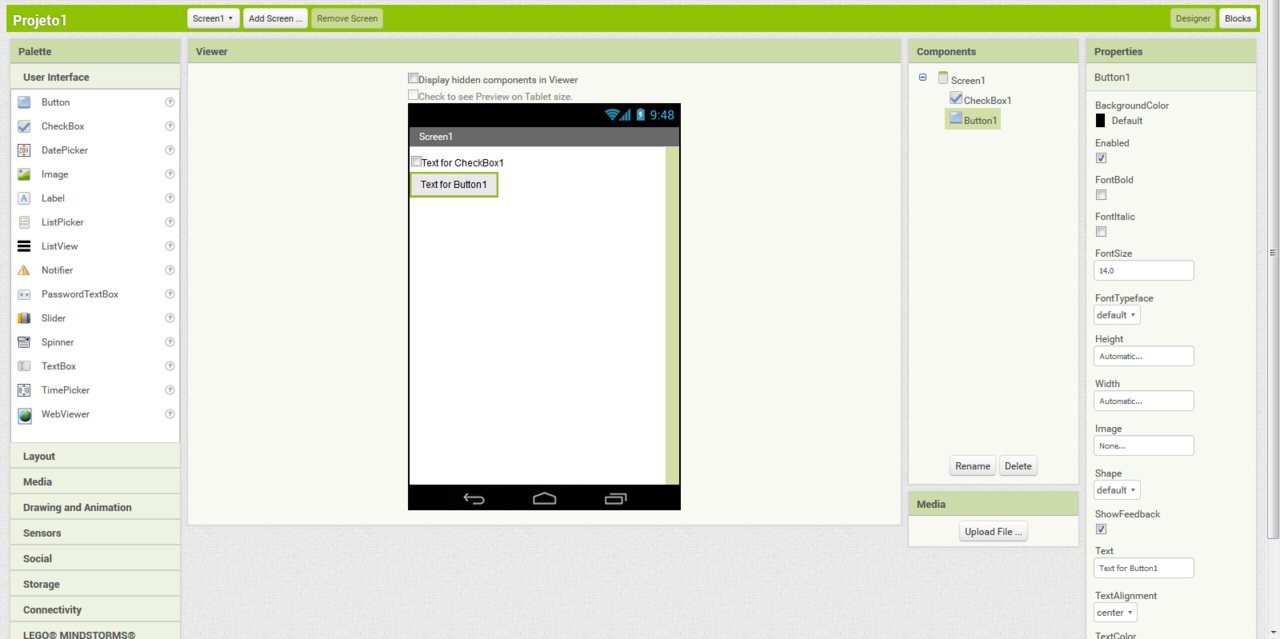
**ECO FARM BOT**

**ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΗΣ**

**1ο Φύλλο Έργου – Διεπαφή Χρήστη (User Interface)**

Η εφαρμογή αναπτύχθηκε στην πλατφόρμα **MIT App Inventor 2** (<https://appinventor.mit.edu/>). Πρόκειται για ένα ολοκληρωμένο περιβάλλον ανάπτυξης εφαρμογών για συσκευές Android. Απαραίτητη προϋπόθεση για τη χρήση του περιβάλλοντος είναι η ύπαρξη λογαριασμού e-mail στη Google.



Η εφαρμογή θα πρέπει να υλοποιηθεί σε δύο φάσεις. Στην πρώτη φάση θα υλοποιηθεί η διεπαφή χρήστη (User Interface), το πώς δηλαδή θα παρουσιάζεται αυτή στο χρήστη. Ενώ στη δεύτερη θα επιλεγούν τα κατάλληλα block κώδικα που θα δώσουν την επιθυμητή λειτουργικότητα στην εφαρμογή.

