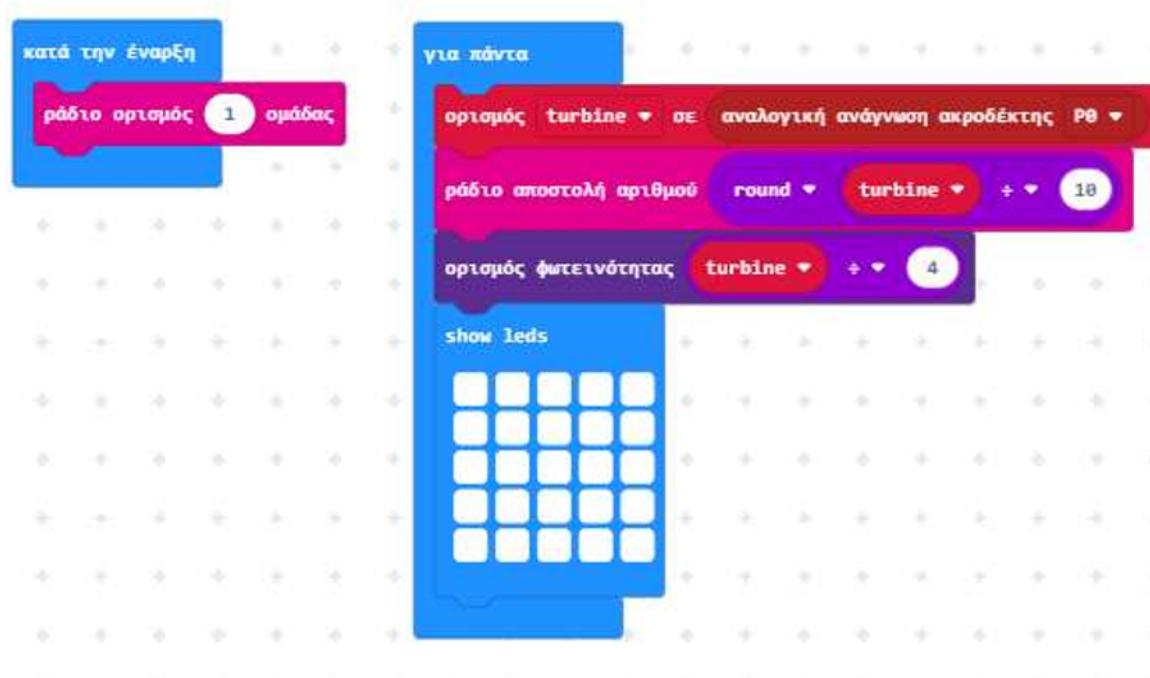


Transmitter only

Η ανεμογεννήτρια στέλνει στον καταναλωτή ενέργεια.

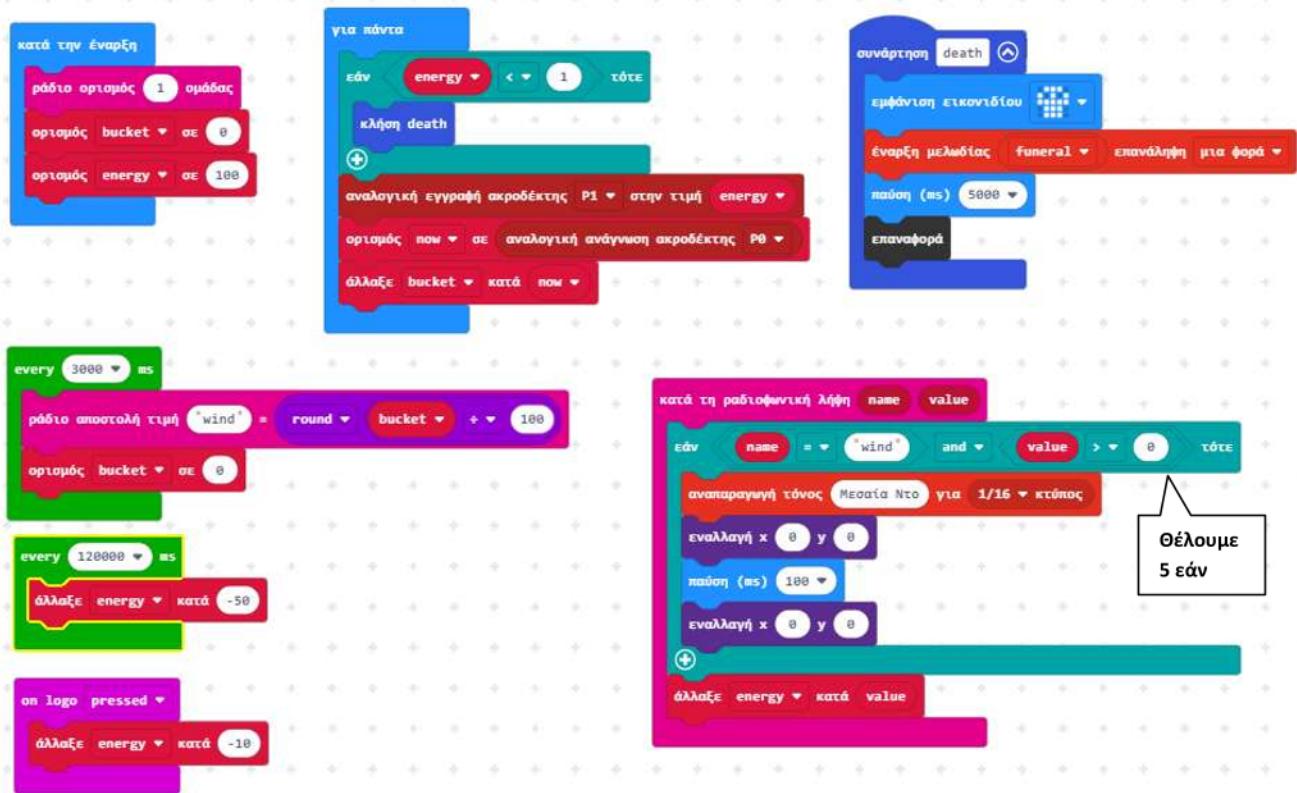
Αν έχω 4 καταναλωτές μοιράζω δια τέσσερα την ενέργεια πριν την στείλω. Ισοκατανομή ενέργειας.



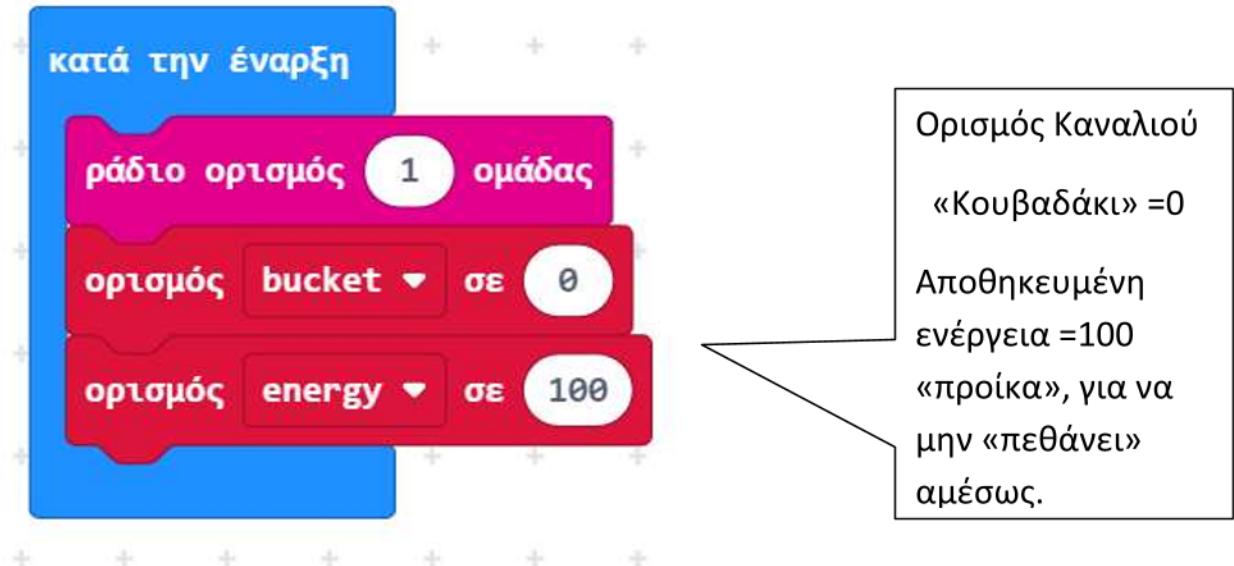
Εδώ είναι υπολογισμένο για **έναν** καταναλωτή.

**Ανεμογεννήτρια – Γενικό Υπόδειγμα για όλους τους παραγωγούς ενέργειας
Παραγωγή – Αποστολή – Λήψη – Αποθήκευση – Κατανάλωση Ενέργειας**

(Παράδειγμα Ανεμογεννήτρια)

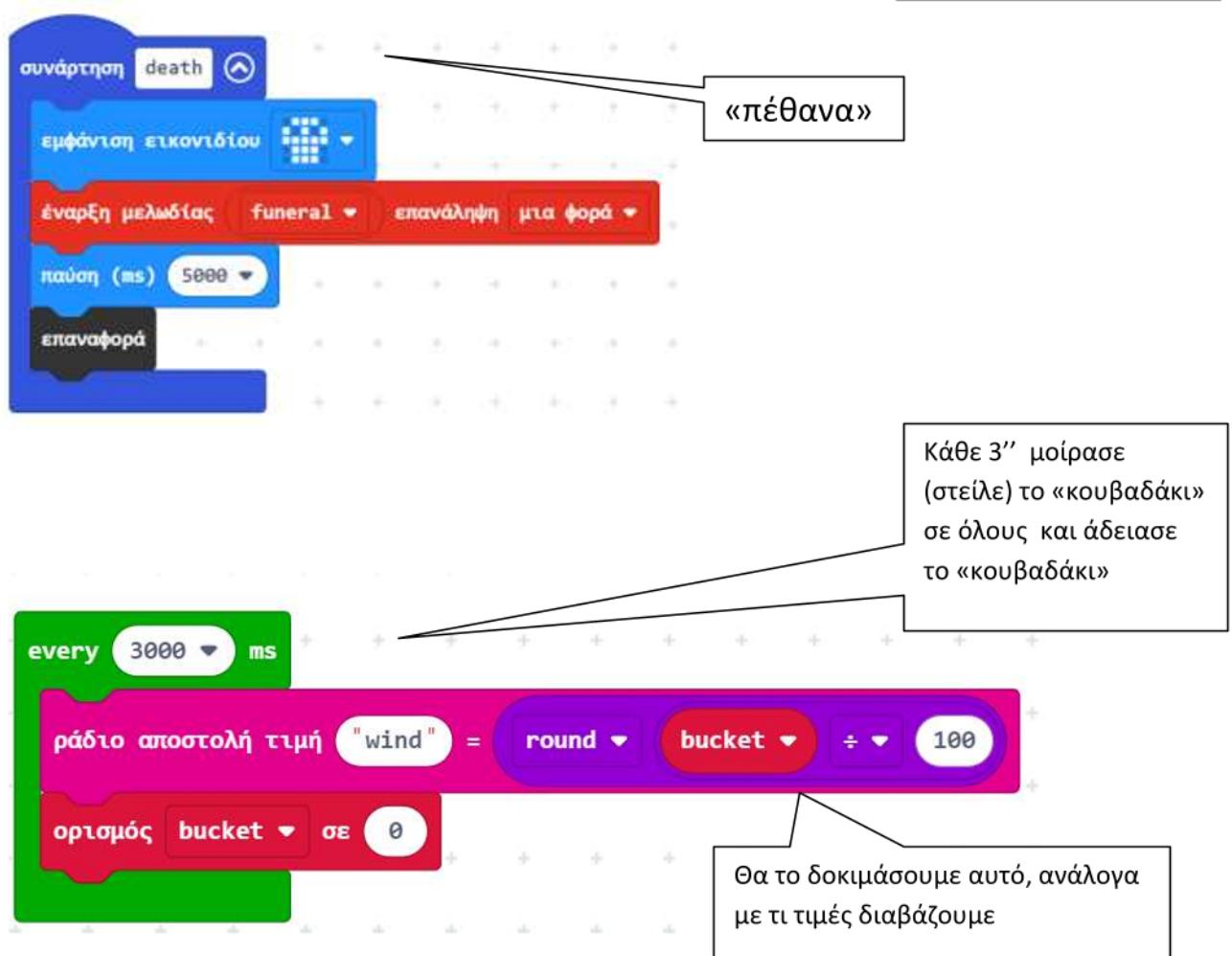
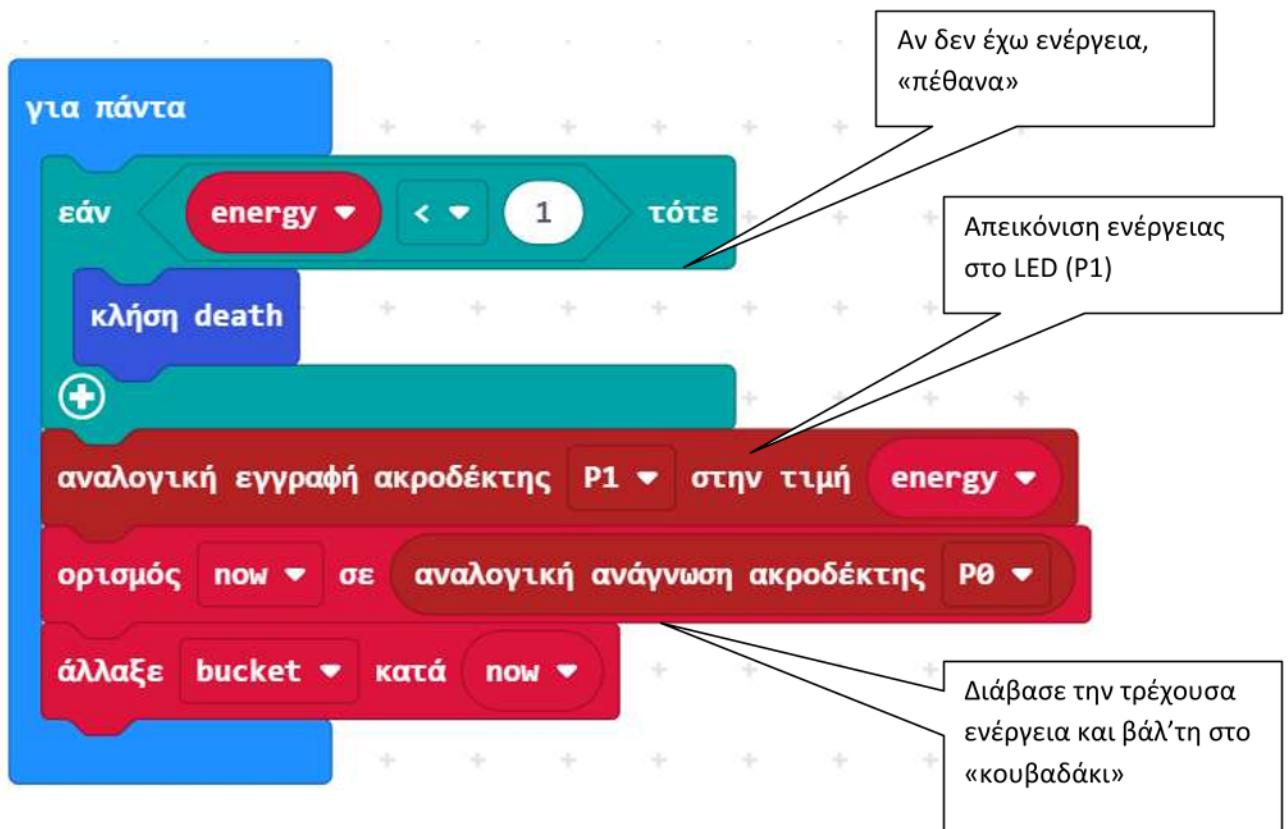


