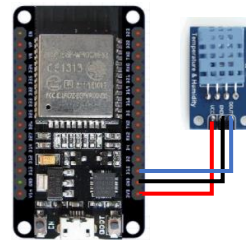


ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ  
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ  
ΗΛΕΚΤΡΟΝΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

«Ανάπτυξη android εφαρμογής παρακολούθησης  
περιβαλλοντικών συνθηκών με esp32»



**Φοιτητής**

Μιμήκος Σταμάτης 514088

**Επιβλέπων**

Δρ. Κυριάκος Τσιακμάκης

Ιούνιος 2021

Ανάπτυξη android εφαρμογής παρακολούθησης περιβαλλοντικών συνθηκών με esp32

Κωδικός: 20201

Φοιτητής: Μιμήκος Σταμάτης

Εισηγητής: Δρ Κυριάκος Τσιακμάκης

Ημερομηνία ανάληψης Π.Ε. 15-10-2020

Ημερομηνία περάτωσης Π.Ε. 02-06-2021

*Βεβαιώνω ότι είμαι ο συγγραφέας αυτής της εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, έχω καταγράψει τις όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών, εικόνων και κειμένου, είτε αυτές αναφέρονται ακριβώς είτε παραφρασμένες. Επιπλέον, βεβαιώνω ότι αυτή η εργασία προετοιμάστηκε από εμένα προσωπικά, ειδικά ως πτυχιακή εργασία, στο Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του ΔΙ.ΠΑ.Ε.*

*Η παρούσα εργασία αποτελεί πνευματική ιδιοκτησία του φοιτητή Μιμήκου Σταμάτη που την εκπόνησε. Στο πλαίσιο της πολιτικής ανοικτής πρόσβασης, ο συγγραφέας/δημιουργός εκχωρεί στο Διεθνές Πανεπιστήμιο της Ελλάδος άδεια χρήσης του δικαιώματος αναπαραγωγής, δανεισμού, παρουσίασης στο κοινό και ψηφιακής διάχυσης της εργασίας διεθνώς, σε ηλεκτρονική μορφή και σε οποιοδήποτε μέσο, για διδακτικούς και ερευνητικούς σκοπούς, άνευ ανταλλάγματος. Η ανοικτή πρόσβαση στο πλήρες κείμενο της εργασίας, δεν σημαίνει καθ' οιονδήποτε τρόπο παραχώρηση δικαιωμάτων διανοητικής ιδιοκτησίας του συγγραφέα/δημιουργού, ούτε επιτρέπει την αναπαραγωγή, αναδημοσίευση, αντιγραφή, πώληση, εμπορική χρήση, διανομή, έκδοση, μεταφόρτωση (downloading), ανάρτηση (uploading), μετάφραση, τροποποίηση με οποιονδήποτε τρόπο, τμηματικά ή περιληπτικά της εργασίας, χωρίς τη ρητή προηγούμενη έγγραφη συναίνεση του συγγραφέα/δημιουργού.*

Η έγκριση της πτυχιακής εργασίας από το Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του Διεθνούς Πανεπιστημίου της Ελλάδος, δεν υποδηλώνει απαραίτητως και αποδοχή των απόψεων του συγγραφέα, εκ μέρους του Τμήματος.



## Περίληψη

Η εργασία αυτή αφορά τη μελέτη και ανάπτυξη εφαρμογής σε κινητό τηλέφωνο για την παρακολούθηση περιβαλλοντικών συνθηκών με esp32. Η εφαρμογή android επικοινωνεί με το esp32 μέσω ασύρματου τοπικού δικτύου για την παρακολούθηση θερμοκρασίας και υγρασίας σε έναν εσωτερικό ή εξωτερικό χώρο. Ο χρήστης έχει τη δυνατότητα μέσω της φορητής συσκευής μικρού κόστους και κατανάλωσης και να επιβλέπει στο κινητό του τις τιμές των αισθητήρων σε πολύ μικρό χρόνο. Η χρησιμότητα της εφαρμογής είναι η παρακολούθηση της θερμοκρασίας/υγρασίας σε διάφορα σημεία ενός σπιτιού, αποθήκη, γενικά ενός χώρου. Ακόμη, ο χρήστης μπορεί να προσθέσει περισσότερους κόμβους με αισθητήρια για να παρακολουθήσει ενιαία από την εφαρμογή. Επιπλέον, προστέθηκε η δυνατότητα επίβλεψης των τιμών των αισθητήρων διαδικτυακά με τη βοήθεια της υπηρεσίας Thingspeak. Κάθε αισθητήρας-κόμβος ταυτόχρονα στέλνει τα δεδομένα του διαδικτυακά μέσω καναλιού στο Thingspeak που έχει οριστεί και ο χρήστης μέσω της εφαρμογής που υλοποιήθηκε έχει τη δυνατότητα να τα παρακολουθεί σε πραγματικό χρόνο από το κινητό.

## « Development of android application for monitoring environmental conditions with esp32»

### **Abstract**

This work concerns the study and development of an application on a mobile phone for the monitoring of environmental conditions with esp32. The android application communicates with esp32 via wireless local area network to monitor temperature and humidity indoors or outdoors. The user has the ability through the mobile device of low cost to monitor the prices of the sensors from mobile in a very short time. The application is monitoring temperature / humidity in various parts of a house, warehouse, generally a space. Also, the user can add more nodes with sensors to watch uniformly from the application. In addition, the ability to monitor sensor prices online with the help of the Thingspeak service has been added. Each sensor-node simultaneously sends its data online via a channel to the defined Thingspeak and the user through the implemented application has the ability to monitor it in real time from the mobile.

