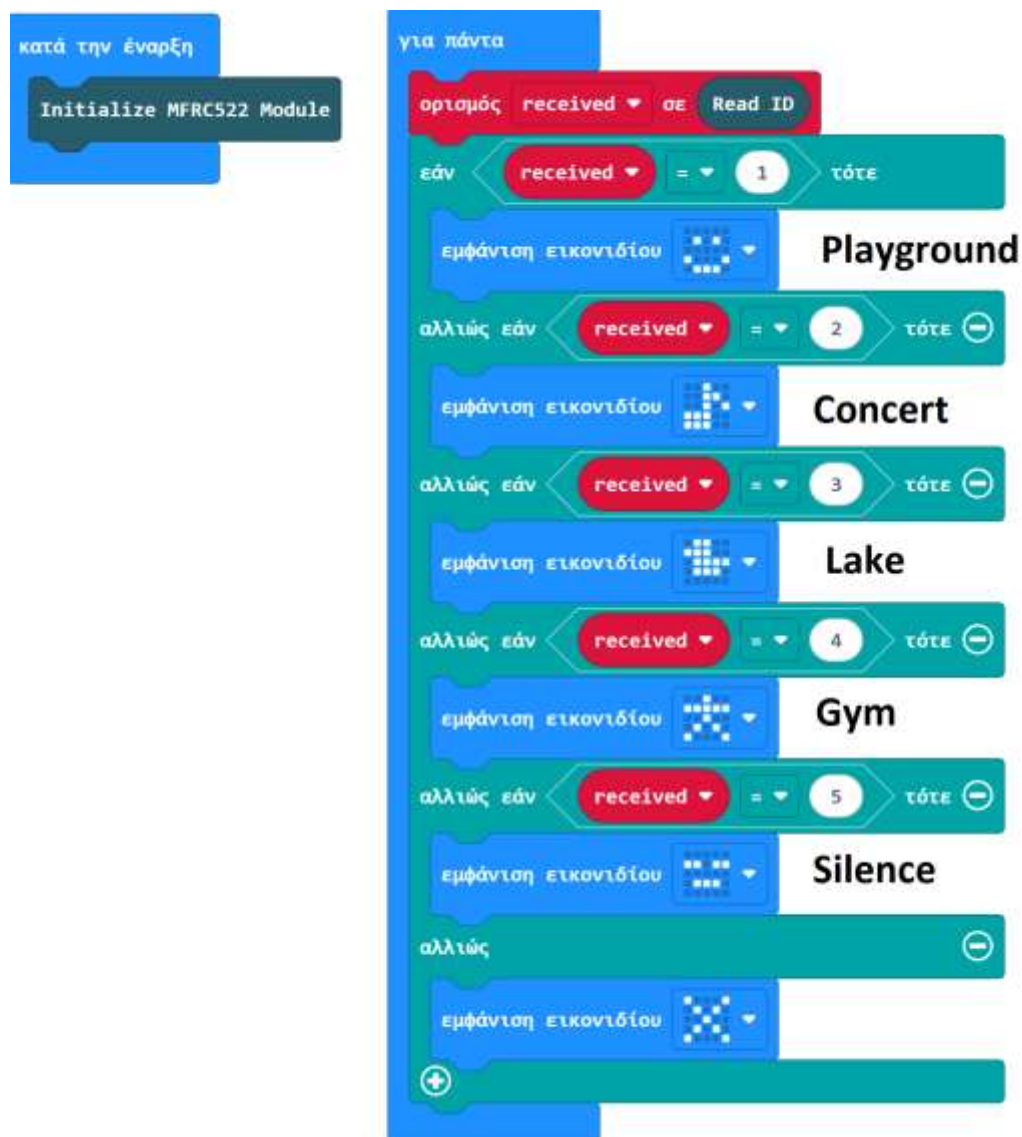
Κάρτα	ψηφίο 1	ψηφίο 2	ψηφίο 3	ψηφίο 4	ψηφίο 5	ψηφίο 6	ψηφίο 7	ψηφίο 8	ψηφίο 9	ψηφίο 10	ψηφίο 11	ψηφίο 12
#1												
#2												
#3												
#4												
#5												



Αντιστοίχιση σε θέσεις του πάρκου

Χρειάζεται να προσθέσουμε την **επέκταση** RFID-MRRC522.

Αντικαθιστούμε κάθε αριθμό 1,2...5 με τον αριθμό που καταγράψαμε από το προηγούμενο πίνακα.



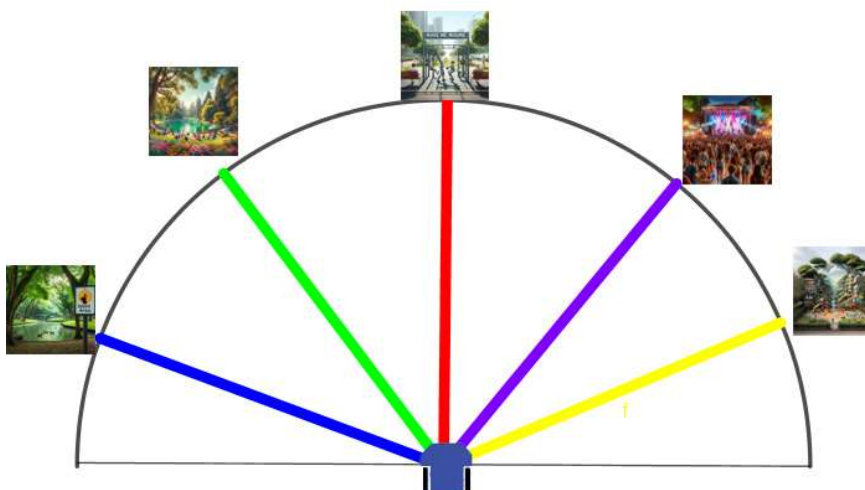
Όταν διαβάσει την κάρτα, εμφανίζει ένα εικονίδιο

(ευτυχισμένος, eighth note, πάπια, φιγούρα, κοιμάται)



Ακτινωτό Πάρκο – Δοκιμές

Ελεύθερο Μονοπάτι σε σχήμα Βεντάλιας



				
5 Silence	4 Lake	3 Gym	2 Concert	1 Playground
0,0,255	0,255,0	255,0,0	127,0,255	255,255,0

Πατώντας το Κουμπί Α του micro:bit, επιλέγουμε προς ποιά προορισμό θα πάμε. Η μεταβλητή Place περιέχει τη θέση που θέλουμε να πάμε.