allEnergies.hex



Αναστασία Κωνσταντέλου





Για να μη συντονιστούν όλα τα micro:bit, θα βάλουμε **πρώτους** αριθμούς

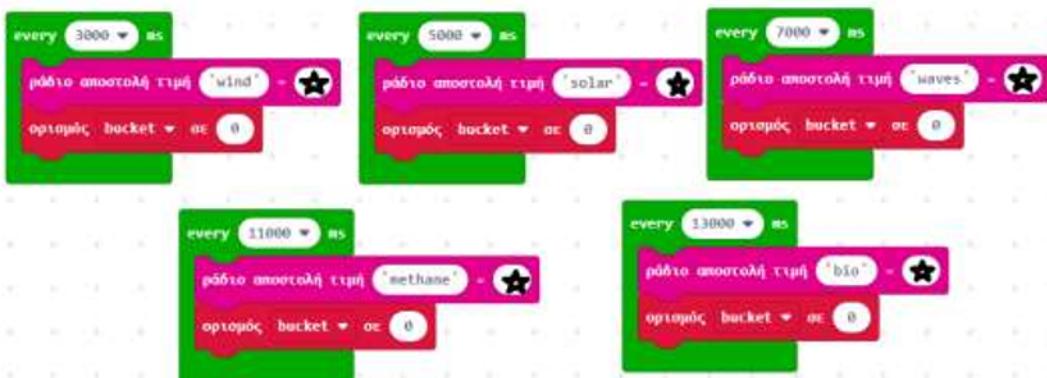
A Scratch script consisting of two blocks:

- An "every [] ms" control loop.
- Inside the loop:
 - A green "radio received" sensor block with the value "wind".
 - An assignment block: "radio received" = $\text{round}(\text{bucket} \div 100)$.
 - A green "set bucket to [0]" control block.

Below the script is a table:

name	time
wind	3000ms
solar	5000ms
waves	7000ms
methane	11000ms
bio	13000ms

Σε κάθε «Σπίτι» θα βάλουμε από μία τέτοια εντολή.



Σε κάθε αστέρι θα δοκιμάσουμε, ανάλογα τις τιμές που διαβάζουμε.

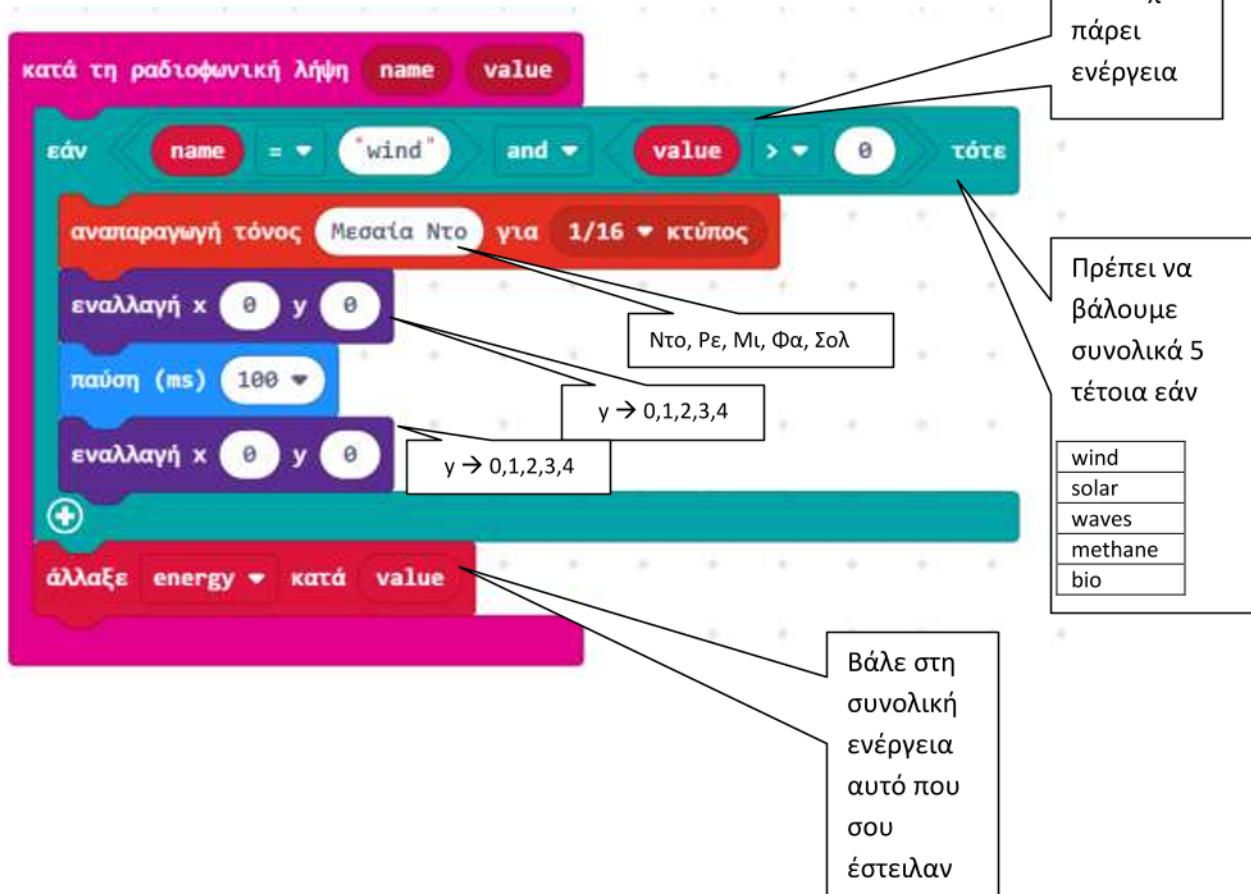
Το bucket μπορεί να έχει τιμές από 0 έως 1023. Το πιθανότερο είναι οι τιμές να φτάνουν μέγιστο το 350, οπότε θα τα διαμορφώσουμε κατάλληλα, ανάλογα με το τι θα διαβάζουμε από κάθε «Σπίτι».

Προσοχή! Στο **waves** διαφοροποιείται ο τρόπος που διαβάζουμε την ενέργεια. Μετράμε κινήσεις της «τραμπάλας», και όχι το pin P0.

Επίσης, κάθε 2 λεπτά (120000ms) θα έχουμε **απώλεια ενέργειας**, είτε καταναλώνουμε είτε όχι. Γι' αυτό το λόγο αρχικά «προικίζουμε» το κάθε σπίτι με ενέργεια, ώστε να μην «πεθάνει» αμέσως.



Όταν το «Σπίτι» λαμβάνει κάποια ενέργεια, αυτή απεικονίζεται στα πάνω led του micro:bit.



Προσοχή! **Τα εάν πρέπει να είναι 5 συνολικά**, όσα και τα σπίτια.



Τέλος, το κάθε σπίτι θα είναι καταναλωτής. Όταν πατάμε το logo θα αφαιρείται κάποιο ποσό ενέργειας.



Παραγωγός – Πομπός Ενέργειας

The image shows a Scratch script for wind energy production and water pumping. The script uses the 'when green flag clicked' hat and various sensors and operators blocks.

- Top Left:** A callout box says: "Ορίζουμε κανάλι. Αρχικοποιούμε το «κουβαδάκι» μερικής συγκέντρωσης ενέργειας". It points to a blue 'when green flag clicked' hat with a 'set [wind v] to [0]' block.
- Top Right:** An icon of a wind turbine is shown with a callout box: "Διάβασε την τρέχουσα ριπή και βάλτη στο «κουβαδάκι»". It points to a red 'if then' block with a 'set [wind v] to [value]' block.
- Middle Left:** A callout box says: "Κάθε 10'' στέλνουμε την ενέργεια που έχει παραχθεί και την συγκεντρώσαμε στο «κουβαδάκι». Αδειάζουμε το «κουβαδάκι»." It points to a green 'every [1000 ms]' loop with a 'say [] for []' block.
- Middle Right:** A callout box says: "Γράφημα της ριπής. Το 20 είναι η μέγιστη ριπή ανέμου. Το δοκιμάζουμε". It points to a blue 'plot bar graph of [blow v]' block with a 'up to [20]' parameter.
- Bottom Left:** An icon of a wind turbine is labeled "ριπή".
- Bottom Middle:** An icon of a bucket is labeled "Μερική συγκέντρωση στο «κουβαδάκι»".
- Bottom Right:** An icon of a water drop falling from a pipe is labeled "Αποστολή + Άδειασμα «κουβαδάκι», πάμε πάλι".
- Bottom Center:** A large black arrow points from left to right, indicating the flow of the process.
- Bottom Center:** A callout box says: "Στρογγυλεύουμε και διαιρούμε δια 10 την ενέργεια που στέλνουμε. (Δεν θέλουμε μεγάλο νούμερο)". It points to a purple 'round [wind v] / [10]' block.
- Bottom Center:** A callout box says: "Μπορούμε επιπρόσθετα να διαιρέσουμε με τον αριθμό των αποδεκτών της ενέργειας για να μοιράσουμε ισόποσα". It points to a green 'if [wind >= [10]] then []' block.

Ορίζουμε δύο μεταβλητές: Μία για τις **ριπές** του ανέμου(**blow**) και μία για τη **μερική συγκέντρωση** της ενέργειας «κουβαδάκι» (**wind**) που θα στείλουμε. Κάθε 10'' (10000ms) αποστέλλουμε τη μερική συγκέντρωση της ενέργειας και αδειάζουμε τη μερική συγκέντρωση. Ουσιαστικά η μεταβλητή **wind** είναι ένα «κουβαδάκι» που συγκεντρώνουμε την παραχθείσα ενέργεια και όταν την αποστέλλουμε τότε το αδειάζουμε.

Αποδέκτης – Καταναλωτής Ενέργειας

totalEnergy:
Συνολική αποθηκευμένη ενέργεια.

Πρόσθεσε στην Συνολική αποθηκευμένη ενέργεια αυτό που έλαβες

Ανεμογεννήτρια



Επίτηδες 1 για να μην «πεθάνει».

όταν πιεστεί το πλήκτρο button A
όλλαξε totalEnergy ← κατά -1

όταν πιεστεί το πλήκτρο button B
όλλαξε totalEnergy ← κατά -2

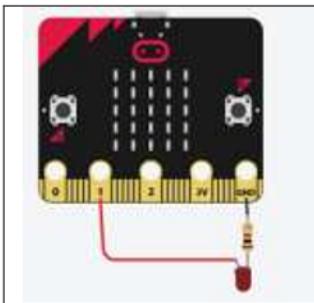
όταν πιεστεί το πλήκτρο button A + B
όλλαξε totalEnergy ← κατά -5

Κατανάλωση ενέργειας.
Δοκιμάζουμε τους αριθμούς.

Power LED. «Όσο πιο πολλή αποθηκευμένη ενέργεια, τόσο πιο πολύ φωτίζει. Συνδεδεμένο στο P1. Αναλογικό.

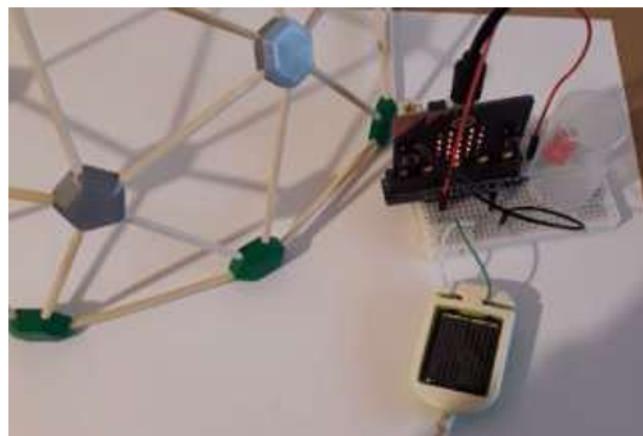
Έλεγχος ενέργειας.
Αν την καταναλώσουμε όλη, «καίγαμε».

Power LED σύνδεση



Αντίσταση 100Ω στην κάθοδο
(Συνήθως βάζουν αντίσταση 470Ω στην άνοδο)

Ηλιακή Ενέργεια



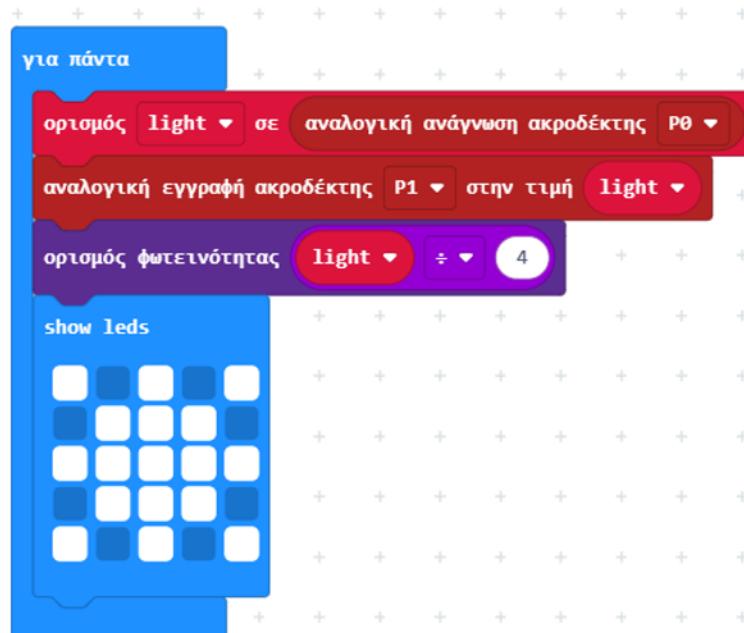
Συνδέουμε τον ηλιακό συλλέκτη στο P0. Συνδέουμε ένα LED στο P1.