## **Ευχαριστίες**

Θέλω να ευχαριστήσω τους γονείς μου για τη πολύτιμη βοήθεια τους.

# Περιεχόμενα

Περίληψη.....	iv
Abstract .....	v
Ευχαριστίες.....	vi
Περιεχόμενα .....	vii
Κατάλογος Σχημάτων .....	viii
Κεφάλαιο 1ο: Εισαγωγή.....	9
1.1 Εισαγωγή.....	9
1.2 Δομή της εργασίας .....	10
Κεφάλαιο 2ο: Το σύστημα παρακολούθησης αισθητήρων .....	12
2.1 Εισαγωγή.....	12
Κεφάλαιο 3ο: Οι γλώσσες προγραμματισμού, τα αναπτυξιακά εργαλεία και τα ηλεκτρονικά εξαρτήματα    21	
3.1 ESP32.....	21
3.2 Αισθητήρας DHT11 .....	26
3.3 Flutter.....	27
3.4 MicroPython.....	33
3.5 Thingspeak.....	35
Κεφάλαιο 4ο: Οι εφαρμογές στο κινητό και στο esp32.....	44
4.1 Εισαγωγή - Πλοήγηση.....	44
4.2 Επεξήγηση και διαγράμματα των εφαρμογών .....	50
4.2.1 Το πρόγραμμα στο esp32.....	50
Κεφάλαιο 5ο: Συμπεράσματα και προτάσεις βελτίωσης.....	58
ΒΙΒΛΙΟΓΡΑΦΙΑ .....	59
ΠΑΡΑΡΤΗΜΑ Α.....	60

## Κατάλογος Σχημάτων

Εικόνα 2.1: Επικοινωνία εφαρμογής στο κινητό με το esp32.....	12
Εικόνα 2.2: Επικοινωνία esp32 με το Thingspeak.....	13
Εικόνα 2.3: Επικοινωνία esp32 με το Thingspeak και με την εφαρμογή στο κινητό.....	14
Εικόνα 2.4: Η επικοινωνία σε ένα εσωτερικό χώρο όταν το κινητό και esp32 βρίσκονται στο ίδιο τοπικό δίκτυο.....	15
Εικόνα 2.5: Η επικοινωνία σε ένα χώρο όταν η εφαρμογή στο κινητό βρίσκεται εκτός του τοπικού δικτύου.....	16
Εικόνα 2.6: Επικοινωνία της εφαρμογής με πολλά esp32-αισθητήρες σε εσωτερικό δίκτυο.....	17
Εικόνα 2.7: Επικοινωνία εφαρμογής με ένα εκτός και με πολλούς κόμβους σε εσωτερικό δίκτυο.....	18
Εικόνα 2.8: Επικοινωνία της εφαρμογής με πολλούς κόμβους όταν βρίσκεται εκτός του εσωτερικού δικτύου.....	19
Εικόνα 2.9: Η εφαρμογή μπορεί να παρακολουθεί esp32 εκτός και εντός δικτύου μέσω Thingspeak.....	20
Εικόνα 3.1: esp32 με τους ακροδέκτες.....	23
Εικόνα 3.2: esp32-CAM.....	25
Εικόνα 3.3: DTH11 Αισθητήρας.....	27
Εικόνα 3.4: Συγκριτικά αποτελέσματα τεχνολογιών για υλοποίηση εφαρμογών σε κινητά – Περίοδος 2018-2021.....	29
Εικόνα 3.5: Συγκριτικά αποτελέσματα τεχνολογιών για υλοποίηση εφαρμογών σε κινητά – Περίοδος Απρίλιος 2021.....	29
Εικόνα 3.6: Συγκριτικά αποτελέσματα τεχνολογιών για υλοποίηση εφαρμογών σε κινητά – Περιοχή Ελλάδα – Περίοδος 2018-2021.....	30
Εικόνα 3.7: Flutter Framework.....	31
Εικόνα 3.8: MicroPython χρησιμοποιείται ευρέως για esp32 και esp8266.....	33
Εικόνα 3.9: Δημιουργία λογαριασμού.....	37
Εικόνα 3.10: Στοιχεία Χρήστη - Thingspeak.....	38
Εικόνα 3.11: Δημιουργία Καναλιού και Ρυθμίσεις.....	38
Εικόνα 3.12: Χαρακτηριστικά Καναλιού Thingspeak.....	39
Εικόνα 3.13: Thingspeak – Προσθήκη πρόσθετων.....	40
Εικόνα 3.14: Πρόσθετα-Widgets.....	40
Εικόνα 3.15: Εμφάνιση τιμών για τον αισθητήρα Θερμοκρασίας.....	40
Εικόνα 3.16: Εμφάνιση τιμών για τον αισθητήρα Υγρασίας.....	41
Εικόνα 3.17: Τα κλειδιά του API για εγγραφή και ανάγνωση αισθητήρων από το κανάλι.....	42
Εικόνα 4.1: Κεντρική Οθόνη με τιμές.....	44
Εικόνα 4.2: Οθόνη για επιλογή κόμβου.....	45
Εικόνα 4.3: Οθόνη για τις επιλογές στη διαχείριση κόμβων.....	46
Εικόνα 4.4: Δημιουργία νέου κόμβου.....	47
Εικόνα 4.5: Επεξεργασία κόμβου.....	48
Εικόνα 4.6: Οθόνη με τιμές από τον αισθητήρα όταν δεν ανταποκρίνεται.....	49
Εικόνα 4.7: Esp32 – Συνολική Εφαρμογή.....	50