Πατώντας το Κουμπί Β επιβεβαιώνουμε την κατεύθυνση και το maqueen ξεκινά να πηγαίνει στον προορισμό που του αναθέσαμε. Το χρώμα που θα φαίνεται κάτω από το maqueen θα είναι το χρώμα του κάθε προορισμού.



38

```

when green flag clicked
  create new strip object with 4 leds as RGB (000 forest)
  create new Place object
  attach new object Place
  
```

```

when button A is pressed
  when Place = 5 then
    change Place by 1
  otherwise
    create new Place object
    attach new object Place
  
```

```

when button B is pressed
  when Place = 1 then
    set speedLeft to 50
    set speedRight to 0
    strip show color red 255 green 0 blue 0
  when Place = 2 then
    set speedLeft to 50
    set speedRight to 20
    strip show color red 0 green 255 blue 0
  when Place = 3 then
    set speedLeft to 50
    set speedRight to 50
    strip show color red 0 green 0 blue 255
  when Place = 4 then
    set speedLeft to 20
    set speedRight to 50
    strip show color red 128 green 128 blue 0
  when Place = 5 then
    set speedLeft to 0
    set speedRight to 50
    strip show color red 0 green 128 blue 128
  +
  motor left move Forward at speed speedLeft
  motor right move Forward at speed speedRight
  wait (ms) 5000
  motor all stop
  motor left move Backward at speed speedLeft
  motor right move Backward at speed speedRight
  wait (ms) 5000
  motor all stop
  
```



Το maqueen έχοντας πάρει εντολή για τον προορισμό του , ανάβει με το αντίστοιχο χρώμα, πηγαίνει μπροστινή κίνηση (Forward), σταματά, στη συνέχεια επιστρέφει στην αφετηρία με αντίστροφη κίνηση (Backward).

Tips:

Για να πάει Δεξιά, δίνουμε μεγάλη ταχύτητα στον αριστερό τροχό και πολύ μικρή στον Δεξί (και αντίστροφα).

Για να πάει Λοξά Δεξιά, , δίνουμε *μεγάλη ταχύτητα* στον αριστερό τροχό και *μικρότερη* στον Δεξί (και αντίστροφα).

Θα μπορούσαμε επίσης, στο κατά την έναρξη να ορίσουμε μεταβλητές *speedLow*, *speedHigh*, ώστε να έχουμε μεγαλύτερη ευελιξία στο πρόγραμμα.

Για να πάει ευθεία, δίνουμε την ίδια ταχύτητα και στους δύο τροχούς.

Όσο μεγαλύτερη η ταχύτητα, τόσο πιο μακριά θα πάει το maqueen.

Στο Makecode **πρέπει να εισάγουμε τις επεκτάσεις *Neopixel* και *maqueen***. Ο maqueen «βλέπει» τα led στο pin15, όπως φαίνεται και στο πρόγραμμα.

Παρατηρείστε: όταν το maqueen πηγαίνει Forward και μετά Backward με την ίδια ταχύτητα και τον ίδιο χρόνο πηγαίνει στην ίδια αφετηρία; Ακολουθεί την ίδια διαδρομή; (για να το δούμε μπορούμε να ιχνηλατήσουμε με μαρκαδόρο τη διαδρομή που κάνει).

Παραλλαγή προγράμματος: Αντί να έχουμε παύσεις για να σταματάμε το maqueen, μπορούμε να χρησιμοποιήσουμε το HC-SR04 Ultrasonic Distance Module (Αισθητήρας Απόστασης με Υπερήχους), το οποίο «βλέπει» εμπόδια και σταματά. Θα πρέπει πρώτα όμως να «ιχνηλατήσουμε τη διαδρομή που κάνει το maqueen για κάθε προορισμό, να το σηματοδοτήσουμε με το αντίστοιχο χρώμα και να τοποθετήσουμε το εμπόδιο σε κάποιο σημείο της ιχνηλατημένης διαδρομής.