Μεταβλητή **light** παίρνει τις τιμές που διαβάζουμε από τον ηλιακό συλλέκτη.

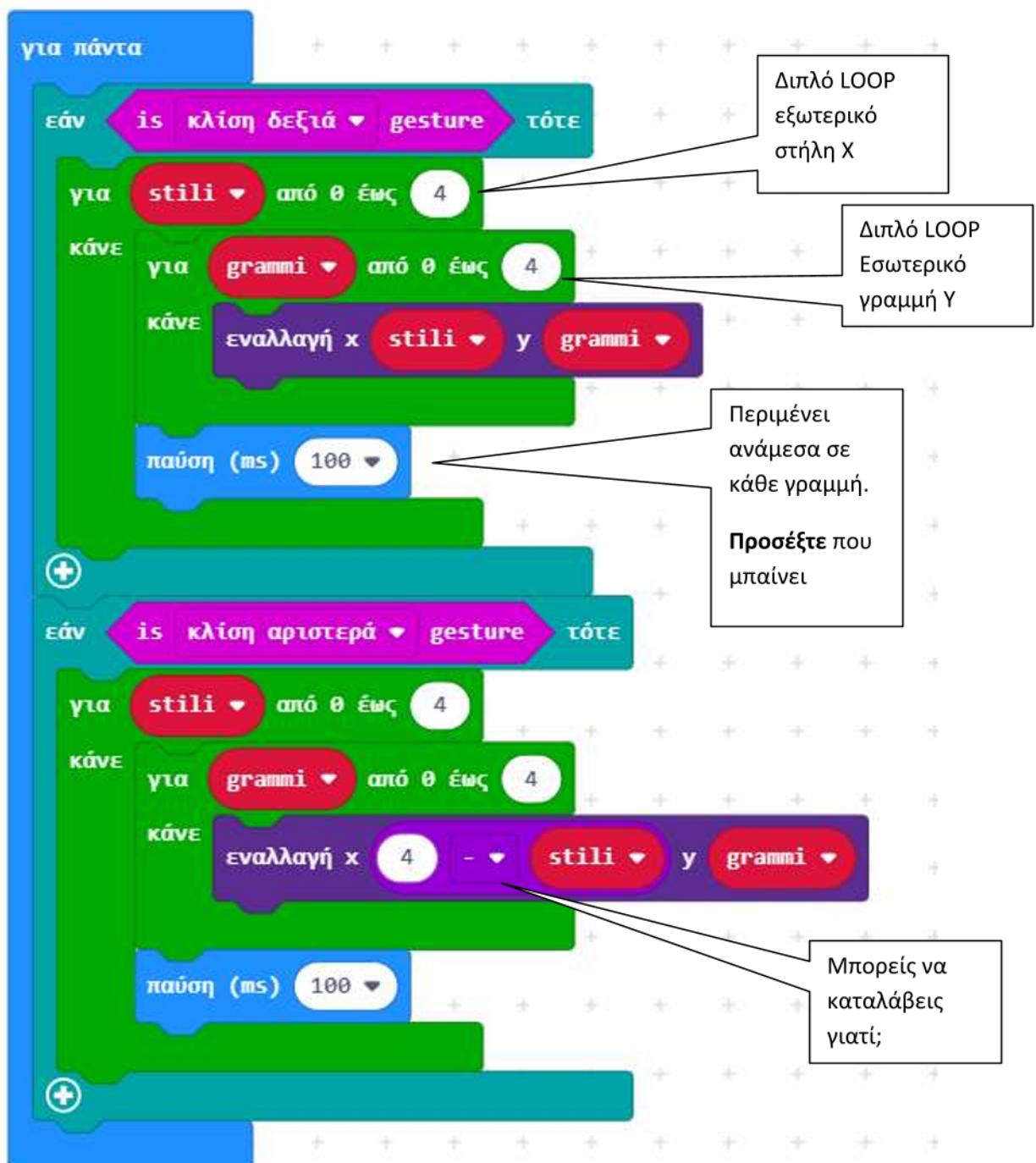
Η αναλογική ανάγνωση (**light**) μας δίνει τιμές από 0 έως 1023.

Ότι διαβάζουμε από το P0, το διοχετεύουμε στο P1.

Η φωτεινότητα παίρνει τιμές από 0 έως 255. Υπάρχει μια αναλογία 4 προς 1, γι' αυτό διαιρούμε το **light** με 4.



Ενέργεια από Παλιρροϊκό κύμα – Προσομοίωση Κύματος



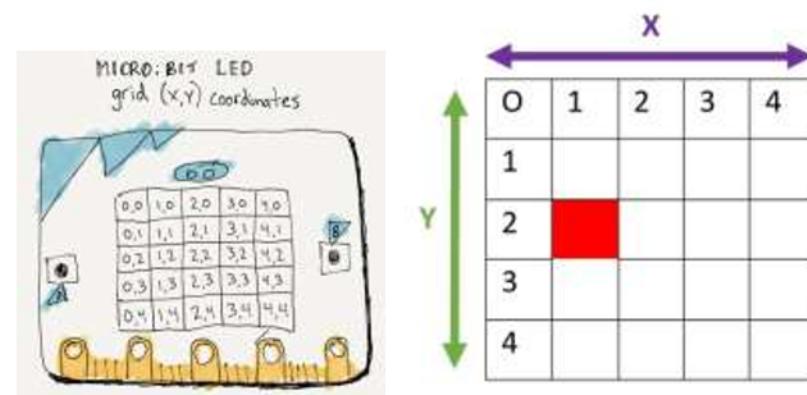
Ορίζουμε **Μεταβλητές** stili → X, grammi → Y

(με λατινικούς χαρακτήρες τα ονόματα των μεταβλητών!)

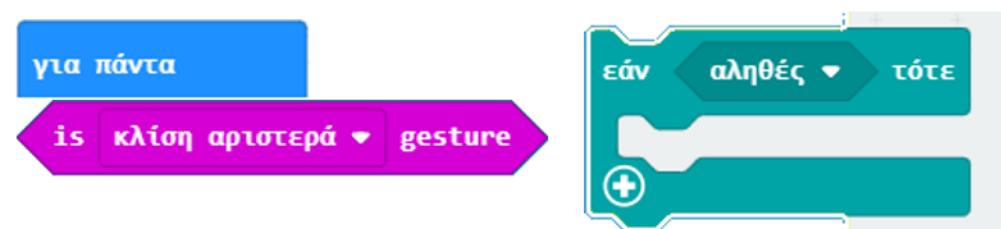
Είναι πολύ σημαντικό τα ονόματα των μεταβλητών να μας θυμίζουν τι αντιπροσωπεύουν.

Βασική εντολή:

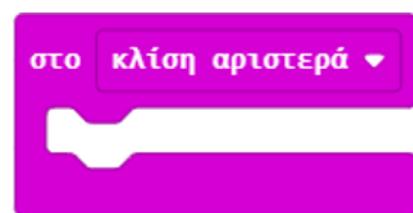
Όταν είναι αναμμένο το Led, το σβήνει και αντίστροφα



?? Γιατί χρησιμοποιούμε τις εντολές:



Αντί για...



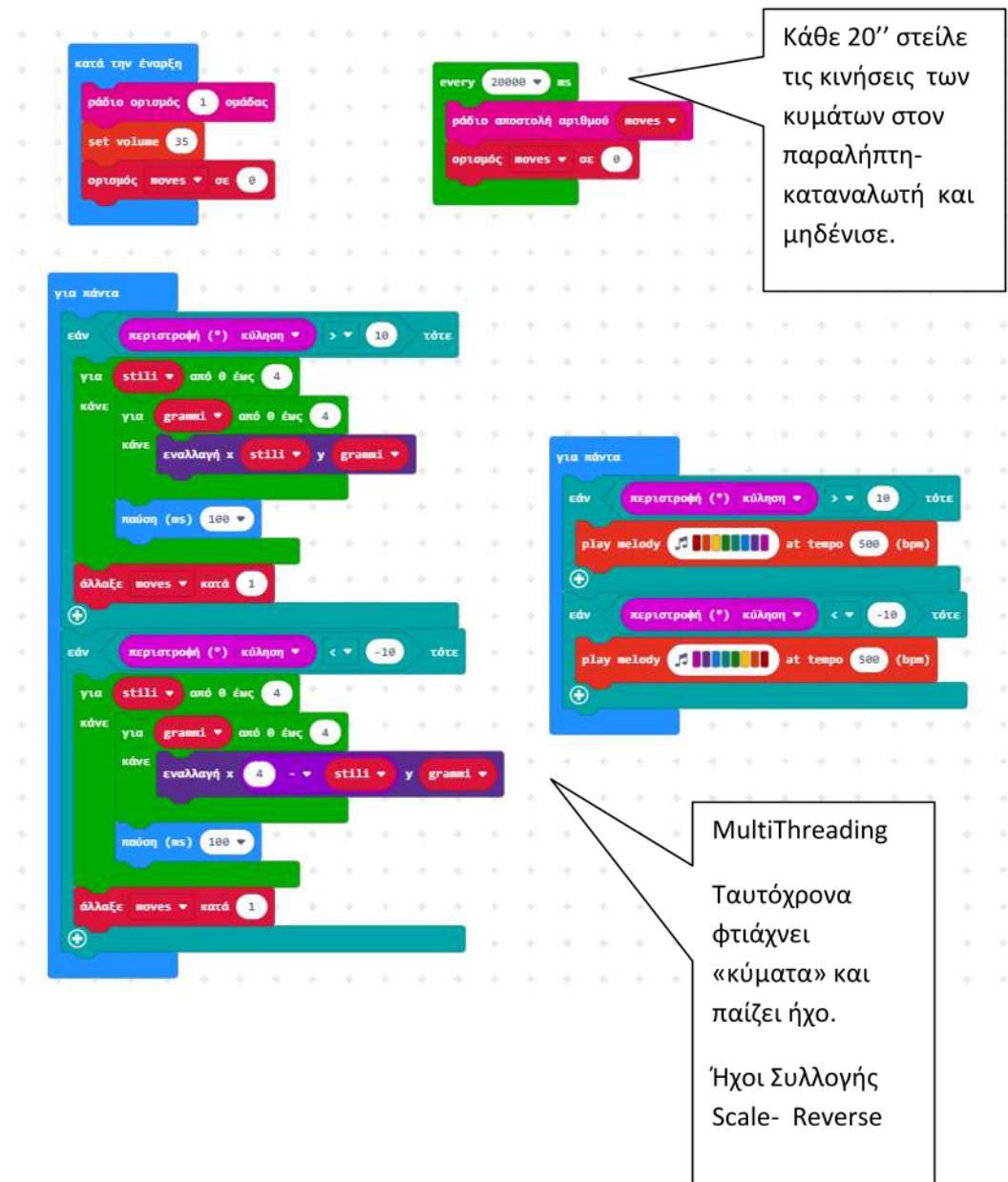
Παραγωγή ενέργειας από Παλιρροϊκό κύμα

Tidal Energy Produce and Receive

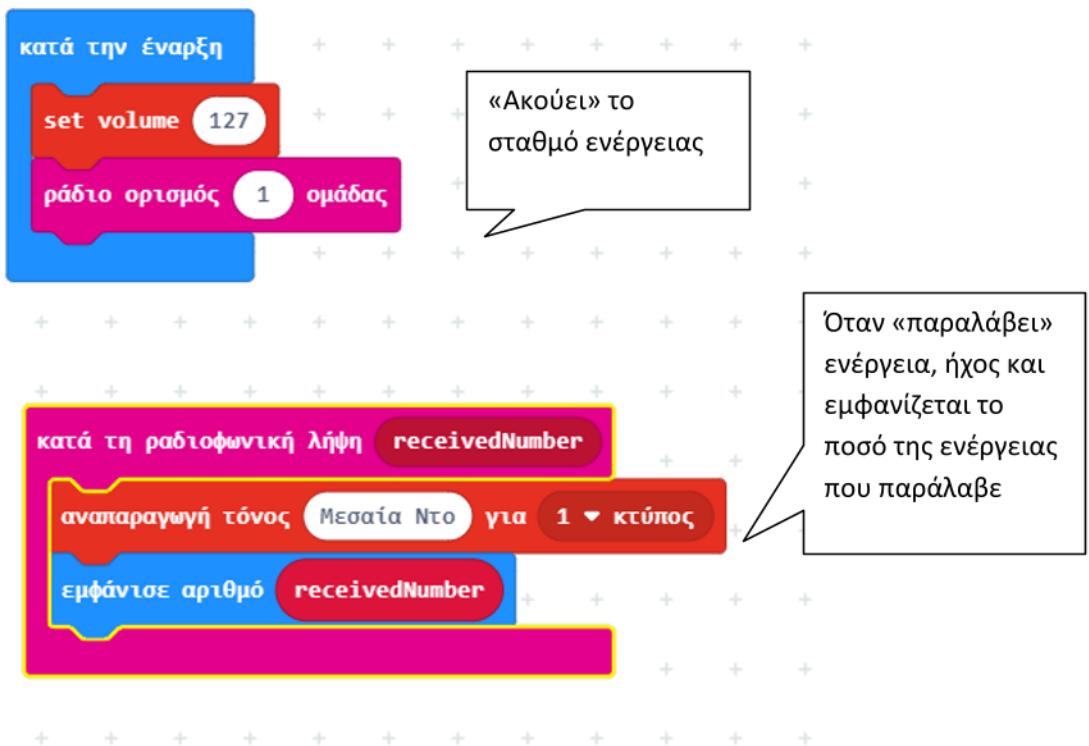
Χρησιμοποιούμε τα κύματα για να παράγουμε ενέργεια (δες στην πίσω σελίδα την ιδέα).

Micro:bit – Παραγωγός (Υδρογεννήτρια – «τραμπάλα»)

Produce – Transmit



Micro:bit – παραλήπτης και καταναλωτής Receive -Consumer



Βιοενέργεια Bitter Orange Battery

Orange electricity!

Τα νεράντζια είναι κάτι κοινό. Ειδικά στην Αττική, σε πολλά πεζοδρόμια μπορούμε να βρούμε νεραντζιές.

Να ένας τρόπος να χρησιμοποιήσουμε τα νεράντζια!