**Διεπαφή Χρήστη (User Interface**)

Η αρχική οθόνη που θα βλέπει ο χρήστης θα πρέπει να έχει την εξής μορφή:



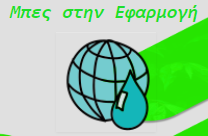
Μέσω αυτής της οθόνης ο χρήστης θα μπορεί να δει πληροφορίες του έργου επιλέγοντας το εικονίδιο:



Καθώς και να δει ποιοι συμμετείχαν για την υλοποίηση του έργου με την επιλογή του εικονιδίου:



Επιλέγοντας ο χρήστης το κεντρικό εικονίδιο,



αποκτά πρόσβαση στη βασική οθόνη της εφαρμογής.



Πρόκειται για εφαρμογή εποπτείας (monitoring - ακολουθώντας τις απαιτήσεις που τέθηκαν). Ο χρήστης μπορεί να εποπτεύει την υγρασία της εκάστοτε γλάστρας τόσο σε αριθμητική μορφή όσο και σε γραφικό (εικόνα γλάστρας). Η εικόνα της κάθε γλάστρας αλλάζει ανάλογα με την τιμή της υγρασίας της. Το όρια τέθηκαν σύμφωνα με τις προδιαγραφές που δόθηκαν πριν της ανάπτυξη της εφαρμογής.

Τέλος, ο χρήστης μπορεί επιλέγοντας το εικονίδιο:



Να δει σε γράφημα την πορεία της υγρασίας της γλάστρας στις 30 τελευταίες μετρήσεις που στάλθηκαν από το βραχίονα.



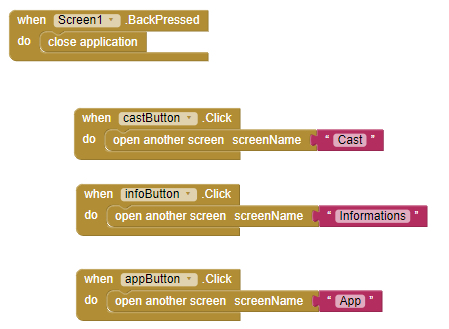
**Ανάπτυξη Εφαρμογής (Μέρος 1ο)**

Το δεύτερο σκέλος της εφαρμογής είναι η λειτουργικότητα της με τα κατάλληλα block προγραμματισμού.

## Αρχική Οθόνη (Screen1)

Για την αρχική οθόνη (Screen1) τα block κώδικα που θα πρέπει να αναπτυχθούν και το τι αυτά κάνουν φαίνεται παρακάτω:

Όταν επιλεχθεί το backPressed (πίσω βελάκι) **κλείνει η εφαρμογή**



Όταν επιλεχθεί το εικονίδιο  **ανοίγει την οθόνη Information**

Όταν επιλεχθεί το εικονίδιο  **ανοίγει την οθόνη App**

Όταν επιλεχθεί το εικονίδιο  **άνοιγει η οθόνη Cast**

## Οθόνη Cast

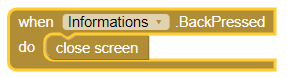
Για την οθόνη **Cast** έχουμε:



Όταν επιλεχθεί το backPressed (πίσω βελάκι) **κλείνει η οθόνη**

## Οθόνη Informations

Για την οθόνη **Informations** έχουμε:

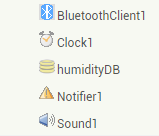


Όταν επιλεχθεί το backPressed (πίσω βελάκι) **κλείνει η οθόνη**

## Οθόνη App

Πρόκειται για την βασική οθόνη της εφαρμογής, εκεί γίνονται όλες οι λειτουργίες της.