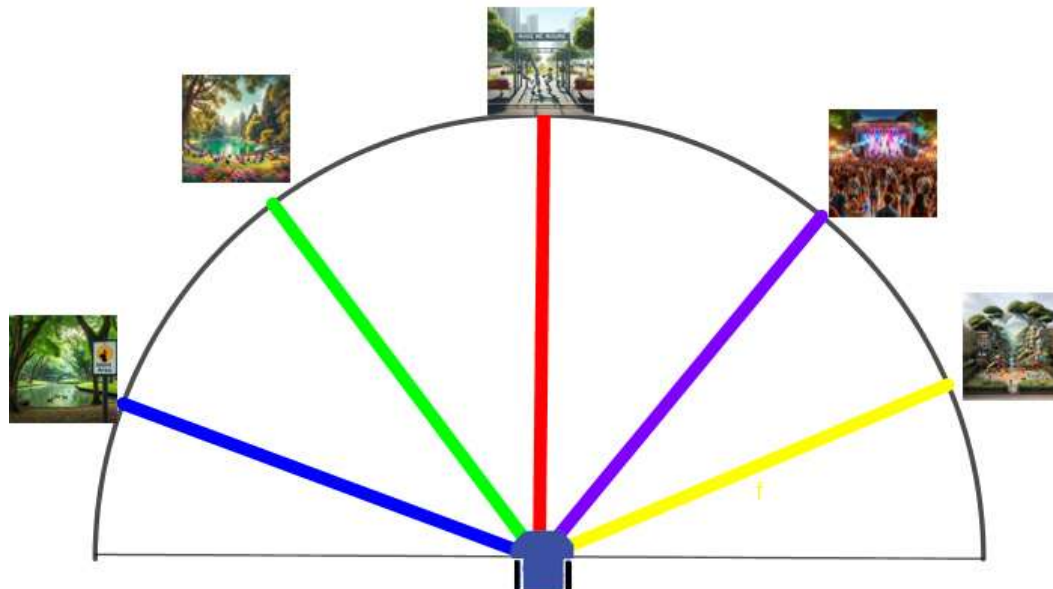


Ακτινωτό Πάρκο – Τελική Μορφή



Το σχήμα του πάρκου μοιάζει με βεντάλια η φοίνικα.

Λειτουργία Shuttle Bus

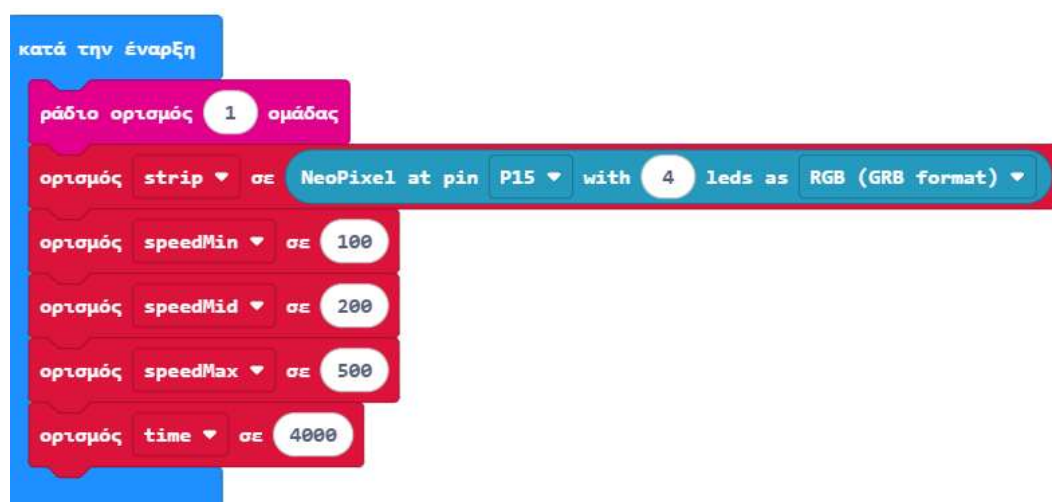


				
5 Silence	4 Lake	3 Gym	2 Concert	1 Playground
0,0,255	0,255,0	255,0,0	127,0,255	255,255,0
Min, Max	Mid, Max	Max, Max	Max, Mid	Max, Min



Θέλουμε **επεκτάσεις** Neopixel & Maqueen

Μεταβλητές	
speedMin	Ελάχιστη ταχύτητα
speedMid	Μεσαία ταχύτητα
speedMax	Μέγιστη ταχύτητα
time	Χρόνος διαδρομής & αναμονής
speedLeft	Ταχύτητα Αριστερού Τροχού
speedRight	Ταχύτητα Δεξιού Τροχού
R	Red (0 – 255)
G	Green (0 – 255)
B	Blue (0 - 255)



Στο επόμενο τμήμα του προγράμματος:

- Όταν το maqueen λάβει αριθμό (1..5) από το Huskylens, ρυθμίζει την ταχύτητα των τροχών και το χρώμα του ανάλογα με αυτό που έλαβε.
- Στη συνέχεια: πηγαίνει εμπρός, σταματά και περιμένει , πηγαίνει πίσω με την κατεύθυνση που ήρθε.
- Σταματά στην Αφετηρία.



```

when green flag clicked clickedNumber
  sprite1s Place = on clickedNumber
  when green flag clicked Place =
  when Place = 1 when
    sprite1s speedLeft = on speedRas
    sprite1s speedRight = on speedRas
    sprite1s R = on 255
    sprite1s G = on 255
    sprite1s B = on 0
    strip = show color red 5 green 5 blue 5
  when Place = 2 when
    sprite1s speedLeft = on speedRas
    sprite1s speedRight = on speedRas
    sprite1s R = on 127
    sprite1s G = on 0
    sprite1s B = on 255
    strip = show color red 5 green 5 blue 5
  when Place = 3 when
    sprite1s speedLeft = on speedRas
    sprite1s speedRight = on speedRas
    sprite1s R = on 255
    sprite1s G = on 0
    sprite1s B = on 0
    strip = show color red 5 green 5 blue 5
  when Place = 4 when
    sprite1s speedLeft = on speedRas
    sprite1s speedRight = on speedRas
    sprite1s R = on 0
    sprite1s G = on 255
    sprite1s B = on 0
    strip = show color red 5 green 5 blue 5
  when Place = 5 when
    sprite1s speedLeft = on speedRas
    sprite1s speedRight = on speedRas
    sprite1s R = on 0
    sprite1s G = on 0
    sprite1s B = on 255
    strip = show color red 5 green 5 blue 5
  when green flag clicked
    motor left = move Forward = at speed speedLeft
    motor right = move Forward = at speed speedRight
    when time = 5
    motor all = stop
    when time = 5
    motor left = move Backward = at speed speedLeft
    motor right = move Backward = at speed speedRight
    when time = 5
    motor all = stop
  