Θα χρειαστούμε:

				
4 νεράντζια	4 κέρματα	4 βίδες	5 κροκοδειλάκια	1 LED

4 **νεράντζια**

1 κροκοδειλάκι **κόκκινο**

1 κροκοδειλάκι **μαύρο**

3 κροκοδειλάκια σε ό,τι άλλο χρώμα θες

4 κέρματα των 5 cent

4 βίδες μεγάλες

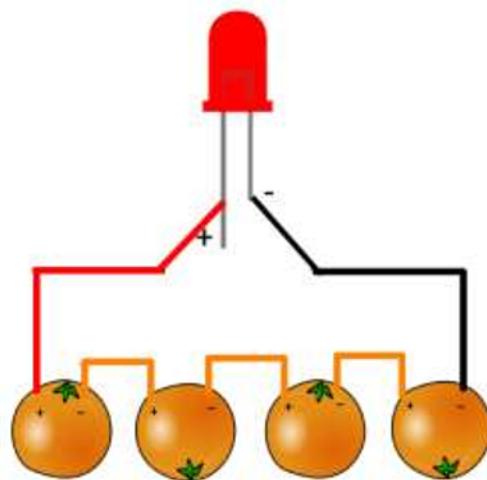
1 LED

- Σε κάθε νεράντζι βάλε ένα κέρμα και μια βίδα.
- Το νεράντζι αυτό είναι μια μπαταρία. Θετικός πόλος (+) το κέρμα, αρνητικός πόλος (-) η βίδα.
- Τοποθέτησε τα νεράντζια εναλλάξ, όπως βλέπεις στην εικόνα. Πάνω βίδα, κάτω κέρμα // πάνω κέρμα, κάτω βίδα.

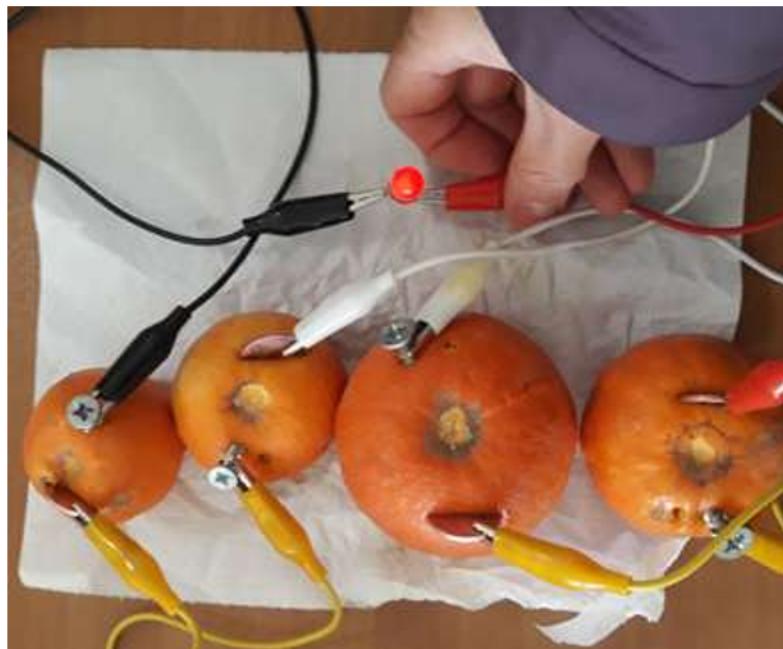


- Άσε ελεύθερο το κέρμα του πρώτου νεραντζιού.

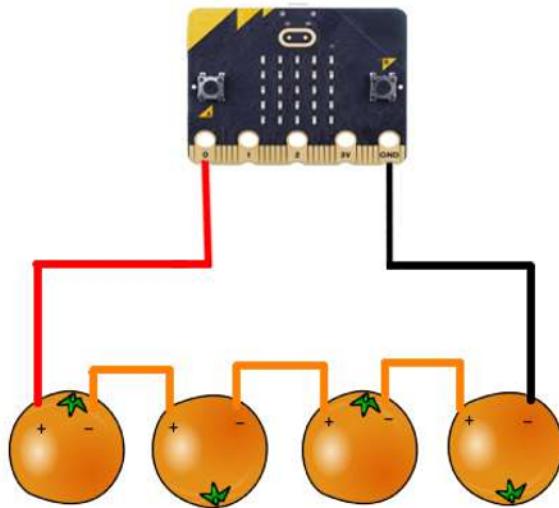
- Ένωσε τη βίδα του ενός με το κέρμα του διπλανού και αντίστροφα. Θα χρειαστείς 3 κροκοδειλάκια μόνο. (βλέπε πορτοκαλί ενώσεις στην εικόνα).
- Βάλε το **κόκκινο** κροκοδειλάκι στο κέρμα του πρώτου νεραντζιού και το **μαύρο** κροκοδειλάκι στη βίδα του τελευταίου νεραντζιού.
- Σύνδεσε την άλλη άκρη από το **μαύρο** κροκοδειλάκι στο κοντό ποδαράκι του LED (-)
- Σύνδεσε την άλλη άκρη από το **κόκκινο** κροκοδειλάκι στο μακρύ ποδαράκι του LED (+)



Άναψε το λαμπάκι!



Εφαρμογή στο micro:bit



Βάλε το + (κόκκινο κροκοδειλάκι) στο P0 και το μαύρο κροκοδειλάκι στο GND.

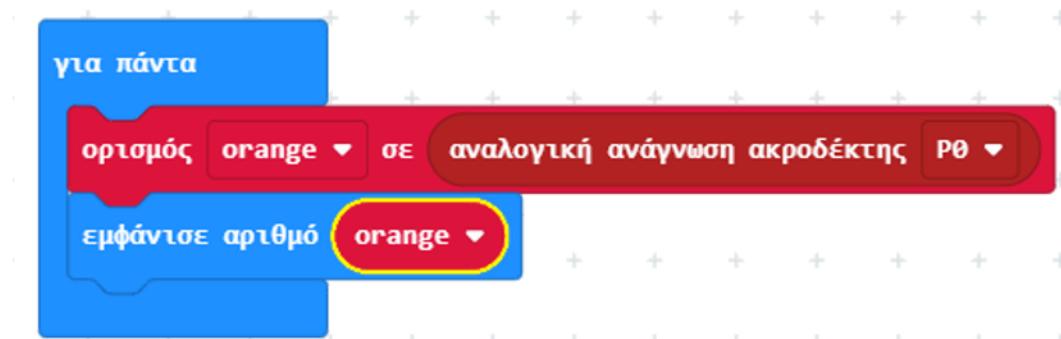
Γράψε το πρόγραμμα στο micro:bit

Θα χρειαστείς εντολές από τις παρακάτω ομάδες εντολών:

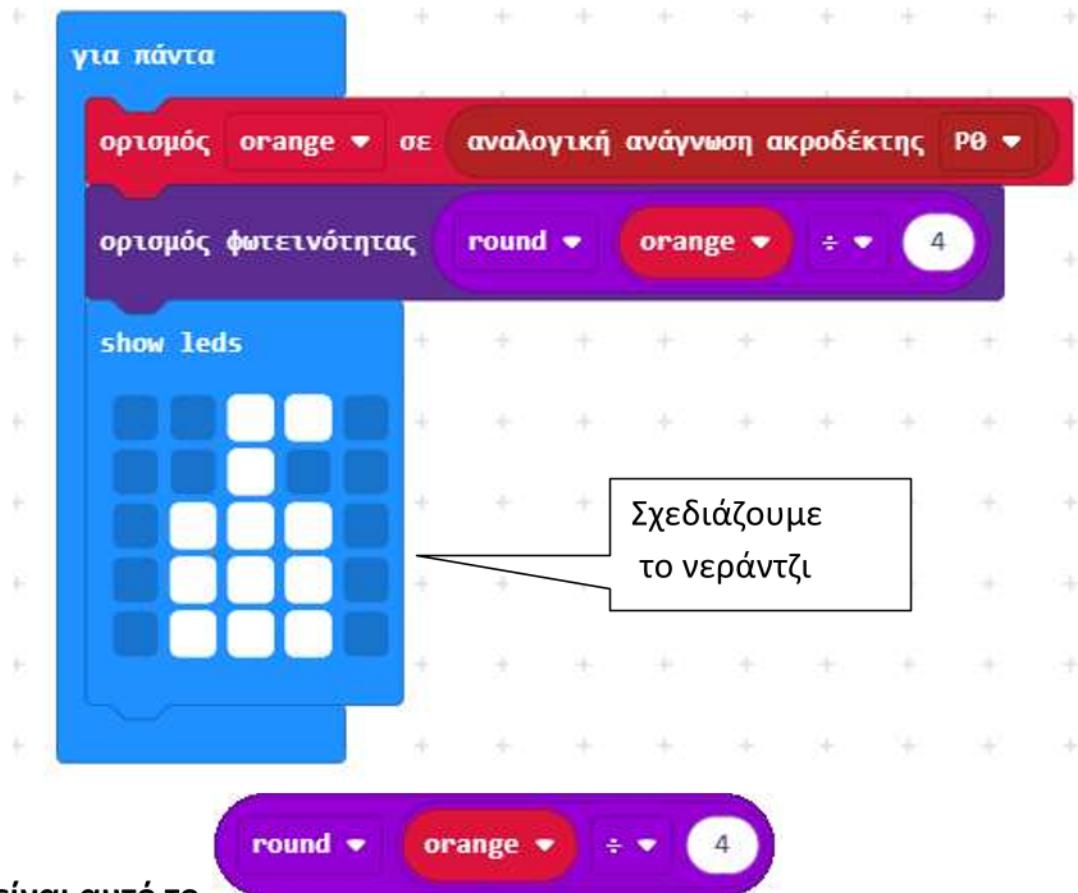
■■■ Βασικά ■■■ Μεταβλητές ◎ Ακροδέκτες

○ Led → ... περισσότερα ■■■ Μαθηματικά

Δημιουργησε μια **μεταβλητή orange** που θα διαβάζει την «ενέργεια» των νεραντζιών. Το μέγιστο που μπορεί να διαβάσει το P0 είναι 1023. Τι «δύναμη» έχει η μπαταρία από νεράντζια που έφτιαξες;



Μπορείς να δοκιμάσεις να εμφανίζεται με εικονίδιο και με αντίστοιχη φωτεινότητα η ενέργεια των νεραντζιών. Όσο πιο ισχυρή, τόσο πιο φωτεινή η οθόνη του micro:bit.



Γίνεται πρώτα η πράξη **orange ÷ 4**

?? Γιατί **orange ÷ 4**

orange παίρνει τιμές από 0 – 1023

Φωτεινότητα παίρνει τιμές από 0 – 255

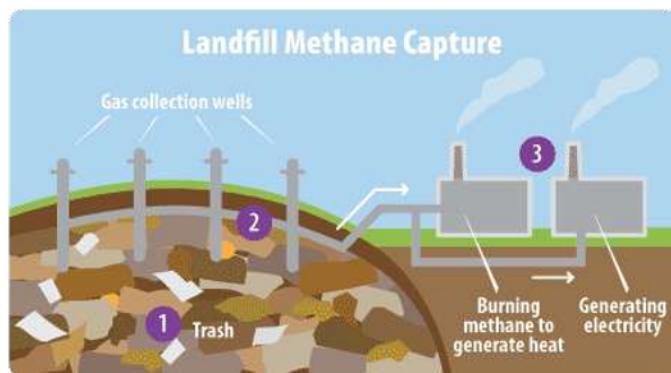
Αναλογία $4 \div 1$

?? **round** Προσαρμόζουμε το **orange** στη **Φωτεινότητα**.

Επειδή θέλουμε **ακέραιο** αριθμό για την φωτεινότητα, το στρογγυλοποιούμε με το **round**