Θα πρέπει να σημειωθεί ότι σε αυτήν την οθόνη έχουν εισαχθεί αντικείμενα:



- Bluetooth (Για λήψη δεδομένων)

- Ρολόι (Για διαχείριση συμβάντων)

- Βάση δεδομένων (Για διαχείριση δεδομένων)

- Notifier (Για εμφάνιση μηνυμάτων προς το χρήστη)

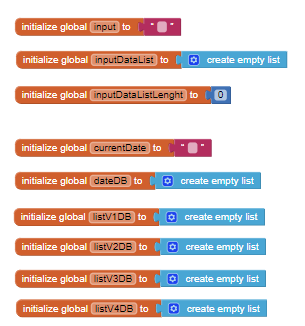
- Sound (Για χειρισμό δόνησης)

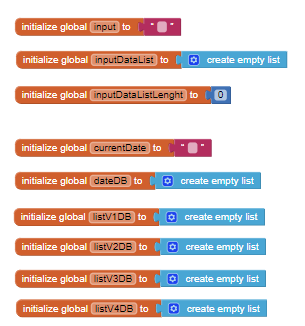
Αρχικά λοιπόν για να μπορέσουμε να διαχειριστούμε τα δεδομένα που θα λαμβάνουμε από τον βραχίονα θα πρέπει να δηλώσουμε κάποιες μεταβλητές και αντίστοιχες λίστες. Έτσι:

- Αρχικοποιούμε μια μεταβλητή **input** η οποία θα δέχεται τη συμβολοσειρά εισόδου από το bluetooth.

- Δηλώνουμε μια άδεια λίστα **inputDataList** στην οποία θα μπαίνουν διακριτά τα δεδομένα εισόδου ώστε να μπορούμε να τα διαχειριστούμε ξεχωριστά.

- Αρχικοποιούμε μια μεταβλητή **inputDataListLenght** στην οποία θα αποθηκεύουμε το μήκος της **inputDataList** ώστε να αποφευχθούν λανθασμένα δεδομένα εισόδου.





- Αρχικοποιούμε μια μεταβλητή **currentDate** στην οποία θα αποθηκεύουμε την ημερομηνία που λαμβάνεται μια μέτρηση υγρασίας.

- Δηλώνουμε μια άδεια λίστα **dateDB** στην οποία θα αποθηκεύονται οι ημερομηνίες που πάρθηκαν μετρήσεις.

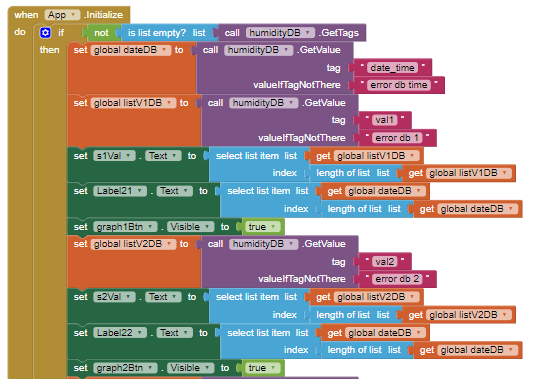
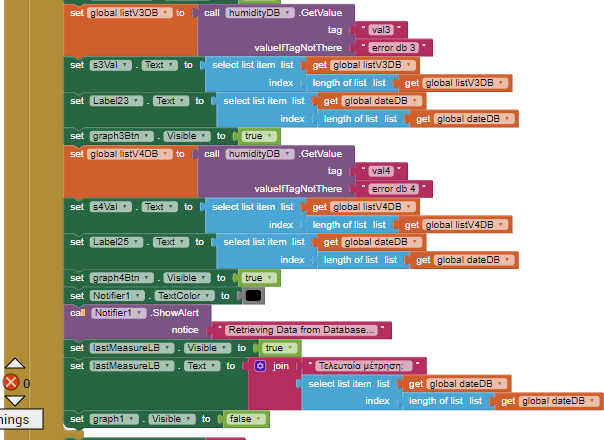
- Δηλώνουμε 4 άδειες λίστες **listV1DB, listV2DB, listV3DB, listV4DB** στις οποίες θα αποθηκεύονται οι τιμές υγρασίας της εκάστοτε γλάστρας.

Όταν θα **αρχικοποιείται η οθόνη App** θα πρέπει να πραγματοποιούνται κάποιοι έλεγχοι δεδομένων ώστε ο χρήστης να έχει μπροστά του μια οθόνη που αντιστοιχεί στην πραγματικότητα. Έτσι λοιπόν κατά την αρχικοποίηση της οθόνης:

- **ελέγχουμε αν η βάση δεδομένων (humidityDB) δεν είναι άδεια**.