```



Λειτουργία Rent a Scooter

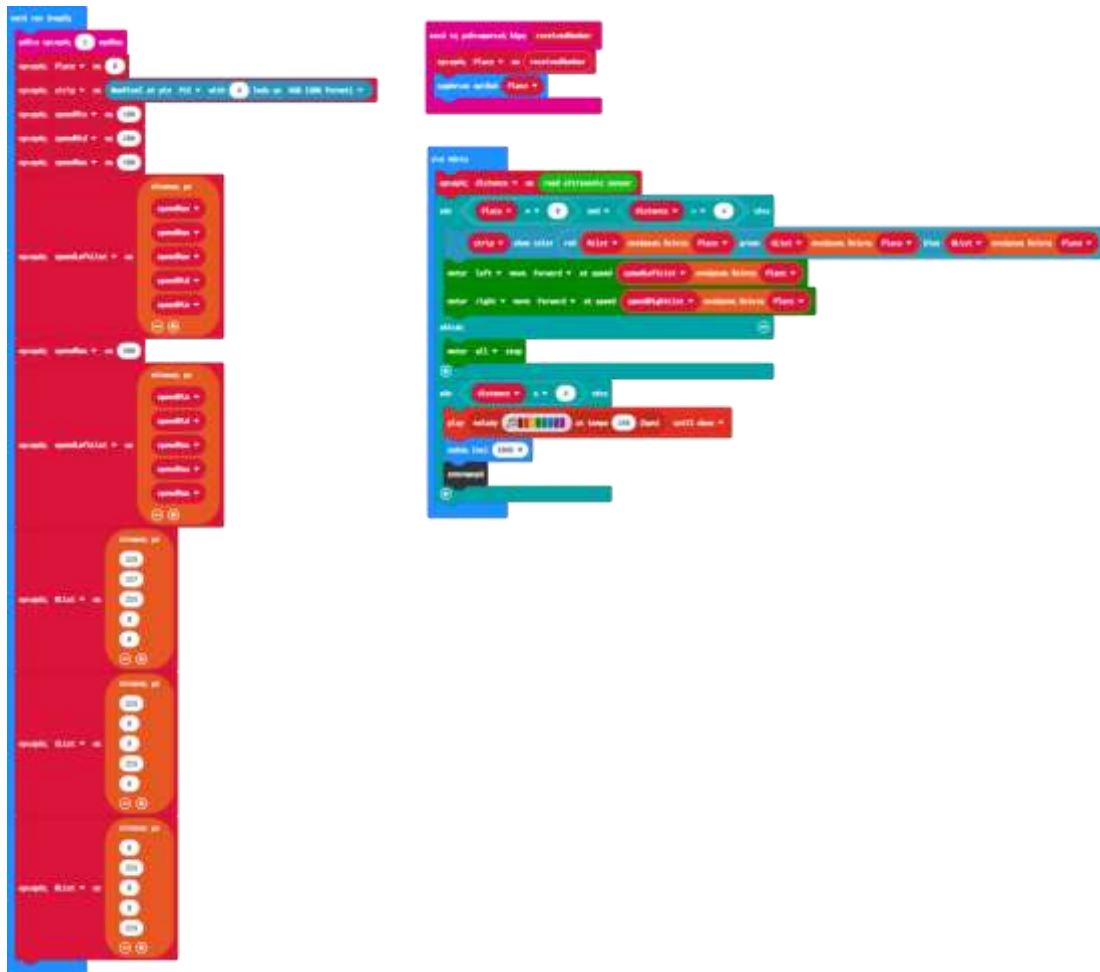


Αυτή τη φορά θα χρησιμοποιήσουμε παράλληλους πίνακες για να δώσουμε κατεύθυνση. Ο δείκτης είναι το Place και οι πίνακες είναι οι:

speedLeftList	speedRightList	Rlist	Glist	Blist
Ταχύτητα	Ταχύτητα	Τιμή	Τιμή	Τιμή
αριστερού	δεξιού τροχού	για	για	για
τροχού		Red	Green	Blue

Place	speedLeftList	speedRightList	Rlist	Glist	Blist	
1	max	min	255	255	0	
2	max	mid	127	0	255	
3	max	max	255	0	0	
4	mid	max	0	255	0	
5	min	max	0	0	255	

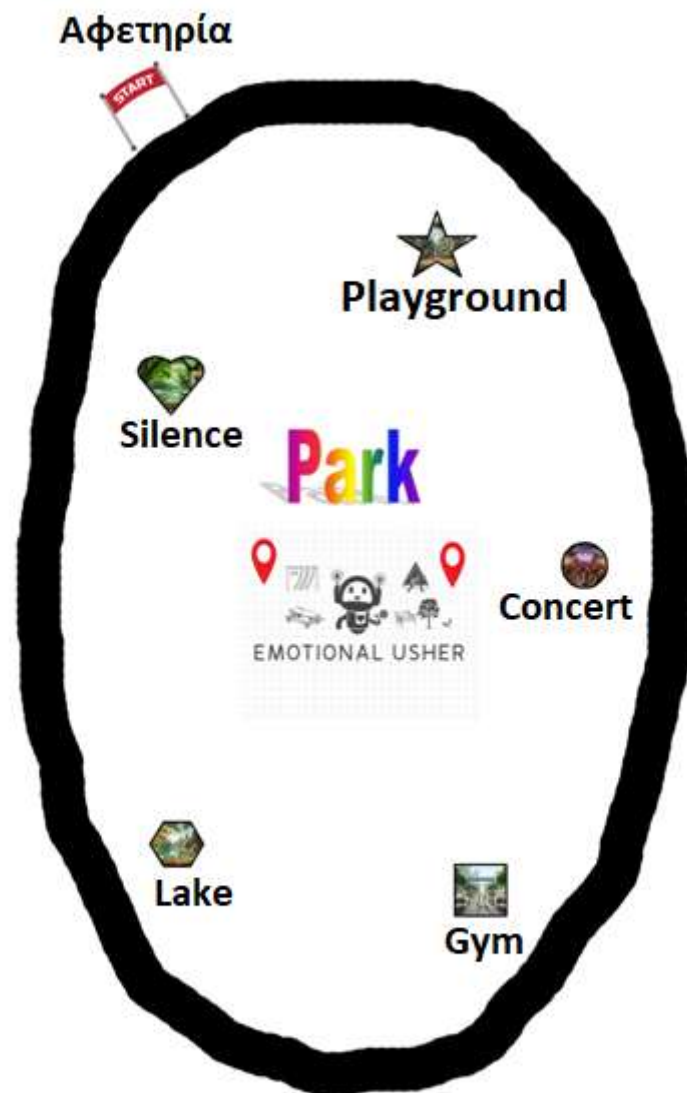
Με το που διαβιβαστεί μέσω **Radio** η κατεύθυνση από το Huskylens, δίνονται οι τιμές της ταχύτητας των τροχών και οι τιμές για το χρώμα του οχήματος. Το όχημα κινείται και ταυτόχρονα διαβάζεται ο Αισθητήρας Υπερήχων (Ultrasonic Sensor). Αν η απόσταση από εμπόδιο είναι μικρότερη των 4cm, φτάσαμε στον προορισμό. Το όχημα σταματά, παίζει ένας ήχος, περιμένει λίγο και μετά κάνει RESET (επαναφορά).