Εικόνα 4.8: Esp32 – Διάγραμμα ροής για το αποστολή τιμών στο Thingspeak .....	51
Εικόνα 4.9: Esp32 – Διάγραμμα ροής για το αποστολή τιμών στο Κινητό αφού απαιτήθηκε .....	53
Εικόνα 4.10: Esp32 – Διαδικασία με Token .....	54
Εικόνα 4.11: Διάγραμμα ροής για την εφαρμογή στο κινητό .....	56



# Κεφάλαιο 1ο: Εισαγωγή

## 1.1 Εισαγωγή

Το Διαδίκτυο των πραγμάτων (IoT) αναφέρεται σε ένα μεγάλο αριθμό “πραγμάτων” που είναι συνδεδεμένα στο Διαδίκτυο, ώστε να μπορούν να μοιράζονται δεδομένα με άλλα “πράγματα” όπως εφαρμογές IoT, συνδεδεμένες συσκευές, βιομηχανικά μηχανήματα και άλλα. Οι συνδεδεμένες στο Διαδίκτυο συσκευές χρησιμοποιούν ενσωματωμένους αισθητήρες για τη συλλογή δεδομένων. Οι συσκευές και τα μηχανήματα που συνδέονται με το IoT μπορούν να βελτιώσουν τον τρόπο που εργαζόμαστε και ζούμε. Μερικά παραδείγματα Internet of Things μπορούν να είναι ένα έξυπνο σπίτι που προσαρμόζει αυτόματα τη θέρμανση και το φωτισμό μέχρι ένα έξυπνο εργοστάσιο που παρακολουθεί τα βιομηχανικά μηχανήματα για να αναζητούν προβλήματα και στη συνέχεια να προσαρμόζεται αυτόματα για την αποφυγή βλαβών.

Το καλύτερο εργαλείο που μπορεί να χρησιμοποιηθεί για να αξιοποιήσει σωστά τα οφέλη των IoT (internet of things - διαδίκτυο των πραγμάτων) είναι το κινητό τηλέφωνο. Υπάρχουν σήμερα δισεκατομμύρια χρήστες έξυπνων τηλεφώνων στον κόσμο και η ανάλυση της αγοράς δείχνει ότι οι χρήστες smartphone θα αυξηθούν σημαντικά τα επόμενα χρόνια.

Επιπλέον, η πρόσβαση σε λύσεις IoT μέσω εφαρμογών για κινητά είναι λογική επειδή η ανάπτυξη εφαρμογών για κινητά είναι φθηνή. Αυτό τα καθιστά ακόμη πιο προσβάσιμα. Επίσης, προσφέρουν μια πιο ευέλικτη πλατφόρμα για τη μετάδοση δεδομένων σε σύγκριση με τις εφαρμογές ιστού.

Η ανάπτυξη IoT σε εφαρμογές για κινητά διευκολύνει τον τηλε-χειρισμό άλλων έξυπνων συσκευών χρησιμοποιώντας μία μόνο φορητή συσκευή. Τα smartphones έχουν απλοποιήσει πάρα πολύ τη ζωή μας. Ωστόσο, με το IoT, είναι δυνατή η διασύνδεση χιλιάδων συσκευών και η ανταλλαγή πληροφοριών, απλοποιώντας περαιτέρω τη ζωή μας. Μερικοί από τους τρόπους με τους οποίους οι εφαρμογές για κινητά αξιοποιούν το Διαδίκτυο των πραγμάτων αφορούν τα wearables, τομείς υγείας, το έξυπνο σπίτι και αγροτικούς τομείς.

Τα Wearables αποτελούν αναπόσπαστο μέρος των λύσεων IoT. Χρησιμοποιώντας συνδέσεις όπως Bluetooth και Wi-Fi, μπορείτε να συνδέσετε φορητά όπως ρολόγια χειρός, γυαλιά και ακόμη και ζώνες με τα smartphones. Με αυτόν τον τρόπο, οποιαδήποτε μορφή δεδομένων μπορεί να ανταλλάσσεται μεταξύ των δύο συσκευών.

Οι οικιακές ιατρικές συσκευές γίνονται πιο δημοφιλείς τα τελευταία χρόνια, χάρη στην αξιοποίηση της δύναμης της ανάπτυξης εφαρμογών IoT. Οι ιατρικές συσκευές συνδέονται με αισθητήρες που λαμβάνουν δεδομένα υγείας από τον ασθενή που μεταφέρονται στην εφαρμογή για κινητά. Αυτά τα δεδομένα μπορούν να μεταφερθούν εξ αποστάσεως στον γιατρό ή τα μέλη της οικογένειας σε

περίπτωση έκτακτης ανάγκης. Με αυτόν τον τρόπο, η ανάγκη επίσκεψης στο νοσοκομείο για μικρά ζητήματα μειώνεται σημαντικά.

Με τη δύναμη των εξελίξεων του IoT, ο αυτοματισμός στο σπίτι έχει μεταφερθεί σε ένα εντελώς νέο επίπεδο. Οι οικιακές συσκευές όπως λαμπτήρες, συστήματα θέρμανσης / ψύξης, κλειδαριές και άλλα μπορούν να ελέγχονται από απόσταση χρησιμοποιώντας μια φορητή συσκευή.

Η ανάγκη για βελτιωμένη γεωργία για την αντιμετώπιση της συνεχώς αυξανόμενης ζήτησης τροφίμων είναι απαραίτητη. Η καλλιέργεια με βάση το IoT χρησιμοποιείται για την παρακολούθηση καλλιεργειών πεδίου με αισθητήρες (φως, θερμοκρασία, υγρασία, υγρασία εδάφους κ.λπ.). Με αυτούς τους αισθητήρες, οι αγρότες μπορούν να παρακολουθούν εξ αποστάσεως τις συνθήκες της γεωργικής γης χρησιμοποιώντας τις κινητές τους συσκευές. Με αυτόν τον τρόπο, μπορεί να επιτευχθεί αποτελεσματική γεωργία.