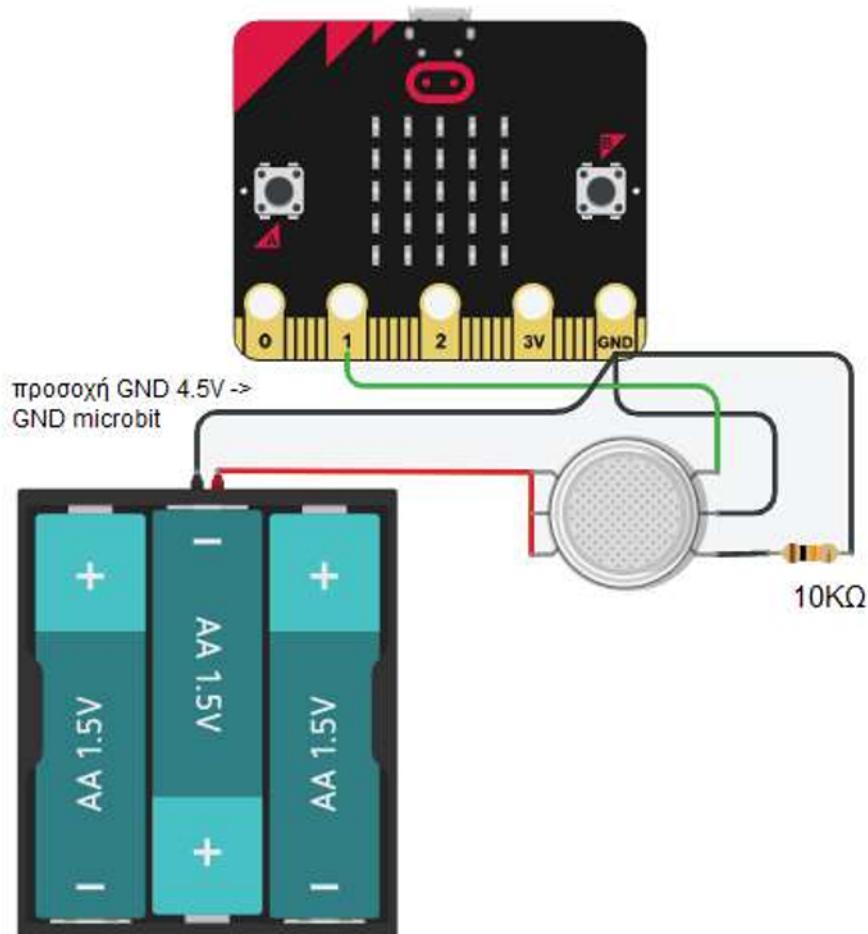
Χωματερή – Μεθάνιο

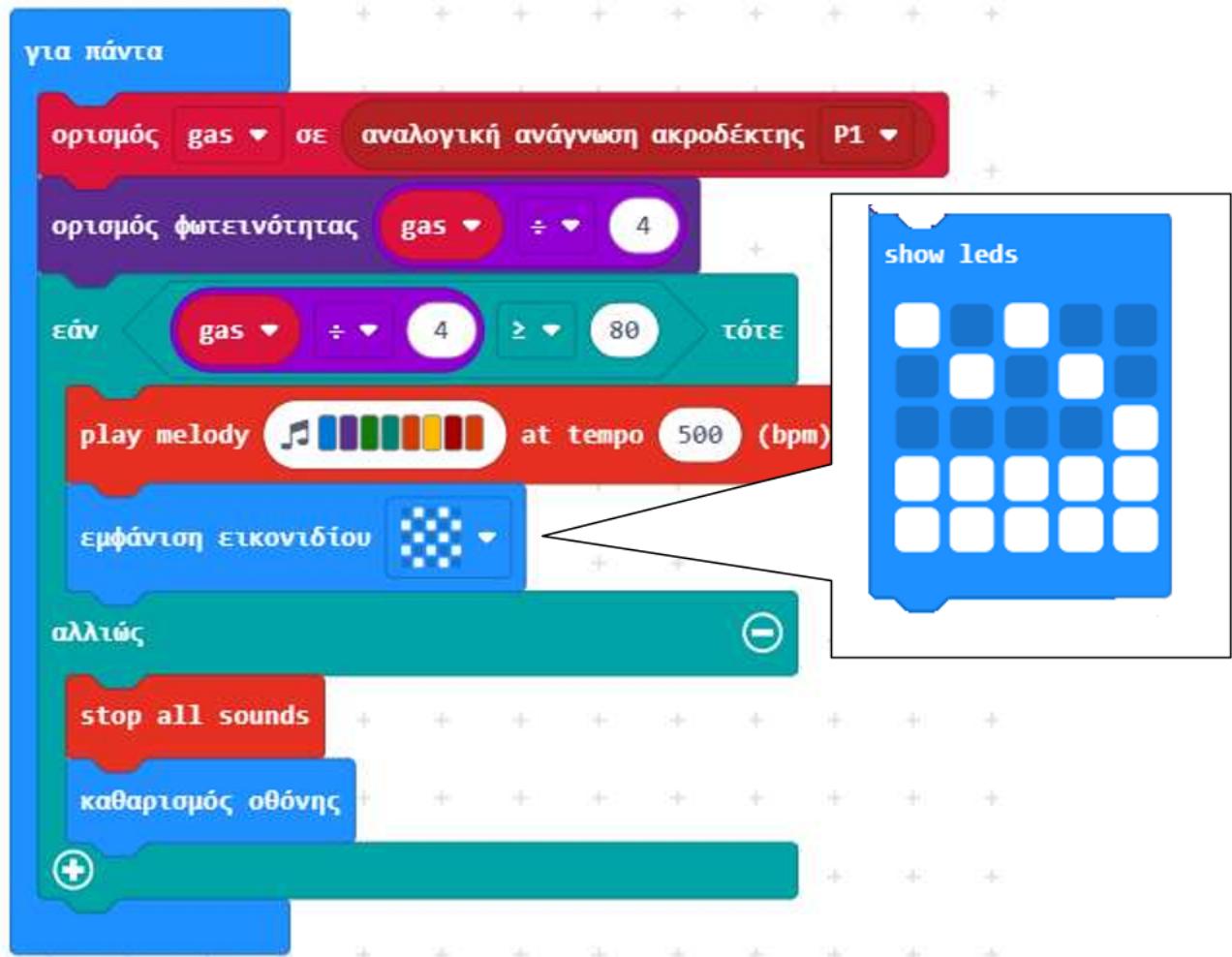
Σύνδεση αισθητήρα μεθανίου



Πραγματική παραγωγή ενέργειας από μεθάνιο

<https://archive.epa.gov/climatechange/kids/solutions/technologies/methane.html>





P1: τιμές από 0 έως 1023

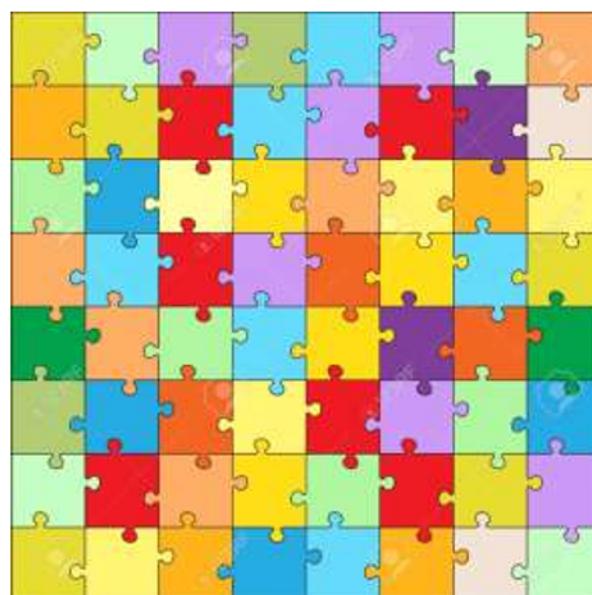
Φωτεινότητα: τιμές από 0 έως 255

Αναλογία 4:1

Εναλλακτικά, απεικόνιση με plot bar graph



Κατεύθυνση 2: Ανταλλακτική Οικονομία & Παιχνίδια



Ανταλλακτική Οικονομία

Ότι περισσεύει σε κάποιον, μπορεί να είναι χρήσιμο για κάποιον άλλο. Προσοχή στους επιτήδειους!

Και βέβαια, παν μέτρον άριστον!



Συναρτήσεις
αποστολής (send)

Περιμένει
περισσότερο (2'') για
να το δουν οι
υποψήφιοι λήπτες

κατά την έναρξη
ράδιο ορισμός 1 ομάδας

Ορισμός καναλιού

“Όταν κουνάω, θέλω να δώσω

επίσημη κούνημα
ορισμός thing * σε τυχαία επιλογή 1 έως 3
εάν thing = 1 τότε
κλήση sendTshirt
εάν thing = 2 τότε
κλήση sendScooter
εάν thing = 3 τότε
κλήση sendDuck
ράδιο αποστολή αριθμού thing

Τυχαίο αγαθό (thing)
1:μπλουζάκι,
2:πατίνι, 3:παπάκι

Συναρτήσεις λήψης
(receive)

συνάρτηση sendTshirt
ερμάνιση εικονιδίου
show leds
παύση (ms) 2000
καθαρισμός οθόνης

συνάρτηση sendScooter
ερμάνιση εικονιδίου
show leds
παύση (ms) 2000
καθαρισμός οθόνης

συνάρτηση sendDuck
ερμάνιση εικονιδίου
show leds
παύση (ms) 2000
καθαρισμός οθόνης

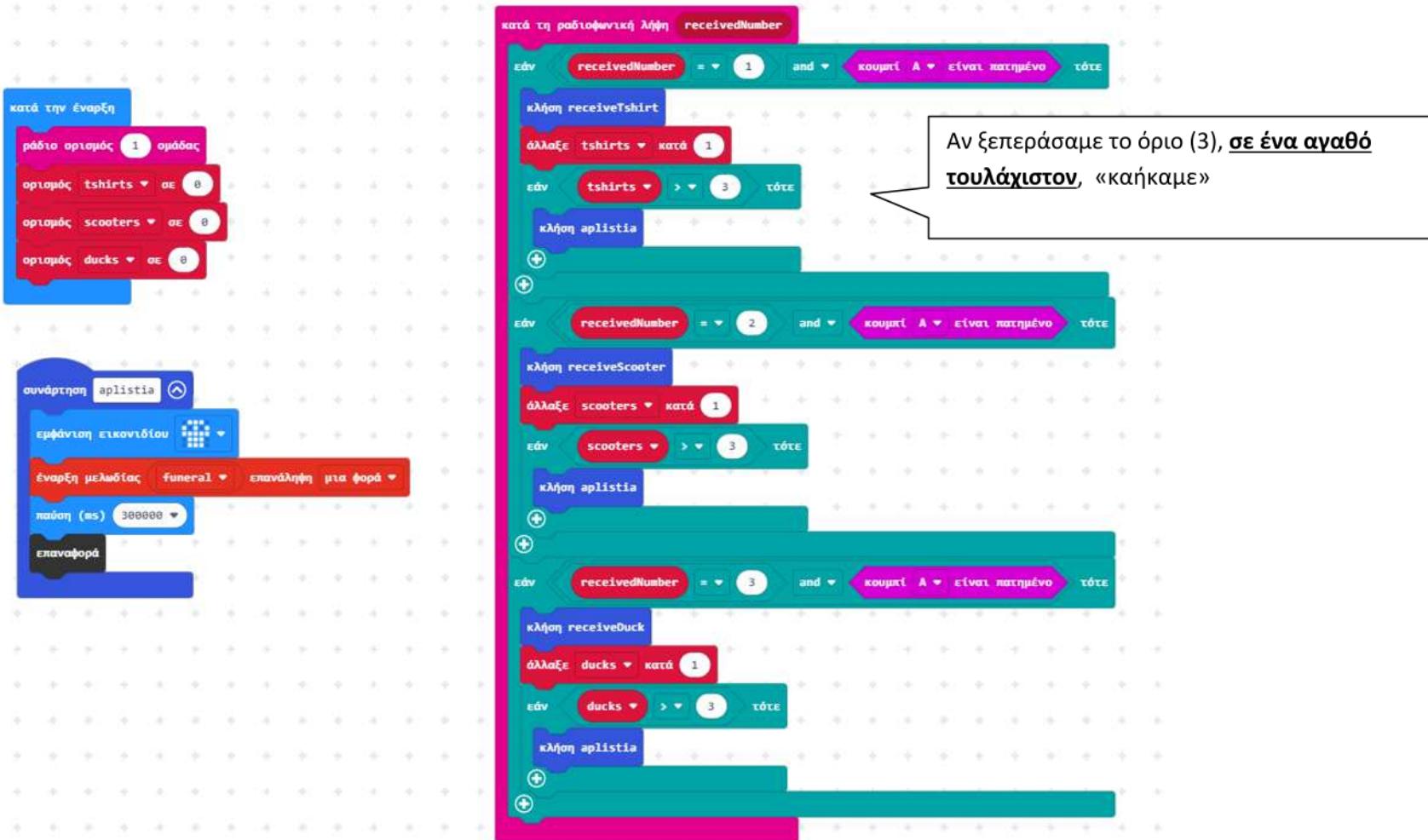
συνάρτηση receiveTshirt
ερμάνιση εικονιδίου
ερμάνιση εικονιδίου
παύση (ms) 500
καθαρισμός οθόνης

συνάρτηση receiveScooter
ερμάνιση εικονιδίου
ερμάνιση εικονιδίου
παύση (ms) 500
καθαρισμός οθόνης

συνάρτηση receiveDuck
ερμάνιση εικονιδίου
ερμάνιση εικονιδίου
παύση (ms) 500
καθαρισμός οθόνης

Αν θέλω το αγαθό,
πρέπει να πατήσω το
κουμπί A μέσα σε 2''

κατά τη ραδιοφωνική λήψη receivedNumber
εάν receivedNumber = 1 and κουμπί A είναι πατημένο τότε
κλήση receiveTshirt
εάν receivedNumber = 2 and κουμπί A είναι πατημένο τότε
κλήση receiveScooter
εάν receivedNumber = 3 and κουμπί A είναι πατημένο τότε
κλήση receiveDuck

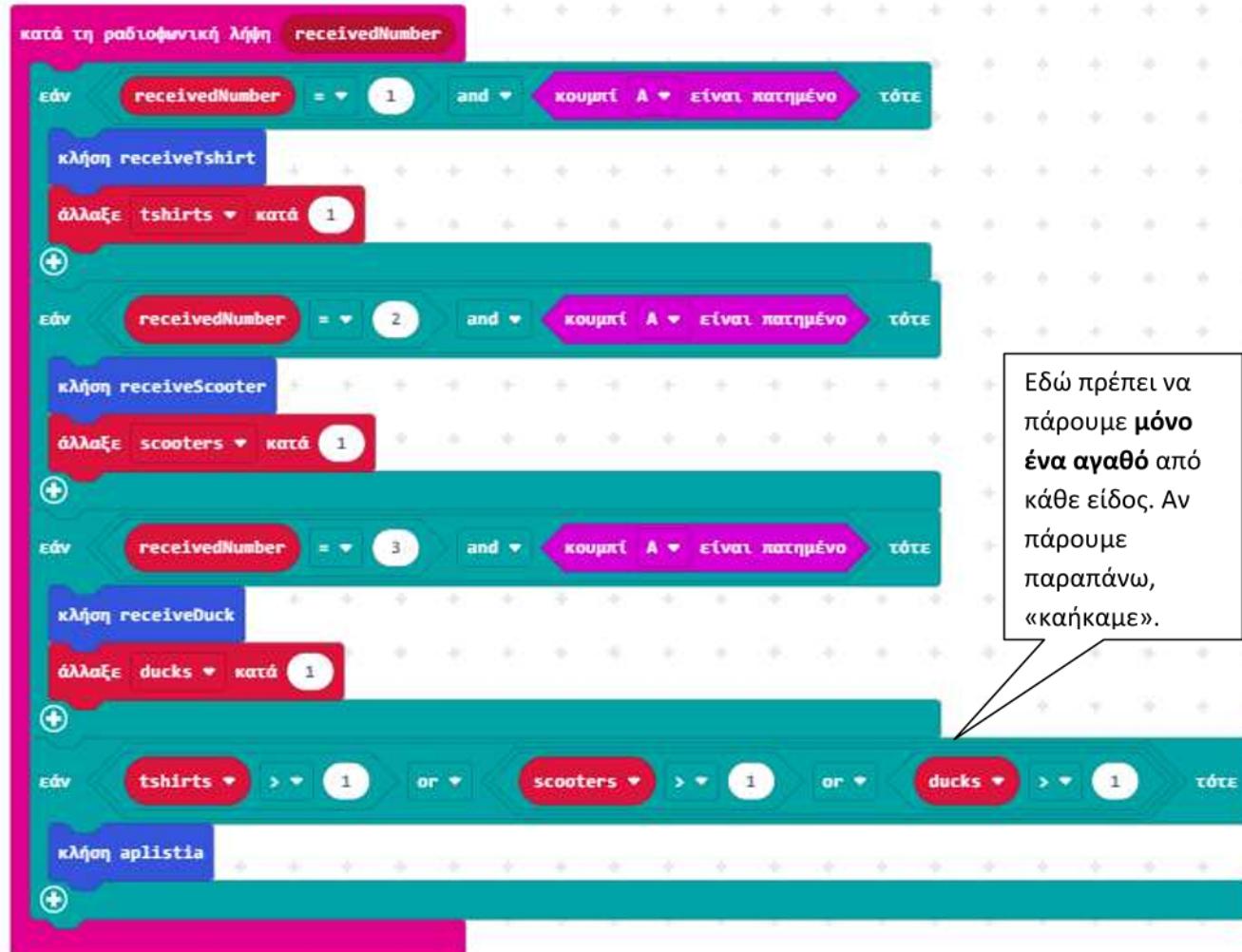


Επέκταση άσκησης: Να υπάρχουν **μετρητές** παραλαβής αγαθών. Αν ξεπεράσουμε κάποιο όριο για κάθε αγαθό, τότε να υπάρχει «**τιμωρία απληστίας**». Πχ να βγάζει μια νεκροκεφαλή να σε αποκλείει για 5'. Οι **μετρητές** πρέπει να **μηδενιστούν στην έναρξη**.