-Σε αυτήν την περίπτωση θα πρέπει να «φορτώσουμε» τα δεδομένα της στις αντίστοιχες λίστες που αρχικοποιήσαμε πριν και ήταν άδειες.

-Αφού φορτωθούν τα δεδομένα οι τελευταίες μετρήσεις για την υγρασία των γλαστρών θα πρέπει να εμφανιστούν στα αντίστοιχα textboxes και labels.

* Στην περίπτωση που δεν υπάρχουν δεδομένα στη βάση δεδομένων θα πρέπει να εμφανιστούν αντίστοιχες τιμές στα ανάλογα πεδία.



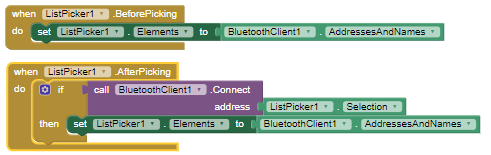
**Ανάπτυξη Εφαρμογής (Μέρος 2ο)**

## Οθόνη App (Συνέχεια…)

Αφού έχουν γίνει οι αρχικοποιήσεις ο χρήστης θα μπορεί **να συνδεθεί μέσω Bluetooth στο βραχίονα** ώστε να λάβει εκ νέου καινούριες μετρήσεις. Αυτό γίνεται επιλέγοντας το αντίστοιχο εικονίδιο



το οποίο ενεργοποιεί τα παρακάτω block κώδικα:



Το πιο βασικό αντικείμενο της οθόνης και γενικότερα της εφαρμογής είναι **το ρολόι και η διαχείριση του**. Έτσι λοιπόν το ρολόι είναι αυτό που πυροδοτεί τα περισσότερα συμβάντα στην εφαρμογή μας. Εκεί ελέγχουμε:

**Α.** Ποια εικόνα θα φαίνεται στο γραφικό κάθε γλάστρας

**Β.** Πότε θα διαβάζουμε δεδομένα από τον βραχίονα μέσω Bluetooth

**Γ.** Πως θα διαχειριστούμε αυτά τα δεδομένα

**Δ.** Ενεργοποίηση/απενεργοποίηση αντικειμένων ώστε η εφαρμογή να είναι φιλική προς το χρήστη.

**Α.** Κάθε γλάστρα ανάλογα με την τιμή της υγρασίας της θα πρέπει να έχει και το αντίστοιχο γραφικό σύμφωνα με τις προδιαγραφές που έχουν τεθεί. Αρχικά λοιπόν θα πρέπει να αρχικοποιήσουμε κάποιες μεταβλητές:

- Αρχικοποιούμε μια μεταβλητή **numOf\_Measures** η οποία θα αποθηκεύει τον αριθμό των μετρήσεων που πάρθηκαν.

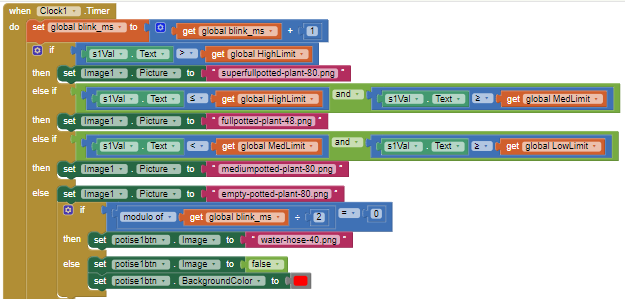
- Δηλώνουμε και αρχικοποιούμε μεταβλητές **HighLimit, MedLimit, LowLimit** που ορίζουν τα όρια της υγρασίας των γλαστρών.

- Τέλος δηλώνουμε μια μεταβλητή **blink\_ms** η οποία θα μας βοηθήσει όταν η υγρασία της γλάστρας είναι πολύ χαμηλή να βλέπει ο γραφικά μια ένδειξη επείγοντος (αναβόσβημα εικονιδίου).