Η εργασία αυτή αφορά τη μελέτη και ανάπτυξη εφαρμογής σε κινητό τηλέφωνο για την παρακολούθηση περιβαλλοντικών συνθηκών με esp32. Η εφαρμογή android επικοινωνεί με το esp32 μέσω ασύρματου τοπικού δικτύου για την παρακολούθηση θερμοκρασίας και υγρασίας σε έναν εσωτερικό ή εξωτερικό χώρο. Ο χρήστης έχει τη δυνατότητα μέσω της φορητής συσκευής μικρού κόστους και κατανάλωσης και να επιβλέπει στο κινητό του τις τιμές των αισθητήρων σε πολύ μικρό χρόνο. Η χρησιμότητα της εφαρμογής είναι η παρακολούθηση της θερμοκρασίας/υγρασίας σε διάφορα σημεία ενός σπιτιού, αποθήκη, γενικά ενός χώρου. Ακόμη, ο χρήστης μπορεί να προσθέσει περισσότερους κόμβους με αισθητήρια για να παρακολουθήσει ενιαία από την εφαρμογή. Επιπλέον, προστέθηκε η δυνατότητα επίβλεψης των τιμών των αισθητήρων διαδικτυακά με τη βοήθεια της υπηρεσίας Thingspeak. Κάθε αισθητήρας-κόμβος ταυτόχρονα στέλνει τα δεδομένα του διαδικτυακά μέσω καναλιού στο Thingspeak που έχει οριστεί και ο χρήστης μέσω της εφαρμογής που υλοποιήθηκε έχει τη δυνατότητα να τα παρακολουθεί σε πραγματικό χρόνο από το κινητό του.

Ο πρώτος στόχος αυτής της εργασίας είναι να μπορεί ο χρήστης με την εφαρμογή android στο κινητό του να παρακολουθεί τους αισθητήρες σε έναν χώρο. Το esp32 που έχει τους αισθητήρες πρέπει να είναι συνδεδεμένος στο ίδιο τοπικό δίκτυο με το κινητό.

Ο δεύτερος στόχος είναι η παροχή δυνατότητας παρακολούθησης των κόμβων μέσω διαδικτύου μέσω της υπηρεσίας Thingspeak αλλά και η προσθήκη, επεξεργασία και διαγραφή νέων esp32 με αισθητήρες.

## **1.2 Δομή της εργασίας**

Στο πρώτο κεφάλαιο παρουσιάζεται μια μικρή εισαγωγή, οι στόχοι της και η δομή της πτυχιακής εργασίας.

Στο δεύτερο κεφάλαιο παρουσιάζεται η εισαγωγή στο σύστημα παρακολούθησης αισθητήρων.

Στο τρίτο κεφάλαιο περιγράφονται οι γλώσσες προγραμματισμού, τα αναπτυξιακά εργαλεία και τα ηλεκτρονικά εξαρτήματα που χρησιμοποιήθηκαν.

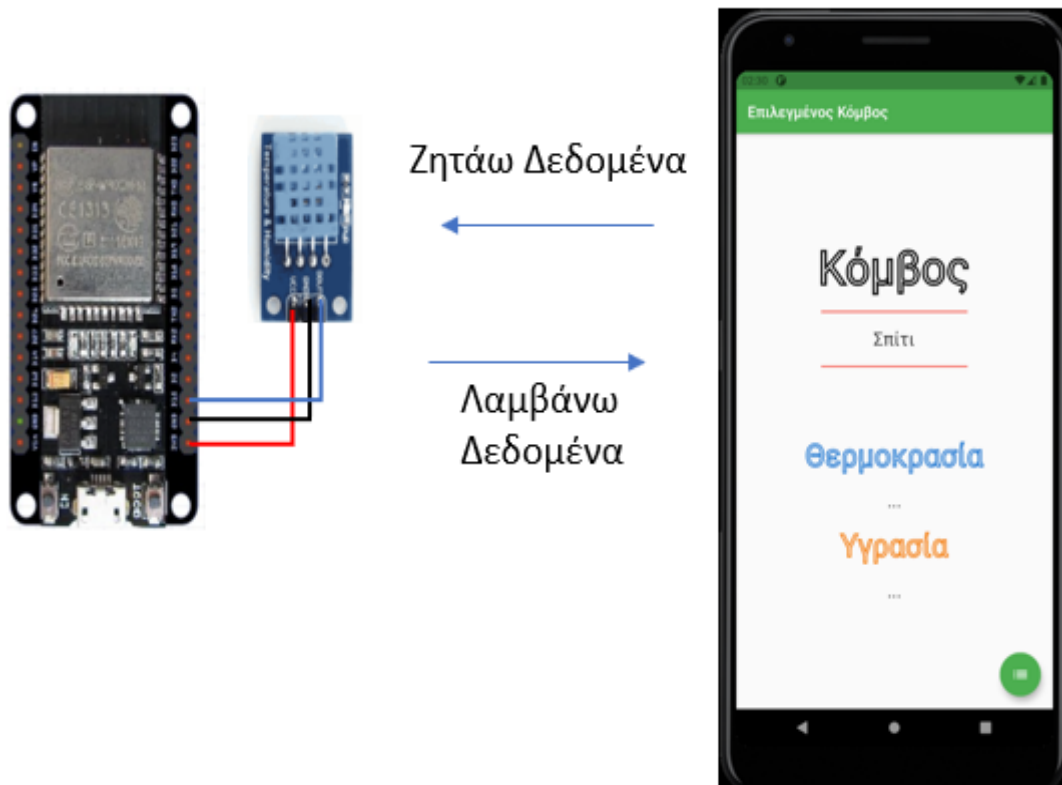
Στο τέταρτο κεφάλαιο περιγράφονται τα προγράμματα στο esp32 και η εφαρμογή στο κινητό με διαγράμματα.