Μηδενισμός
μετρητών

Συνάρτηση
«απλησία»

5min=
300.000
milliseconds



Εδώ πρέπει να
πάρουμε μόνο
ένα αγαθό από
κάθε είδος. Αν
πάρουμε
παραπάνω,
«καήκαμε».

Με tshirts, scooters ducks ψευδές και αληθές

If tshirts == true **and** scooters == true **and** ducks==true

Call ***function full*** (other function, not punishing)



Οικονομία στο ρεύμα

Έξυπνη διαχείριση διακόπτη φωτός



Αναστασία Κωνσταντέλου

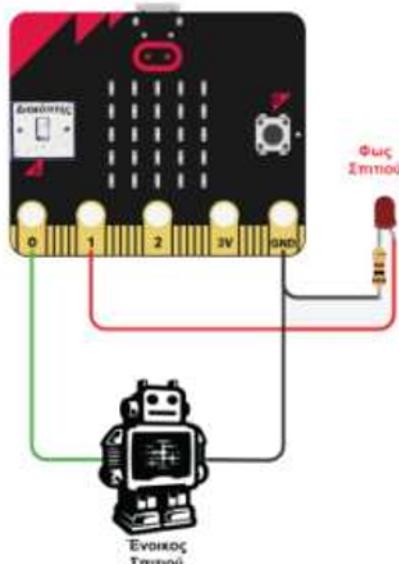


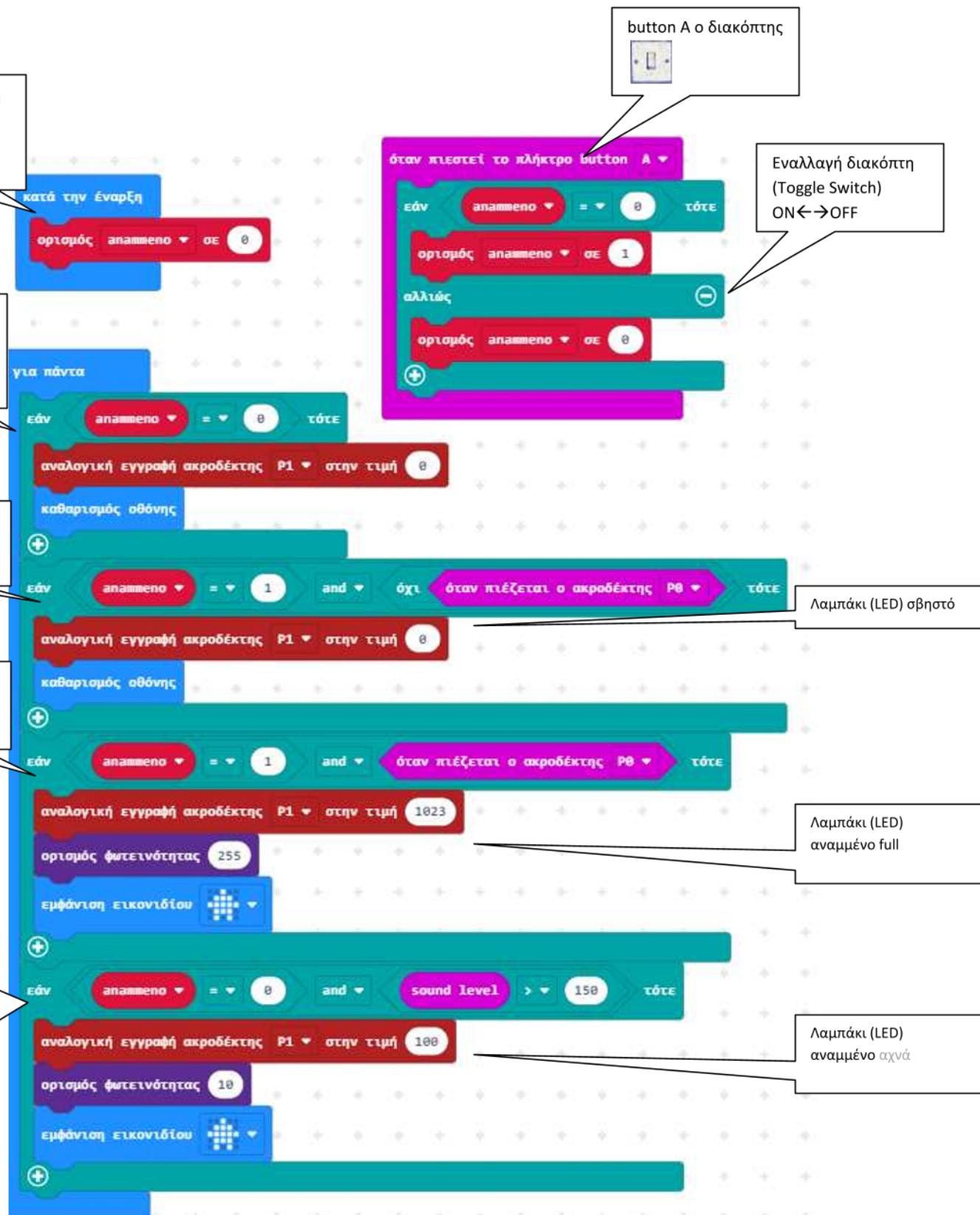
Οικονομία στο φως

Σε ένα Σπίτι – Γεωδαιτικό Θόλο, ζει ένα Ρομπότ



- Ανάβει **και** σβήνει το φως με το button A ενός micro:bit
- Το Ρομπότ είναι τυλιγμένο σε αλουμινόχαρτο και συνδεδεμένο με τον ακροδέκτη **P0** του micro:bit
- Η λάμπα (LED) που φωτίζει το Σπίτι συνδεδεμένη στον ακροδέκτη **P1** του micro:bit.
- Δάπεδο σπιτιού από αλουμινόχαρτο συνδεδεμένο με GND.





Λογικού Τύπου Μεταβλητές, Λογικές Πύλες και Συνθήκες

Η **Μεταβλητή** αναμmeno είναι μια μεταβλητή **λογικού τύπου**. Παίρνει τιμές 0 και 1 ή αλλιώς **Ψευδής / Αληθής**.

anammeno=0 (ψευδής) σημαίνει ότι **δεν** είναι αναμμένο.

anammeno =1 (αληθής) σημαίνει ότι είναι αναμμένο.

Η **Λογική Πύλη ΚΑΙ (and)** παίρνει σαν είσοδο δύο ή περισσότερες Λογικές Μεταβλητές. Βγαίνει Αληθής μόνο όταν όλες οι είσοδοι είναι Αληθείς.

Η **Λογική Πύλη Ή (or)** παίρνει σαν είσοδο δύο ή περισσότερες Λογικές Μεταβλητές. Βγαίνει Αληθής όταν μία τουλάχιστον είσοδος είναι Αληθής.

Η **Λογική Πύλη ΟΧΙ (NOT)** παίρνει μία είσοδο και βγάζει μία έξοδο. Αληθές γίνεται Ψευδές και το αντίστροφό. Το ΟΧΙ(όταν πιέζεται ο ακροδέκτης) εδώ σημαίνει ότι δεν πιέζεται ο ακροδέκτης.

Λογική Συνθήκη είναι μια συνθήκη με Μεταβλητές λογικού τύπου, Λογικές Πύλες καθώς και με **τελεστές σύγκρισης** όπως:

> (μεγαλύτερο)

< (μικρότερο)

≤ (μικρότερο ή ίσο)

≥ (μεγαλύτερο ή ίσο),

Το = που συγκρίνει αν δύο στοιχεία είναι ίσα. (Μην μπερδεύετε με το ίσον της καταχώρησης).

Η Λογική Συνθήκη θα ισχύει ή δεν θα ισχύει (Αληθής ή Ψευδής).

<Συνθήκη>

Παιχνίδι: Πόσο γρήγορα αντιδράς;

Based on: <https://fun-science.org.uk/wp-content/uploads/2020/05/Microbit-projects-independent-book.pdf> activity 11

Στο παιχνίδι αυτό θα φτιάξουμε ένα αντικείμενο (sprite) που έχει τη μορφή ενός led. Το sprite ξεκινά από κέντρο της οθόνης **(2,2)** και κινείται δεξιά-αριστερά.

Προσπαθούμε να πιάσουμε το sprite με το **κουμπί A**, όταν βρίσκεται στο κέντρο της οθόνης (2,2).

Αν το πιάσουμε να εμφανίζεται μια **καρδούλα**.

Αν δεν το πιάσουμε, να εμφανίζεται μια **νεκροκεφαλή**.

Θα χρειαστούμε τα μπλοκ εντολών:

 **Βασικά**  **Είσοδος**

Αλλά κυρίως:

 **Μεταβλητές**  **Παιχνίδι**  **Λογική**

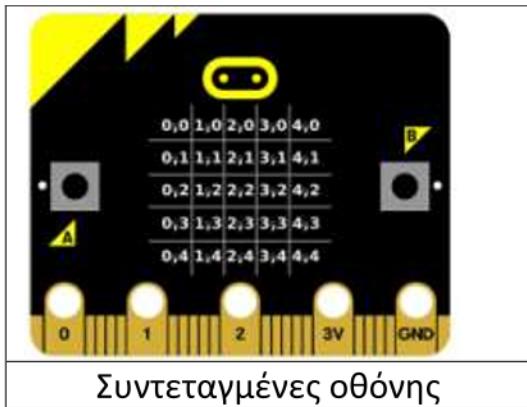
Που «ανεβάζουν» τις δυνατότητες του παιχνιδιού.

Εκτέλεση:

Δημιουργούμε πρώτα το αντικείμενο (sprite) στις



 **Μεταβλητές**



Στο **Παιχνίδι** βάζουμε την εντολή δημιουργία σπράιτ σε 2,2 (στο κέντρο της οθόνης)

δημιουργία σπράιτ σε x: 2 y: 2

εμφανίζεται ένα Led στο κέντρο της οθόνης.

Στο **Κατά την έναρξη** εισάγουμε:

Μεταβλητές **Παιχνίδι**

ορισμός sprite ▾ σε δημιουργία σπράιτ σε x: 2 y: 2

Στο **Για πάντα** εισάγουμε:

Μεταβλητές **Παιχνίδι**

sprite ▾ μετακίνηση κατά 1
sprite ▾ αν στην άκρη, αναπήδηση

Έτσι, το sprite μετακινείται δεξιά-αριστερά στην οθόνη.

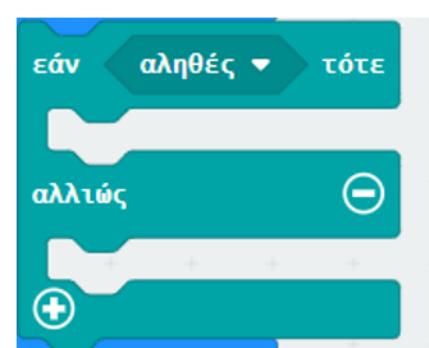
Πώς θα “πιάσουμε” το sprite;

⦿ Είσοδος

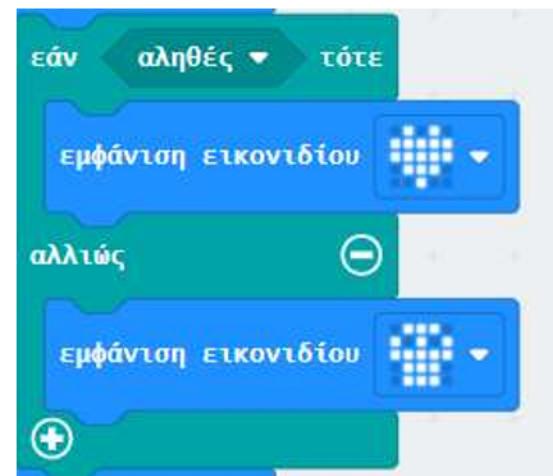
Με την εντολή από

Μέσα στο: **όταν πιεστεί το πλήκτρο button A**

☒ Λογική

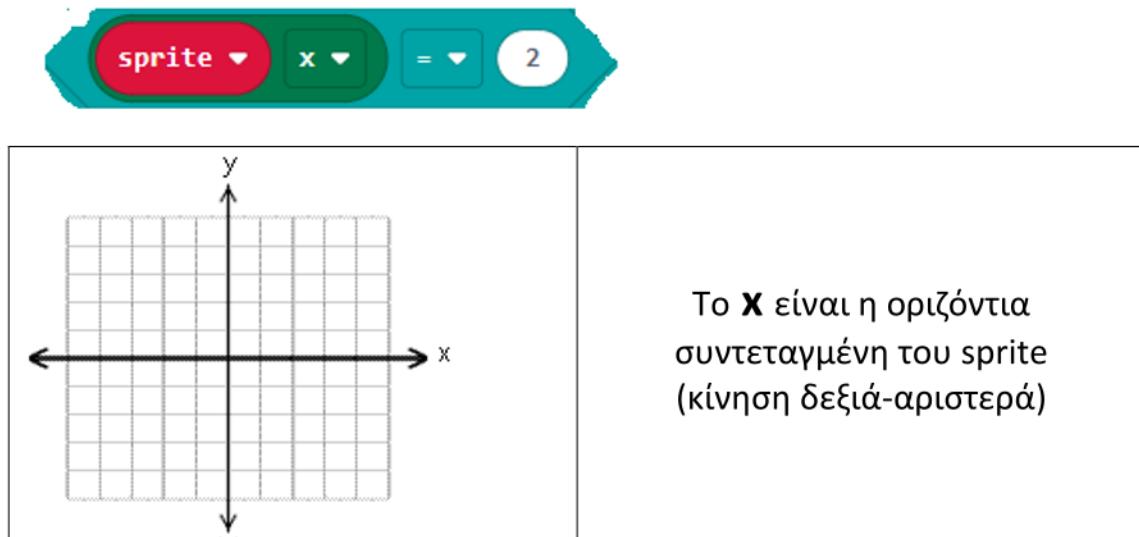


Αν το πιάσουμε, θα βγάζει καρδούλα, αν δεν το πιάσουμε θα βγάζει νεκροκεφαλή.