45

Κυκλικό Πάρκο



Σχεδιάσαμε το πάρκο στο **Gimp**. Χαρτί A3 (297mm x 420mm), Resolution: 300px/inch. Μέγεθος trail: **177px** (στην εκτύπωση βγαίνει $\approx 1,5\text{cm}$.)

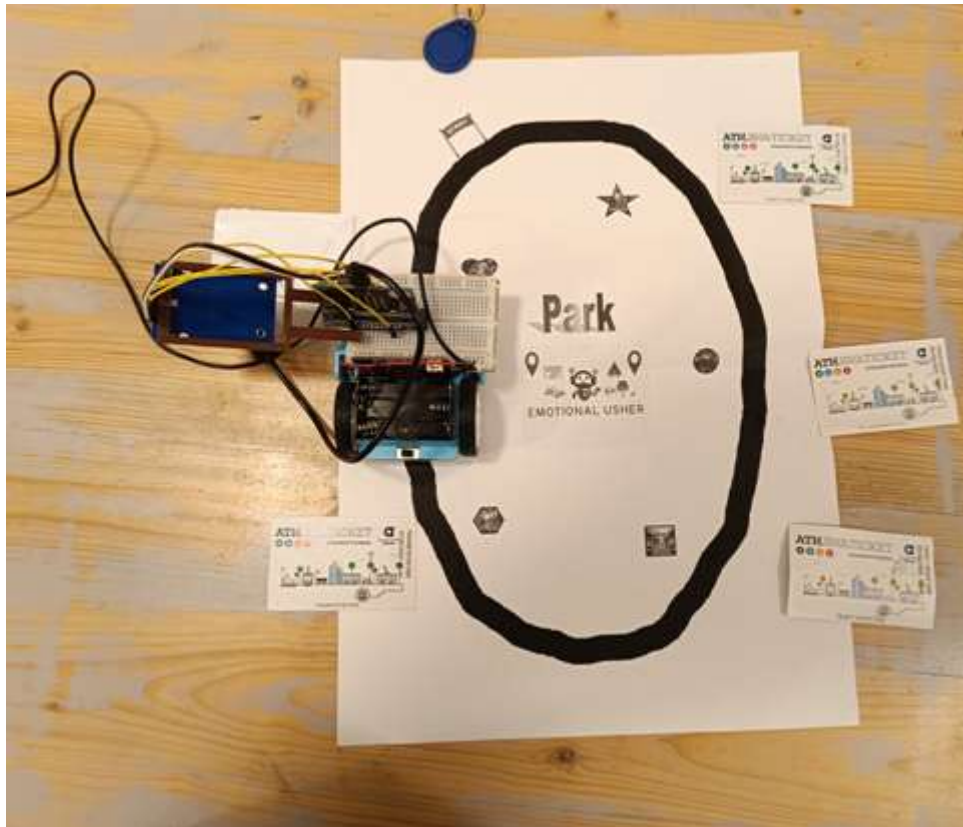


Σε κάθε θέση που πρέπει να σταματήσει το αυτοκινητάκι Maqueen τοποθετήσαμε ένα RFID tag. Δοκιμάσαμε Keyfob, κάρτα και ακυρωμένα εισιτήρια του μετρό!





- Τοποθετήσαμε το micro:bit που διαβάζει το RFID στο μπροστινό μέρος του maqueen.
- (έχουμε 2 micro:bit πάνω στο αυτοκινητάκι).
- Το RFID-MFRC522 Προεξέχει στα αριστερά (έχουμε εκτυπώσει μια βάση στον 3D εκτυπωτή)
- Το maqueen κινείται δεξιόστροφα.



Στο στιγμιότυπο το RFID διαβάζει τη λευκή κάρτα και το maqueen σταματά στην αντίστοιχη θέση (5 - Concert)



48


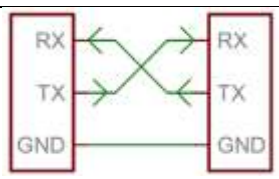


Middleman (Σειριακή Tx, Rx + Bluetooth)

Ενδιάμεσο micro:bit, ανάμεσα στην Huskylens Camera και το Tablet.