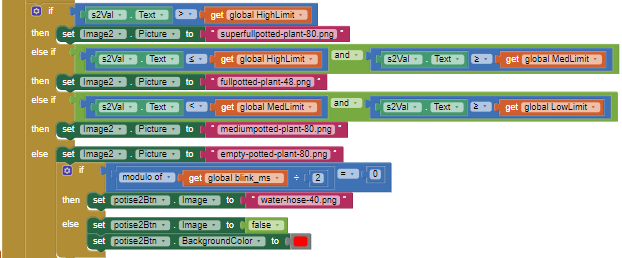


Όταν ενεργοποιείται λοιπόν το ρολόι αρχικά θα πρέπει το γραφικό των γλαστρών να είναι ανάλογο της υγρασίας.

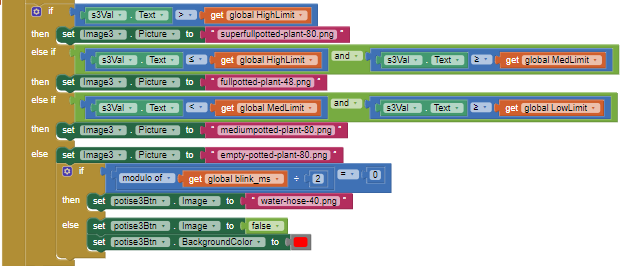
Έχουμε τα αντίστοιχα **block κώδικα για την πρώτη γλάστρα**:



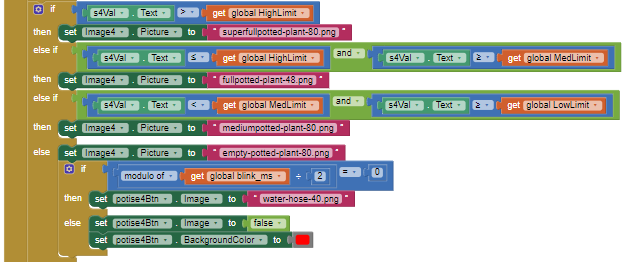
**για την δεύτερη γλάστρα:**

****

**για την τρίτη γλάστρα:**

****

**για την τέταρτη γλάστρα:**

****

Το επόμενο κομμάτι κώδικα είναι και το πιο κρίσιμο. Θα πρέπει:

-να γίνεται **έλεγχος αν υπάρχει σύνδεση με το βραχίονα (1),**

**-αν υπάρχει** πρέπει να τίθενται κατάλληλες τιμές σε αντικείμενα (labels, textboxes …) **(2)**και

- έπειτα **να διαβάζονται τα δεδομένα** που στέλνει ο βραχίονας **(3)**

**- να αποκωδικοποιούνται** (για ποια γλάστρα έγινε η μέτρηση) **(4)**

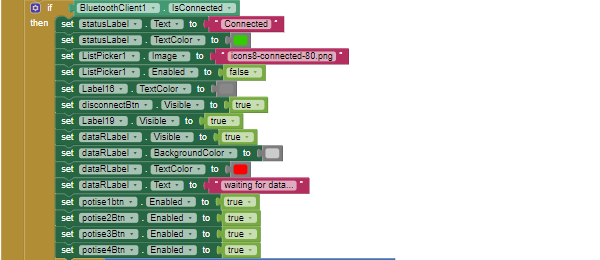
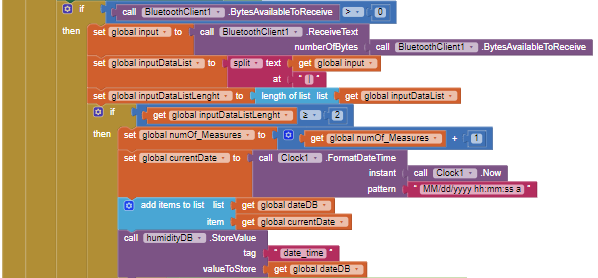
**- να αποθηκεύονται** στις **κατάλληλες** λίστες **(5)**