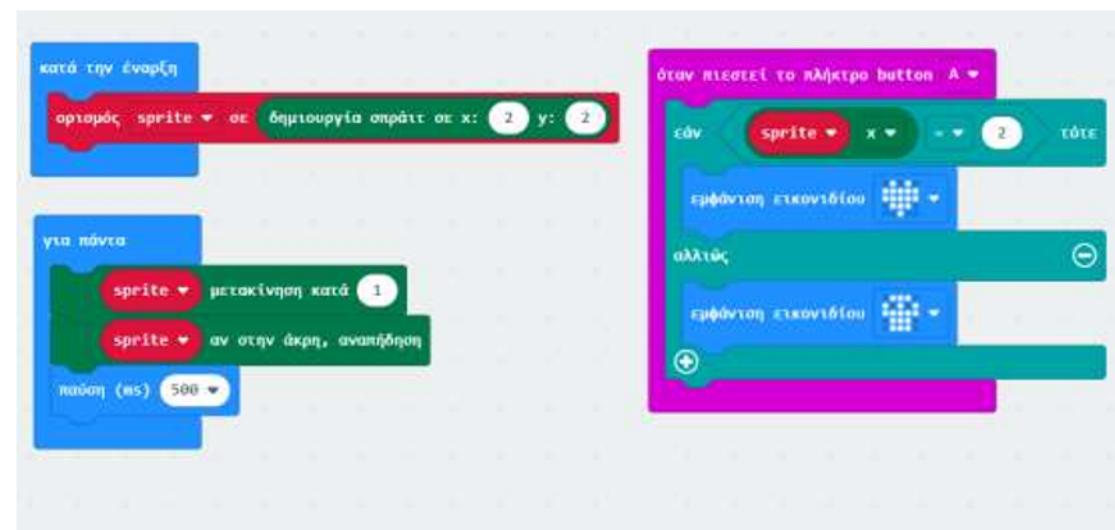
Το «Αν το πιάσουμε» είναι μια **συνθήκη** που δεν την έχουμε βάλει ακόμα στο πρόγραμμα.

Από τη **Λογική** **Μεταβλητές** **Παιχνίδι** φτιάχνουμε τη συνθήκη:



Θα χρειαστεί να βάλετε και μια παύση (πόσο;) στο για πάντα, για να μπορείτε να πιάσετε το sprite.

Δείτε ολοκληρωμένο το πρόγραμμα:



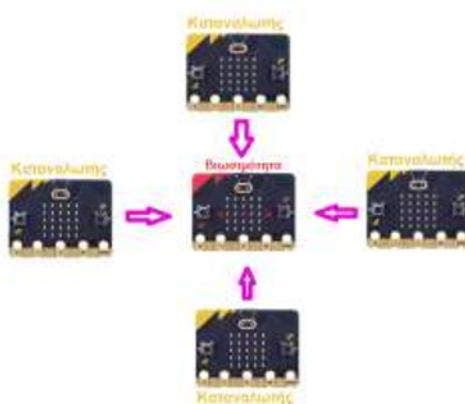
Επαπειλούμενη Βιωσιμότητα

Επαπειλούμενη Βιωσιμότητα

Endangered Sustainability

Το έργο αυτό συνδυάζει το παιχνίδι «πόσο γρήγορα αντιδράς» και την αλληλεπίδραση με άλλα micro:bit.

Έχουμε ένα micro:bit που αντιπροσωπεύει τη **βιωσιμότητα**, και γύρω του micro:bit **καταναλωτές**.



Το micro:bit **βιωσιμότητα** παίζει το παιχνίδι «πόσο γρήγορα αντιδράς», ξεκινώντας με μια αρχική «προίκα» χρόνου καθυστέρησης, ώστε να μπορεί κανείς να παίζει άνετα το παιχνίδι.

Τα micro:bit **καταναλωτές**, όσο πατάμε τα buttons A ή B ή A+B, **αφαιρούν** από την «προίκα» της βιωσιμότητας, λίγο, περισσότερο ή ακόμα περισσότερο, στέλνοντας (**radio**) αριθμούς στο micro:bit **βιωσιμότητας**. «Ετσι, αν υπάρχει κατάχρηση από την μεριά των **καταναλωτών**, η **βιωσιμότητα** δυσκολεύεται όλο και περισσότερο. Αν ο χρόνος καθυστέρησης – η «προίκα» μικρύνει πάρα πολύ, δεν μπορούμε να έχουμε βιωσιμότητα

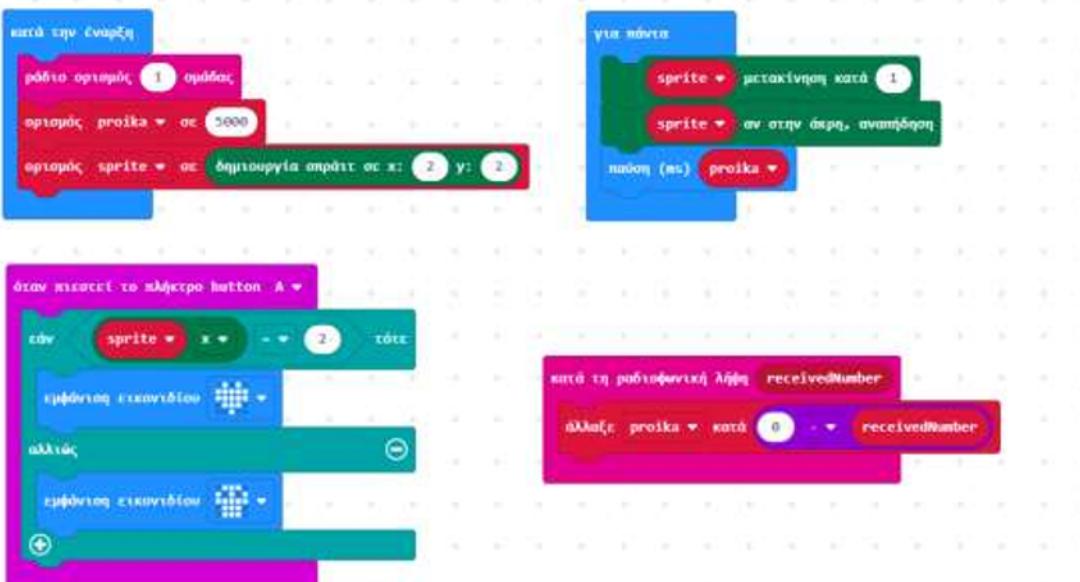
Ο κώδικας για τη **βιωσιμότητα**

(Το παιχνίδι «πόσο γρήγορα αντιδράς*» με εντολές **radio** και μεταβλητή «προίκα»).

* Το παιχνίδι «πόσο γρήγορα αντιδράς»: Ένα led πηγανοέρχεται στη μεσαία γραμμή.

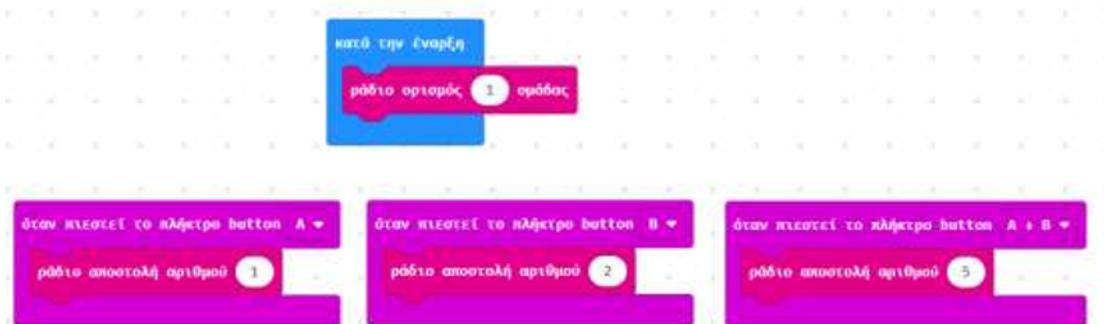
Πρέπει να το «χτυπήσεις» όταν είναι ακριβώς στο κέντρο. Χρησιμοποιείς το button A για να το «χτυπήσεις». Αν το «χτυπήσεις» κερδίζεις, αλλιώς χάνεις. Χρησιμοποιούμε εντολές από

το  **Παιχνίδι**



endangered sustainability game

Ο κώδικας για τους **καταναλωτές**

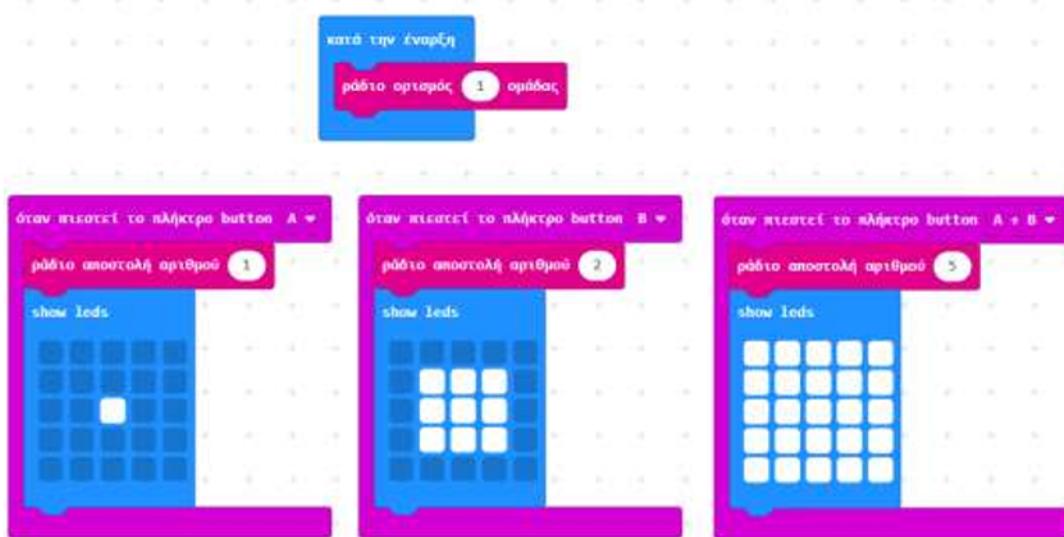


endangered sustainability consumer

Με λίγη περισσότερη αλληλεπίδραση

Καταναλωτής, όσο πιο πολύ ζητώ, να φαίνεται στα leds

(προσθήκη κώδικα στον καταναλωτή)



Προσθήκη στη **βιωσιμότητα** στέλνει μήνυμα στους καταναλωτές να

προσέχεις και τέλος ότι πεθαίνει

(προσθήκη στη βιωσιμότητα)

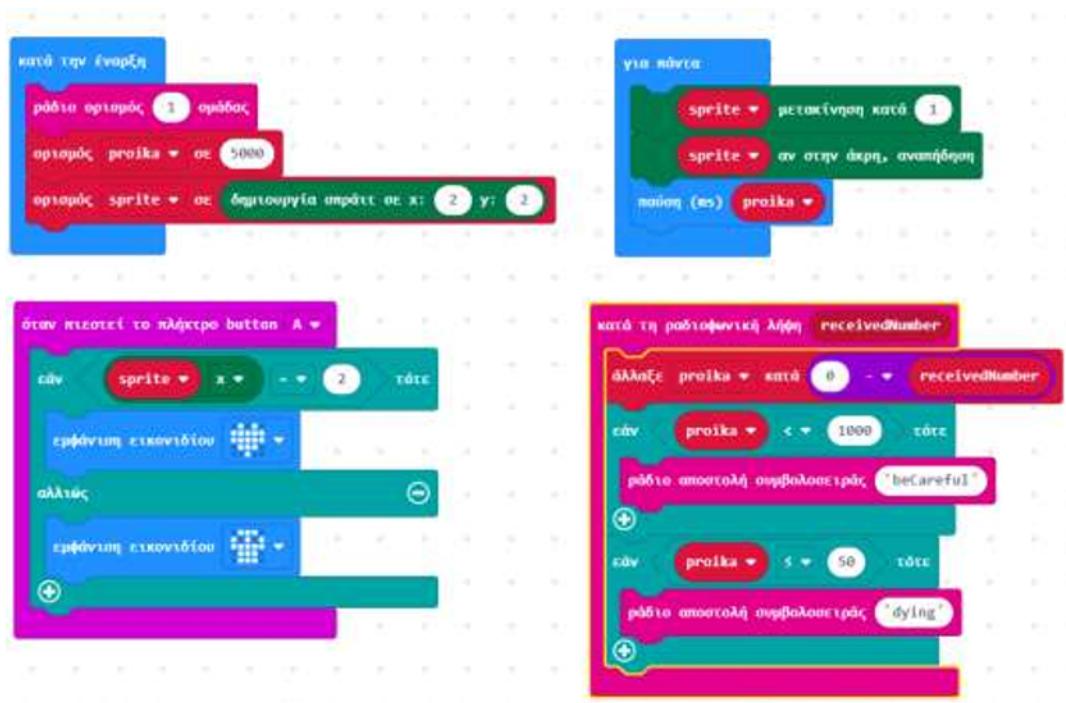


Μπορούμε επίσης να βάλουμε και ηχητικά εφέ.