Τέλος, στο πέμπτο κεφάλαιο παρουσιάζονται τα συμπεράσματα της εργασίας και θέματα για βελτίωση ενώ στο τέλος της εργασίας παρατίθενται κάποια κομμάτια από τους κώδικες που χρησιμοποιήθηκαν για την υλοποίηση του πειραματικού μέρους στην εργασία.

## Κεφάλαιο 2ο: Το σύστημα παρακολούθησης αισθητήρων

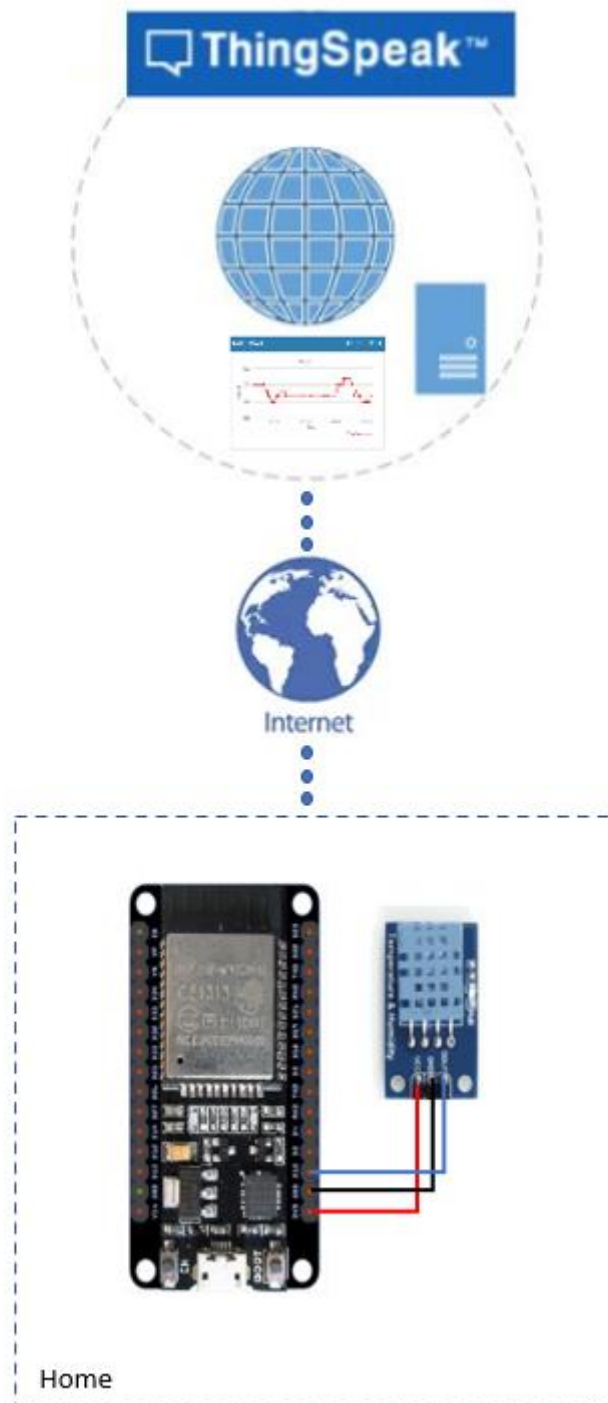
### 2.1 Εισαγωγή

Στην εικόνα 2.1. παρουσιάζεται το σύστημα παρακολούθησης που υλοποιήθηκε ανάμεσα στον στην εφαρμογή στο κινητό και στο esp32.