**- να ενημερώνεται** η βάση δεδομένων **(6)**

**- αν δεν υπάρχει σύνδεση με το βραχίονα** **(7)**

-να τίθενται κατάλληλες τιμές σε αντικείμενα (labels, textboxes …) **(8)**

Η παραπάνω λογική υλοποιείται με τα παρακάτω block κώδικα:

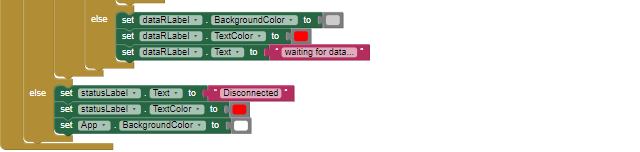
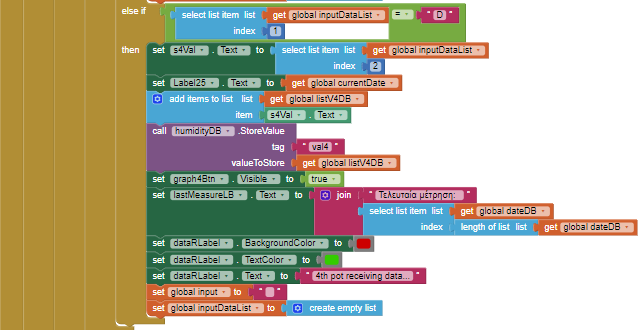
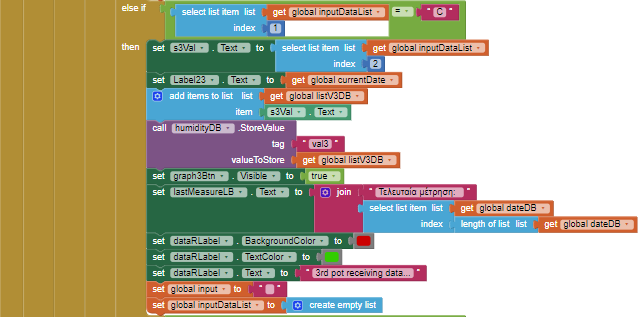
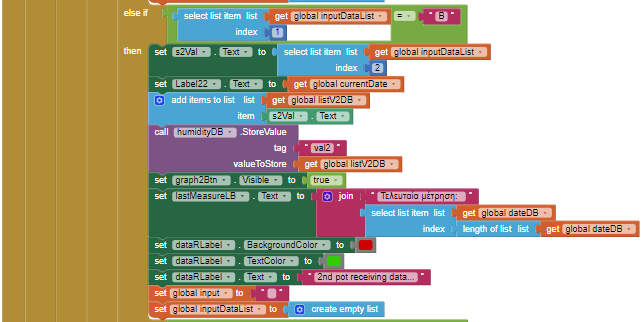
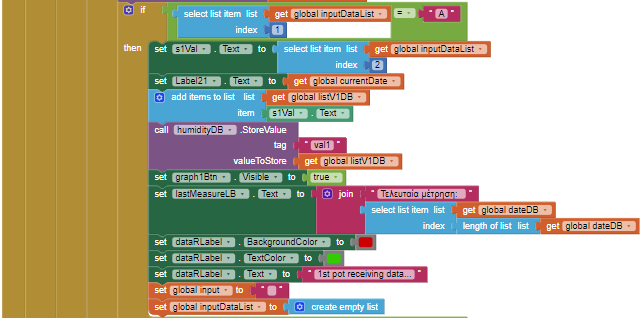
**(6)**