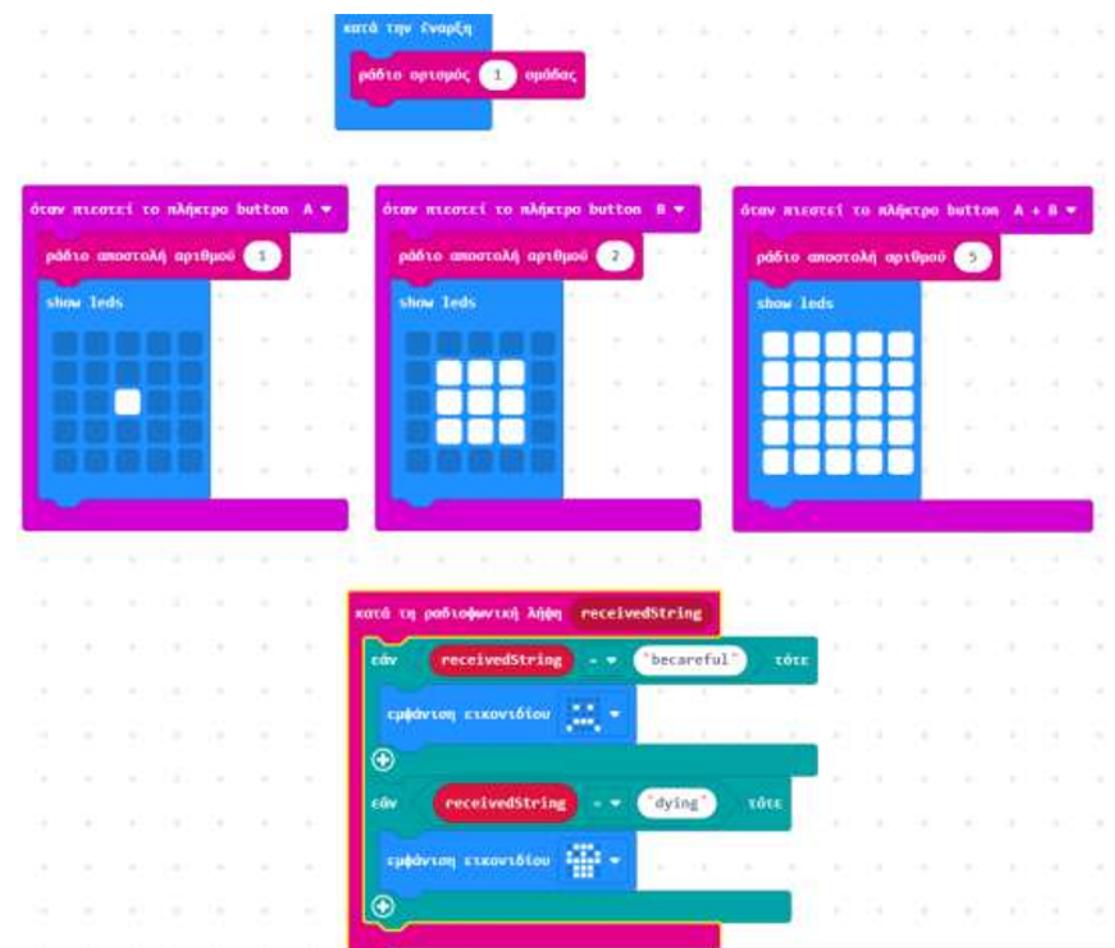
Όταν λάβει ο **Καταναλωτής**, εμφανίζει μήνυμα προειδοποίησης
(προσθήκη κώδικα στον καταναλωτή)



Ολοκληρωμένος κώδικας για βιωσιμότητα



Ολοκληρωμένος κώδικας για καταναλωτή

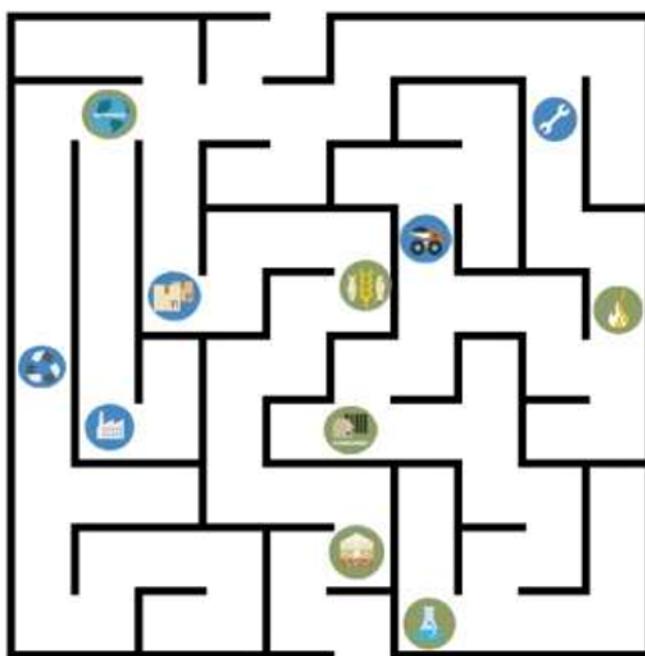


Λαβύρινθος (Maze Run)

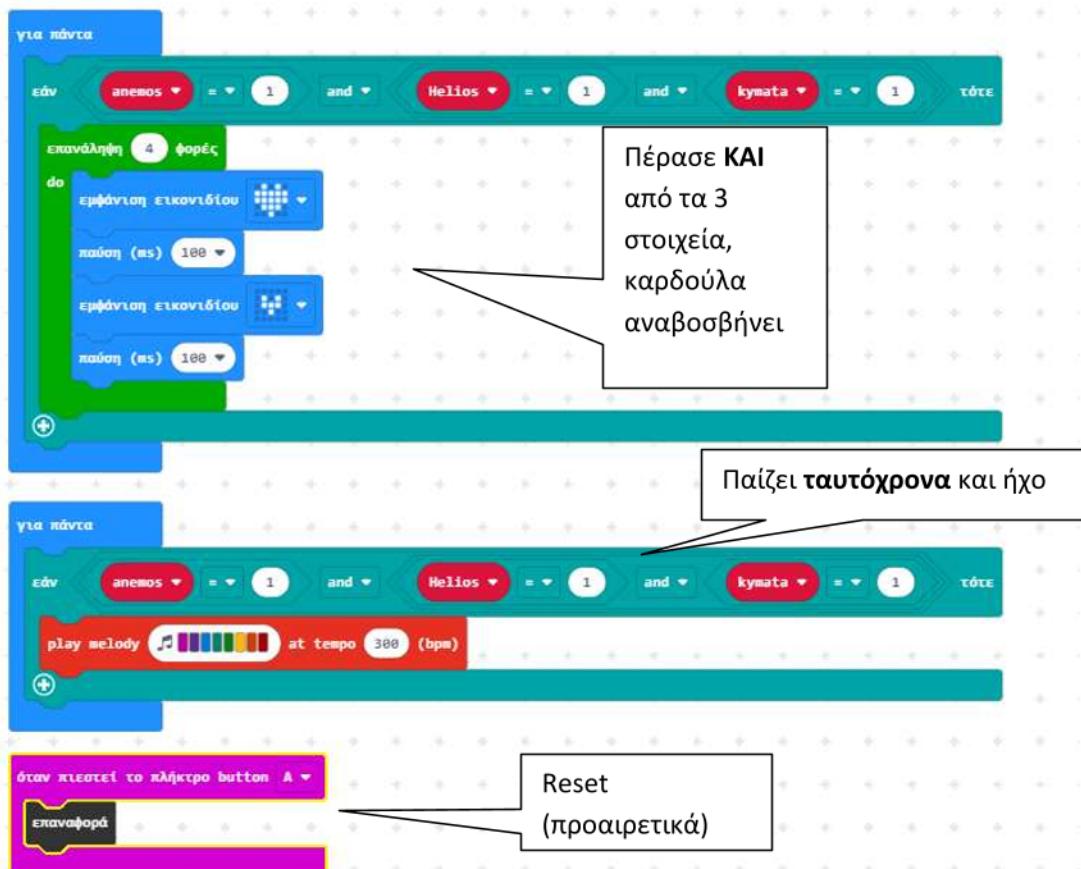
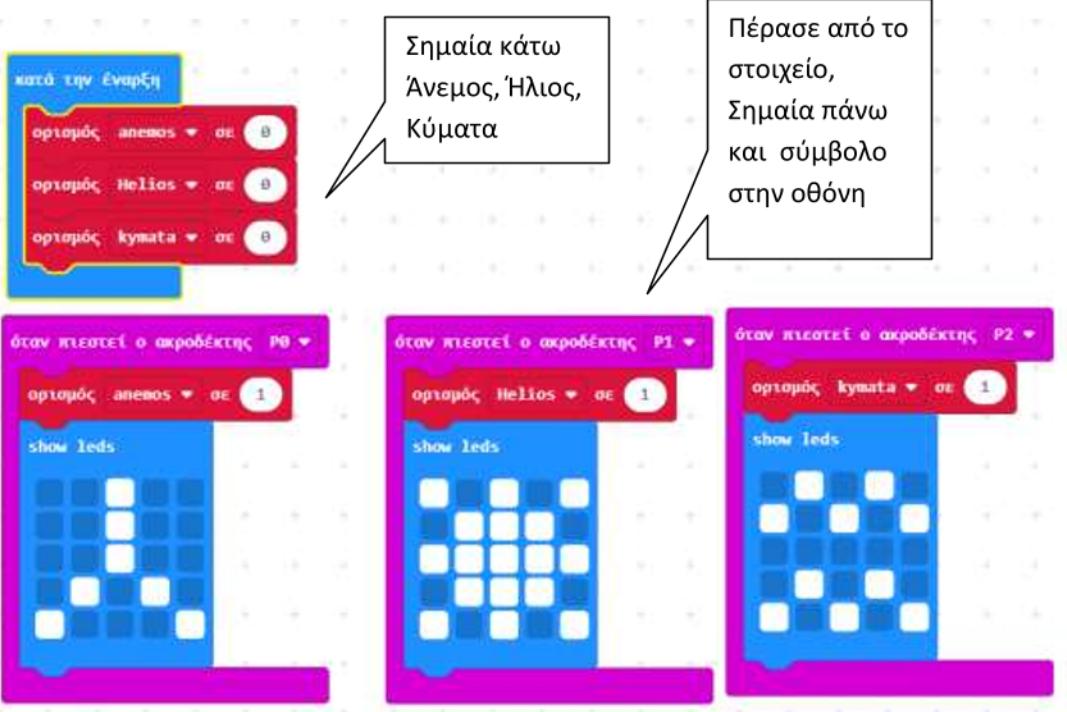
Θα χρειαστούμε επιπλέον:

- Πινέζες (γυμνές, χωρίς πλαστικό από επάνω) ή καρφάκια με κεφάλι
- Κουτί συσκευασίας από χαρτόνι
- Αλουμινόχαρτο
- Ταινία Χαλκού (προαιρετικά)

Ο Λαβύρινθος φτιάχτηκε με το Maze Generator: <https://www.mazegenerator.net>

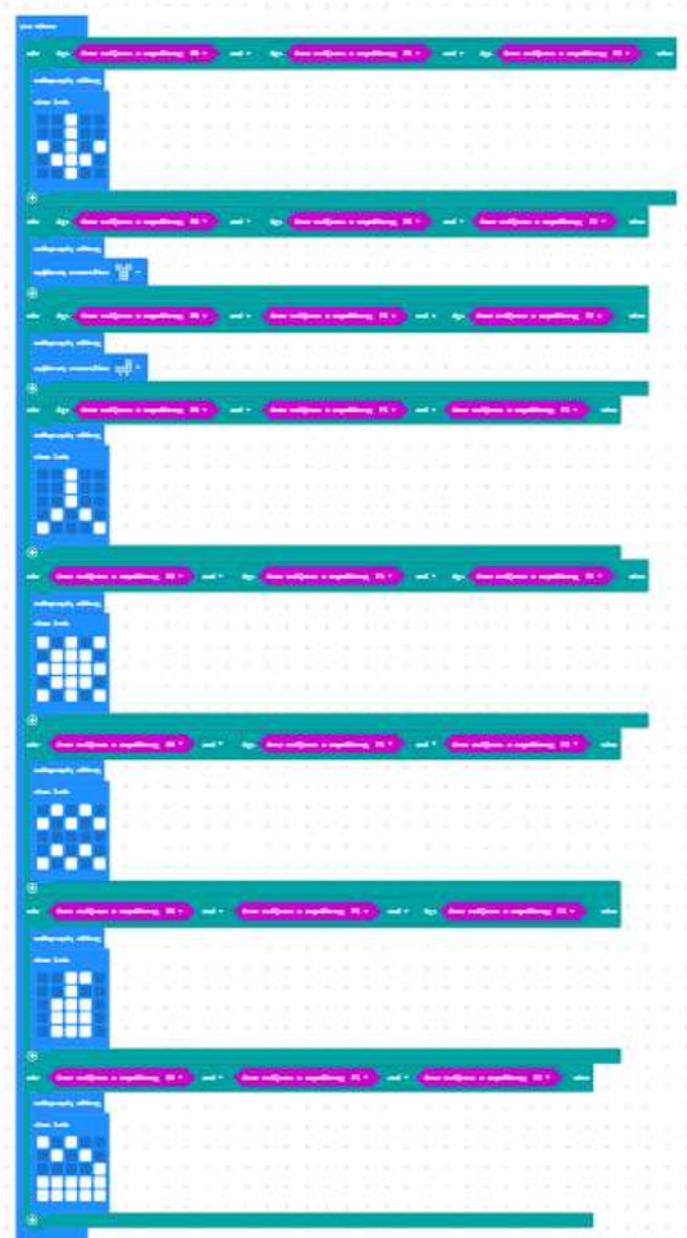


Πάνω από το κουτί συσκευασίας θα τοποθετήσουμε τον λαβύρινθο (μπορούμε να τον πλαστικοποιήσουμε). Σε όλες εικόνες που φαίνονται θα βάλουμε τις πινέζες και θα τρυπήσουμε το χαρτόνι. Προαιρετικά μπορούμε να καλύψουμε τις πινέζες με ταινία χαλκού. Στις τρεις πινέζες που υπάρχουν στο **σωστό δρόμο** θα βάλουμε κροκοδειλάκια (από το κάτω μέρος) και θα συνδέσουμε με τα P0, P1 και P2. P0 → Άνεμος, P1 → Ήλιος, P2 → Κύματα. Στη Γείωση(GND) βάζουμε ένα κροκοδειλάκι με ένα ρομποτάκι τυλιγμένο σε αλουμινόχαρτο. Με το ρομποτάκι διατρέχουμε τον λαβύρινθο. Όταν περνάμε από τον σωστό δρόμο και αγγίζουμε τα στοιχεία (Άνεμο, Ήλιο, Κύματα), τα leds του micro:bit εμφανίζουν το αντίστοιχο εικονίδιο. Αν συμπληρώσουμε και τα τρία στοιχεία, βγαίνει μια καρδούλα που αναβοσβήνει και ήχο.



Τα Τρία Δυαδικά Ρομποτάκια

Τα Τρία Δυαδικά Ρομποτάκια



Δείγμα κώδικα για 0,0,0:



Σπίτι Γεωδαιτικός Θόλος που περιμένει να του βάλουμε μέσα Ρομποτάκια. Στη σκεπή του σπιτιού το micro:bit, μας προτρέπει να τα βάλουμε.

Δάπεδο σπιτιού από αλουμινόχαρτο συνδεδεμένο με GND.

Ρομποτάκια τυλιγμένα με αλουμινόχαρτο χρωματισμένο συνδεδεμένα με **P0**, **P1**, **P2** και με αντίστοιχα χρώματα στα κροκοδειλάκια.

0	0	0			Insert Robot (Προτροπή)
0	0	1			Μπλουζάκι
0	1	0			Πατίνι
0	1	1			Ανεμογεννήτρια Wind Turbine
1	0	0			Ήλιος Solar Energy
1	0	1			Κύματα Tidal Energy
1	1	0			Φρούτο Bioenergy
1	1	1			Χωματερή - Μεθάνιο

Εξοικείωση με: δυαδικό σύστημα αρίθμησης- λογική πράξη AND - λογική πράξη NOT.

Εμφανίζουμε τα εικονίδια που δημιουργήσαμε στο έργο.

Σημείωση: ενδεχομένως να αλλάξουμε τα χρώματα των ρομπότ, ανάλογα με τα κροκοδειλάκια που έχουμε.

Προσοχή!

This function works best when the micro:bit is using batteries for power, instead of the USB cable.