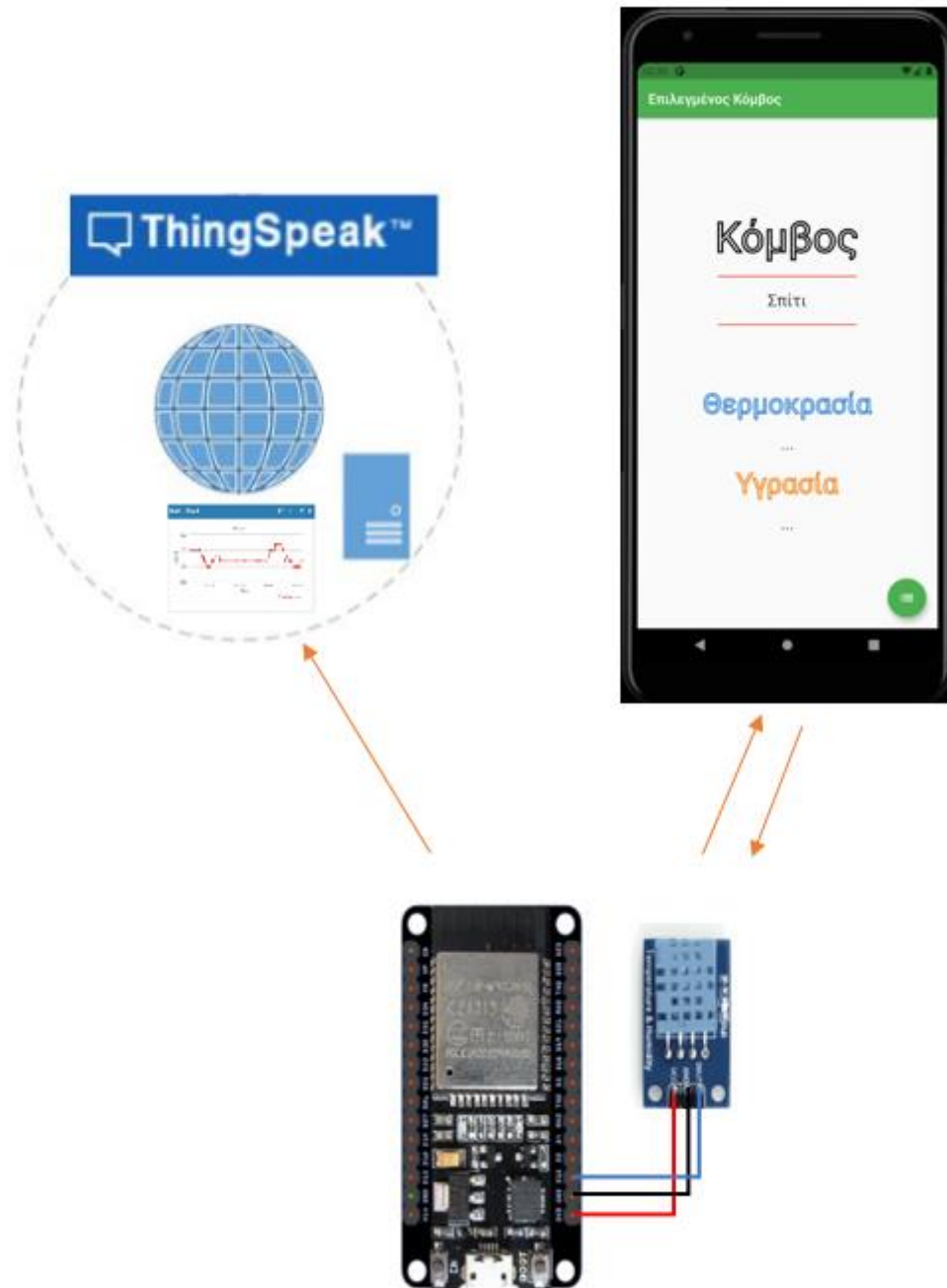
Εικόνα 2.1: Επικοινωνία εφαρμογής στο κινητό με το esp32

Αν η εφαρμογή στο κινητό γνωρίζει τη διεύθυνση του κόμβου-esp32 τότε με την επιλογή του κόμβου η εφαρμογή του ζητάει να επιστρέψει τις τιμές των αισθητήρων.



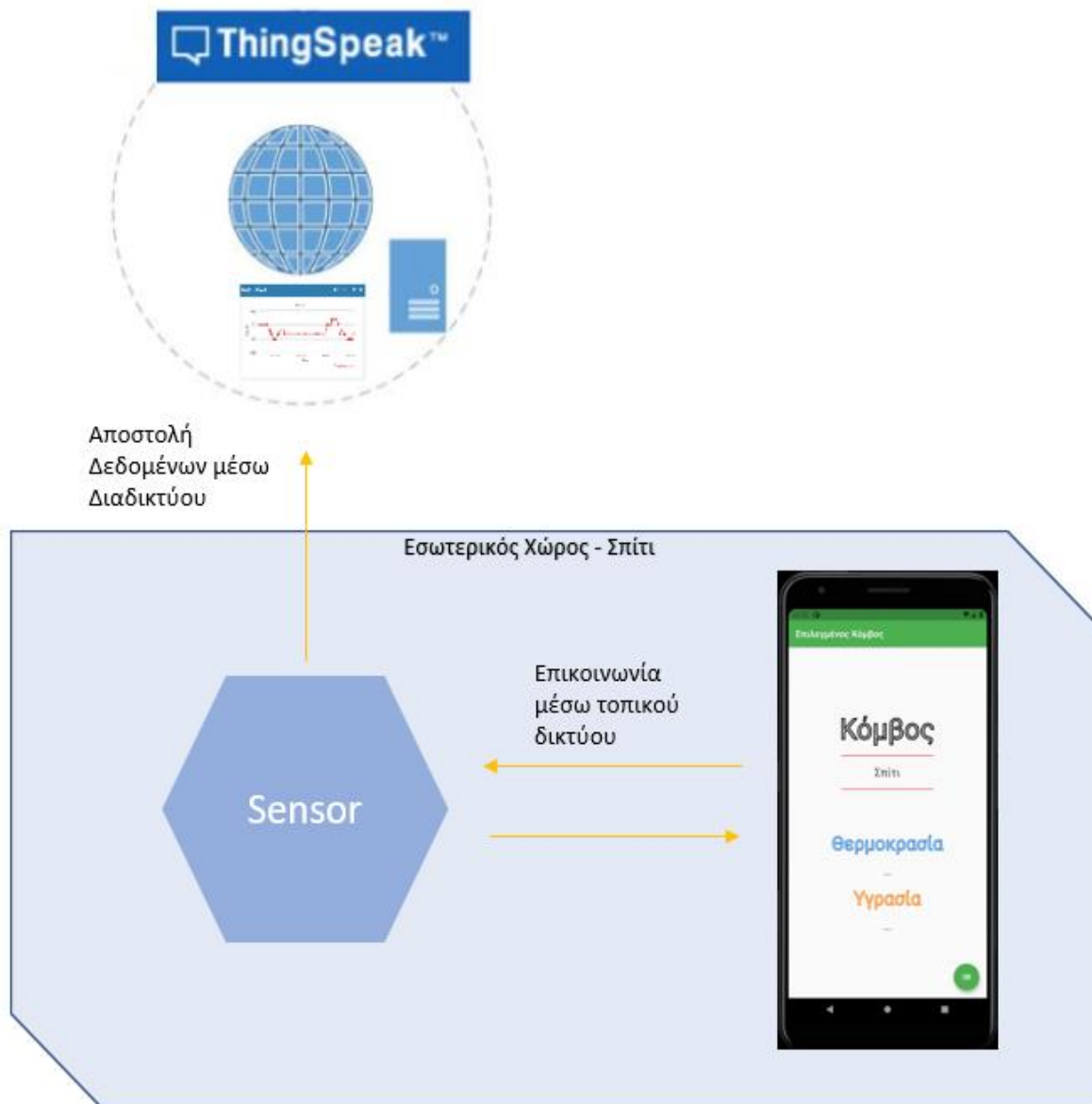
Εικόνα 2.2: Επικοινωνία esp32 με το Thingspeak