(Ασυμβατότητα **Radio Commands** – **Bluetooth**)



Συνδέσεις 		
Huskylens (Sender)	Middleman (Receiver)	
Tx (P1)	Rx(P0)	
GND	GND	

Στο Tablet τρέχει η **Bluetooth App**



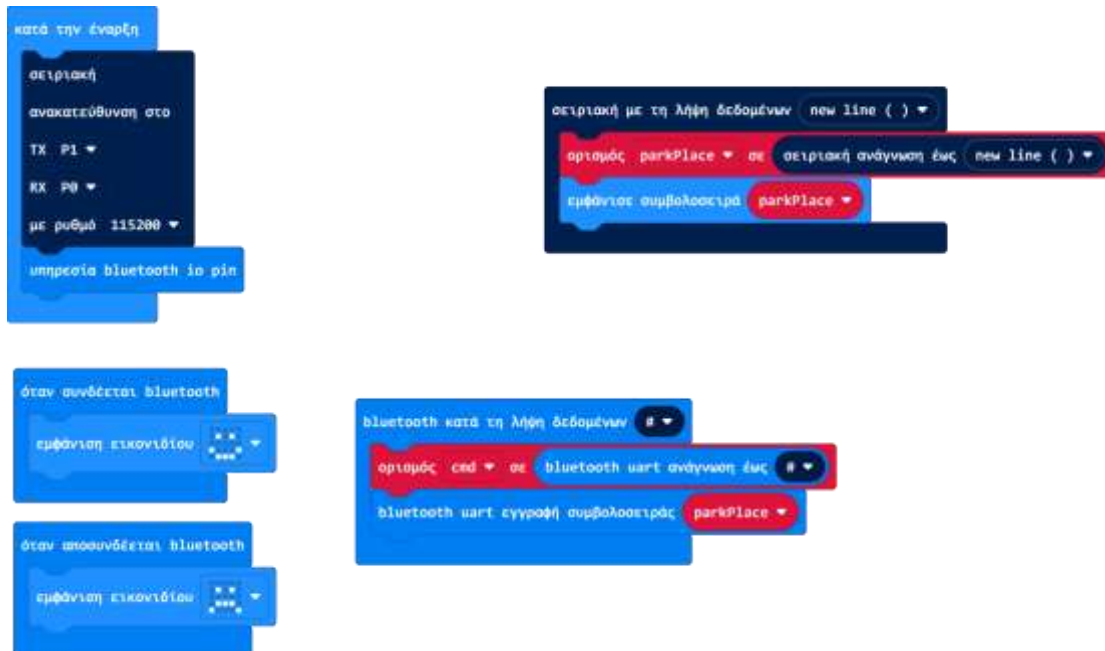
🖥️ Πρόγραμμα για Huskylens



Σημείωση: Μπορούμε να έχουμε **Radio Commands** στη Huskylens, δεν φαίνονται εδώ.



Πρόγραμμα για Middleman



Υπηρεσία Bluetooth io pin → Αυτόματη σύνδεση (microbit + Tablet)

Συνδέουμε το microbit Middleman με το Tablet via Bluetooth.

Από τις ρυθμίσεις του Tablet για το Bluetooth κάνουμε σύνδεση για καλύτερη επικοινωνία.

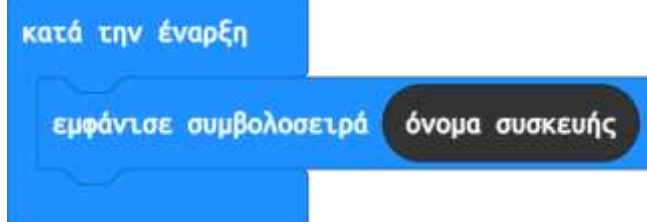
Σημείωση: Αρχικοποίηση Bluetooth microbit : Press (Reset + A +B) – Release Reset

Ενεργοποίηση του Bluetooth του micro:bit

Πατάμε ταυτόχρονα τα A, B reset Αφήνουμε το reset. Στην οθόνη

εμφανίζεται το σήμα του Bluetooth 

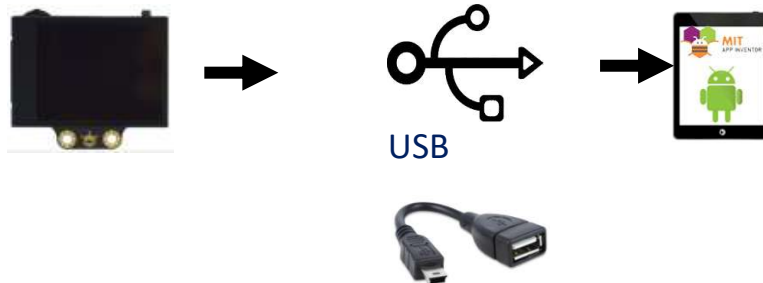
Πως μπορώ να δω το όνομα συσκευής Microbit:





51

Σειριακή USB με καλώδιο OTG



Το πρόγραμμα που τρέχει στο Huskylens

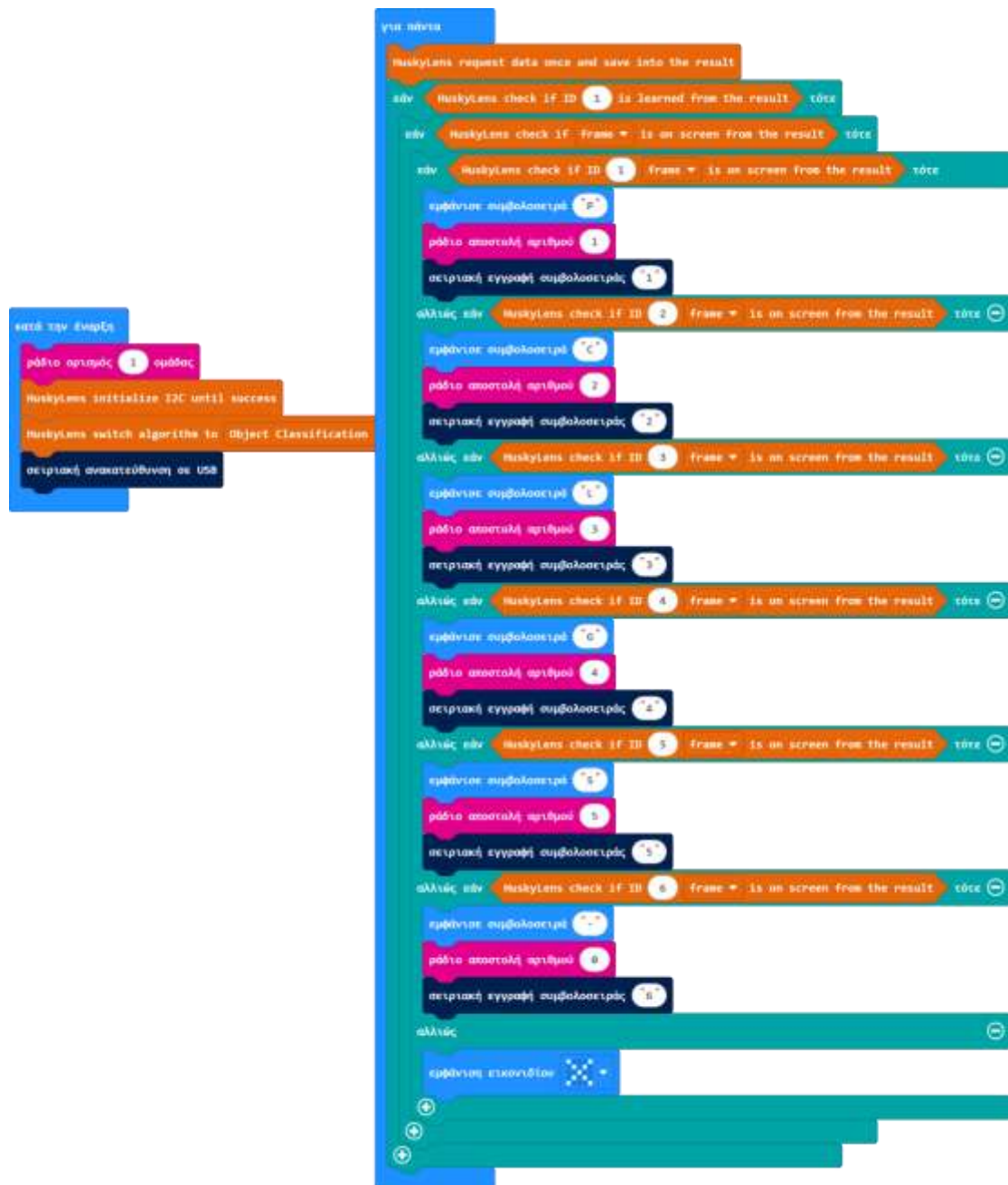
Βάζουμε τις εντολές:

σειριακή ανακατεύθυνση σε USB

σειριακή εγγραφή συμβολοσειράς “ ” (ανάλογα τι αναγνωρίζει η κάμερα)



52



Στο Tablet τρέχει η **USB OTG App**



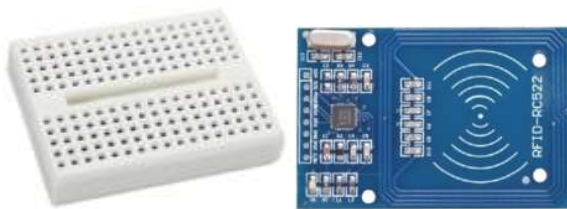
Κυκλικό Πάρκο (τελική διαμόρφωση)

Σε κάθε θέση που πρέπει να σταματήσει το αυτοκινητάκι Maqueen τοποθετήσαμε ένα NFC tag, και συγκεκριμένα ακυρωμένα εισιτήρια του μετρό.



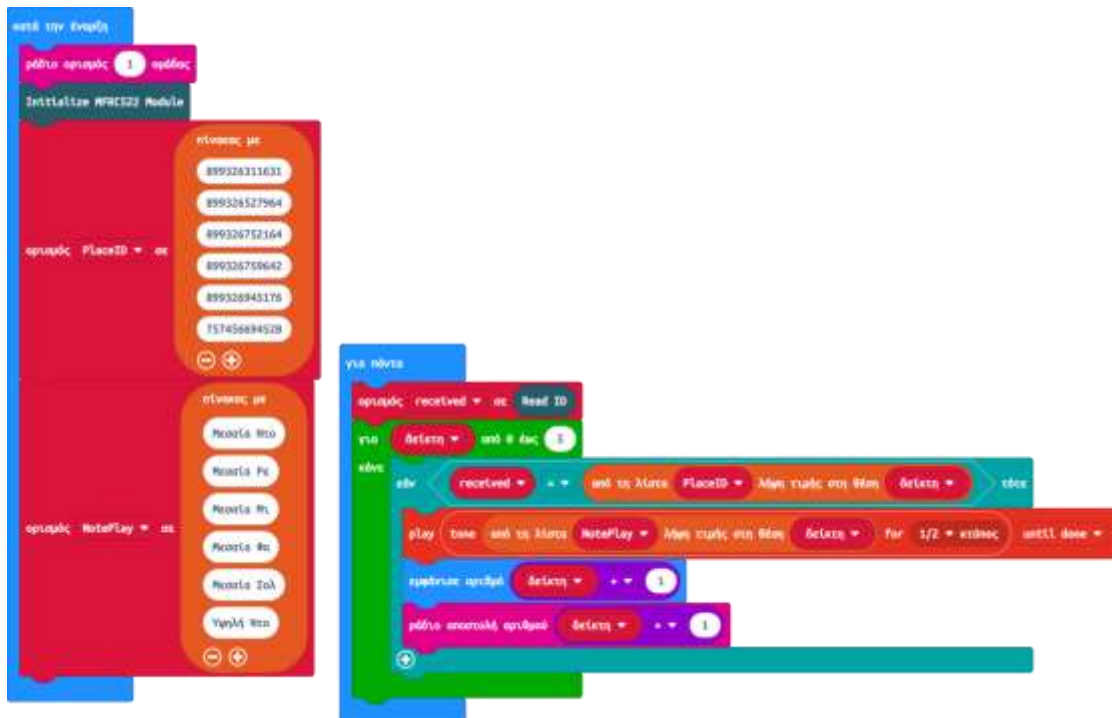


Τοποθετήσαμε το micro:bit «**Αναγνώστη**» που διαβάζει το NFC στο μπροστινό μέρος του maqueen. (έχουμε 2 micro:bit πάνω στο αυτοκινητάκι). Χρησιμοποιήσαμε **mini breadboard 170 tie points** και εκτυπώσαμε 3D μία θήκη-βάση για το **RFID-MFRC522**. Το RFID-MFRC522 προεξέχει μπροστά από το maqueen.