**(5)**

**(3)**

**(2)**

**(1)**



**(6)**

**(5)**

**(4)**

**(8)**

**(7)**

**Ανάπτυξη Εφαρμογής (Μέρος 3ο)**

## Οθόνη App (Συνέχεια…)

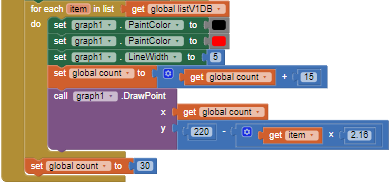
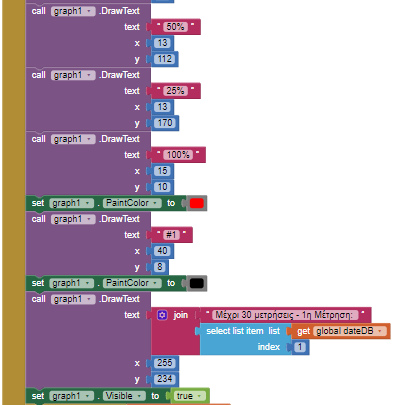
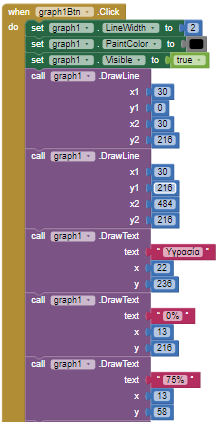
Αν και εφόσον υπάρχουν μετρήσεις στη βάση δεδομένων ή έχουν ληφθεί από το βραχίονα εμφανίζεται κάτω από κάθε γλάστρα το εικονίδιο



Επιλέγοντας το ο χρήστης θα μπορεί να δει σε **γράφημα στην οθόνη** του τις 30 τελευταίες μετρήσεις της εκάστοτε γλάστρας. Εδώ θα πρέπει να σημειωθεί ότι θα πρέπει να δηλώσουμε μια μεταβλητή ως εξής:



Με αυτό τον τρόπο θα έχει μια καλύτεροι εποπτεία της υγρασίας της γλάστρας. Τα block που ενεργοποιούνται όταν επιλεγεί το συγκεκριμένο εικονίδιο είναι:



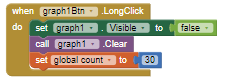
- Δημιουργούμε τους άξονες X-Y στην οθόνη με τις αντίστοιχες τιμές αναφοράς

Όταν ο χρήστης πατήσει παρατεταμένα το κουμπί

- Οι τιμές που είναι αποθηκευμένες στην αντίστοιχη λίστα θα πρέπει να αναπαρασταθούν στους άξονες X-Y που έχουμε δημιουργήσει.



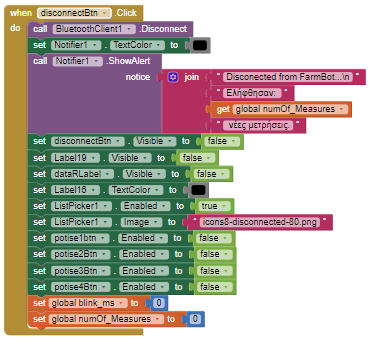
το γράφημα θα σβήσει, τα αντίστοιχα block είναι:



Ο κώδικας για το γράφημα θα πρέπει να είναι ο ίδιος (αλλάζοντας βέβαια μόνο τις αναφορές στα αντικείμενα) και για τις υπόλοιπες 3 γλάστρες.

Θα πρέπει επίσης να μεριμνήσουμε τι θα γίνει όταν ο χρήστης επιλέξει να **αποσυνδεθεί από το βραχίονα.** Τότε θα πρέπει

* να γίνει αποσύνδεση του βραχίονα
* να ενημερωθεί ο χρήστης για τις μετρήσεις που πάρθηκαν
* να τεθούν στα αντίστοιχα αντικείμενα (labels, textboxes, μεταβλητές…) οι κατάλληλες τιμές



Τέλος, αν ο χρήστης πατήσει το **πλήκτρο πίσω** θα πρέπει:

* να αποσυνδέεται ο βραχίονας
* να ειδοποιείται ο χρήστης με κατάλληλο μήνυμα
* να κλείνει η οθόνη