Εκτός από την επικοινωνία του esp32 με το κινητό μέσω του Wifi το esp32 στέλνει τις τιμές που διάβασε από τους αισθητήρες στο Thingspeak, όπως φαίνεται στην Εικόνα 2.2. Το Thingspeak ως server έχει μια διεύθυνση και κάποια χαρακτηριστικά και η εφαρμογή στο esp32 πρέπει να τα γνωρίζει.



Εικόνα 2.3: Επικοινωνία esp32 με το Thingspeak και με την εφαρμογή στο κινητό

Οι δύο βασικές λειτουργίες επικοινωνίας παρουσιάζονται στην Εικόνα 2.3. Σε συγκεκριμένες χρονοθυρίδες το esp32 αποστέλνει τις τιμές που διάβασε από τους αισθητήρες στο server Thingspeak. Αν ζητηθεί από την εφαρμογή στο κινητό το esp32 της στέλνει τις τιμές που διάβασε από τους αισθητήρες.

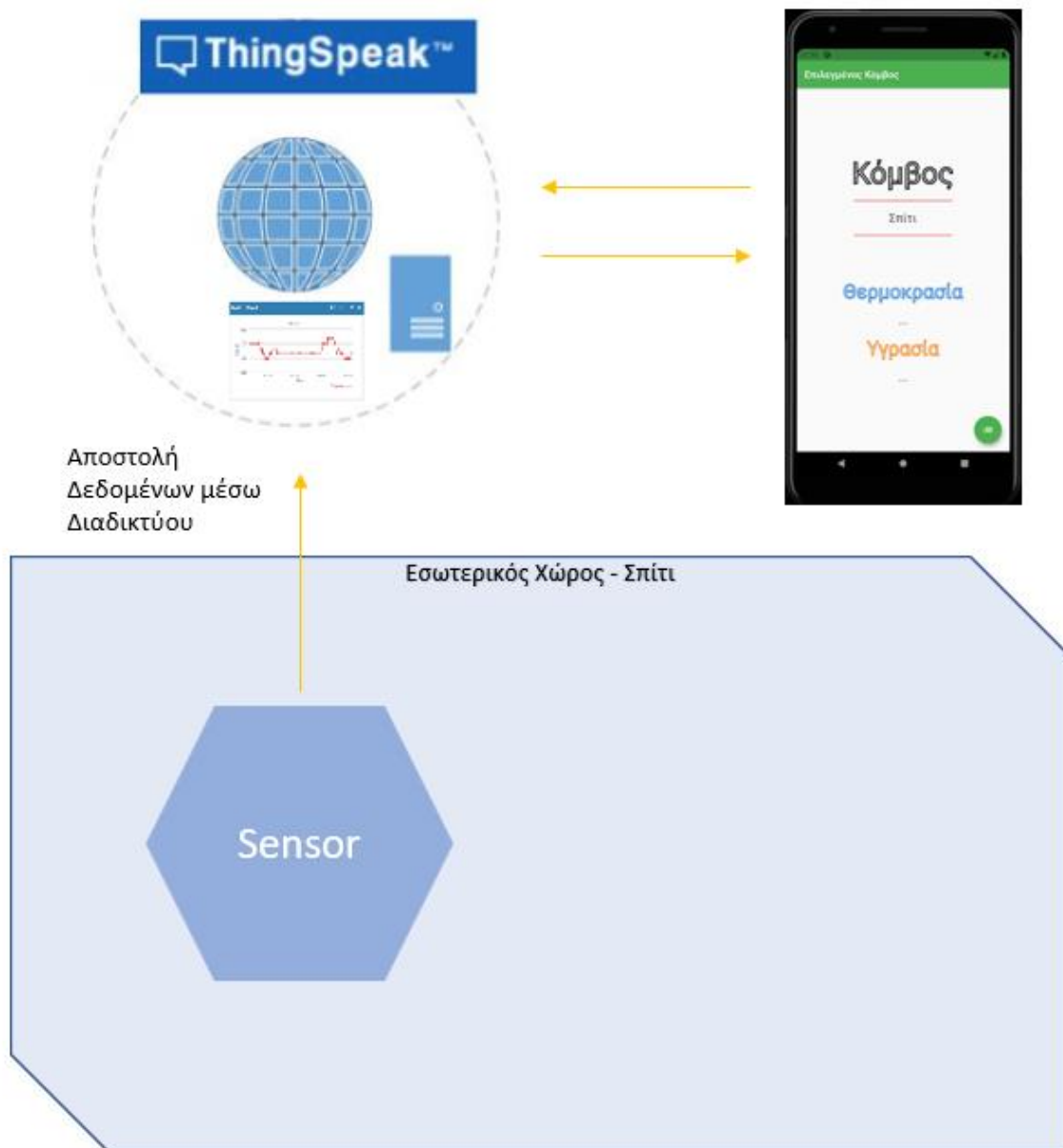


Εικόνα 2.4: Η επικοινωνία σε ένα εσωτερικό χώρο όταν το κινητό και esp32 βρίσκονται στο ίδιο τοπικό δίκτυο