Πρόγραμμα για τον «Αναγνώστη»

Έχουμε 2 πίνακες, έναν με τους αριθμούς ID των εισιτηρίων που χρησιμοποιούμε και έναν με μία νότα που θα παίζει ανάλογα με τον προορισμό. Ο Πίνακας έχει 6 θέσεις, με την τελευταία να είναι η Αφετηρία. Αν ο Αναγνώστης διαβάσει ένα «γνωστό» εισιτήριο, στέλνει κωδικό 1..6 με **Radio**. Ο δείκτης παίρνει τιμές από 0..5, γι' αυτό προσθέτουμε +1.



Πρόγραμμα για το μαqueen

Το μαqueen κινείται διαρκώς πάνω στο μονοπάτι. Αν λάβει (Radio) αριθμό από 1 έως 6, σταματά. Όταν φτάνει στην Αφετηρία, παίζει μία μελωδία, περιμένει λίγο και επανεκκινεί (Reset).





Πυξίδα



Δημιουργήσαμε μία «Πυξίδα» που θα δείχνει την κατεύθυνση που ορίζουμε για τον Ταξιθέτη.



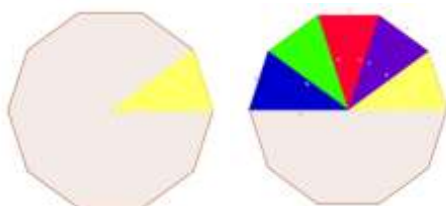
Ο Σχεδιασμός της Πυξίδας έγινε στο Geogebra.



Αρχικά δημιουργήσαμε ένα δεκάγωνο



Στο πάνω μέρος του δεκάγωνου, φτιάχνουμε 5 τρίγωνα και από τις ιδιότητες τα χρωματίζουμε.





Στη συνέχεια κάνουμε αόρατο το δεκάγωνο

Χρησιμοποιήσαμε ένα Micro Servo στη βάση της Πυξίδας. Το Micro Servo συνδέθηκε στο P1 του micro:bit που ελέγχει την κάμερα Huskylens



huskylensCompass.hex

Το πρόγραμμα αυτό στέλνει με Radio στον Ταξιθέτη, με Σειριακή USB (OTG καλώδιο) στο Tablet και κινεί τον δείκτη της πυξίδας ώστε να δείχνει την κατεύθυνση

1 Playground	2 Concert	3 Gym	4 Lake	5 Silence
18°	54°	90°	126°	162°

```

για πάντα
  Huskylens request data once and save into the result
  εάν Huskylens check if ID 1 is learned from the result τότε
    εάν Huskylens check if frame = 1 is on screen from the result τότε
      εάν Huskylens check if ID 1 frame = 1 is on screen from the result τότε
        αριθμός TakenDetected = 1
        set servo P1 = angle to 18
      αλλιώς εάν Huskylens check if ID 2 frame = 1 is on screen from the result τότε
        αριθμός TakenDetected = 2
        set servo P1 = angle to 54
      αλλιώς εάν Huskylens check if ID 3 frame = 1 is on screen from the result τότε
        αριθμός TakenDetected = 3
        set servo P1 = angle to 90
      αλλιώς εάν Huskylens check if ID 4 frame = 1 is on screen from the result τότε
        αριθμός TakenDetected = 4
        set servo P1 = angle to 126
      αλλιώς εάν Huskylens check if ID 5 frame = 1 is on screen from the result τότε
        αριθμός TakenDetected = 5
        set servo P1 = angle to 162
      αλλιώς εάν Huskylens check if ID 6 frame = 1 is on screen from the result τότε
        αριθμός TakenDetected = 6
        set servo P1 = angle to 18
      αλλιώς
        εμφάνισε εικόνα 0
    τέλος
  τέλος

```



Κατάλογος Προγραμμάτων

Maqueen Line Track		
Maqueen Line Track with Flag		
HuskyLens OTG		
HuskyLens Bluetooth		
Ακτινωτό Πάρκο Shuttle Bus		
Ακτινωτό Πάρκο Rent A Scooter		
Κυκλικό Πάρκο		
Simple App		
Bluetooth App		
USB OTG App		
NFC Ανάγνωση		
NFC Εισιτήρια		



Κατάλογος Videos

01RadialParkShuttleBusTraceTracks	Σχεδίαση Ακτινωτού Πάρκου: Σημειώνουμε τα ίχνη των διαδρομών
02RadialParkColors	Χρώματα στο κάτω μέρος του Ταξιθέτη
03RadialParkShuttleBusTestWithOTG	Δοκιμές του Ταξιθέτη με το Tablet (OTG)
04RadialParkShuttleBusWorking	Υλοποίηση του Ακτινωτού Πάρκου
05TestingObstaclesHC_SR04	Δοκιμάζουμε τον Αισθητήρα Υπερήχων HC SR-04 εμπόδια
06ObjectRecognitionVideo	Αρχικές Δοκιμές κάμερας Α.Ι.
07InstructionsTokenTinkercadHD	Οδηγίες Σχεδίασης Tokens (Μάρκες Εισόδου) στο Tinkercad
08CircularParkNFC	Κυκλικό Πάρκο με NFC & Εισιτήρια