Όταν το esp32 βρίσκεται σε ένα σπίτι ή ένα μέρος όπου το τοπικό δίκτυο βρίσκεται πίσω από κάποιο δρομολογητή ή firewall ο τρόπος επικοινωνίας παρουσιάζεται στην Εικόνα 2.4. Όταν βρίσκονται και οι δύο συνδεδεμένοι στο ίδιο τοπικό δίκτυο το κινητό μπορεί να επικοινωνήσει με το esp32. Στη περίπτωση που η εφαρμογή είναι εκτός τοπικού δικτύου τότε δεν μπορεί να επικοινωνήσει με το esp32.

Για να επιτευχθεί η επικοινωνία πρέπει να γίνει port-forwarding στο router, που σημαίνει ότι πρέπει ο χρήστης να μπει στις ρυθμίσεις του router και να βάλει την εξωτερική ip του δρομολογητή να αντιστοιχεί στον esp32.

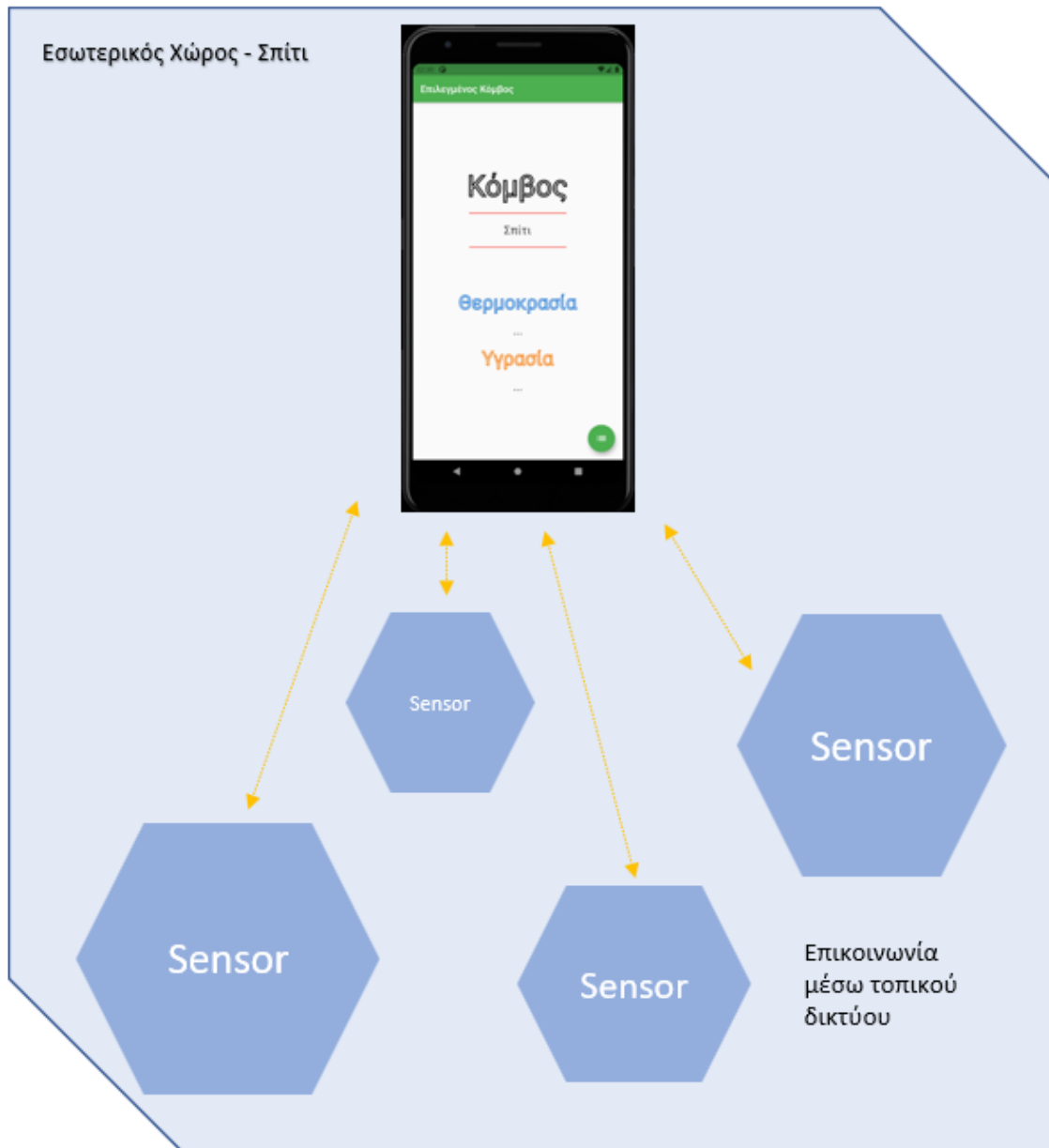
Η επικοινωνία του esp32 με το Thingspeak μπορεί να γίνει χωρίς port forwarding.



Εικόνα 2.5: Η επικοινωνία σε ένα χώρο όταν η εφαρμογή στο κινητό βρίσκεται εκτός του τοπικού δικτύου

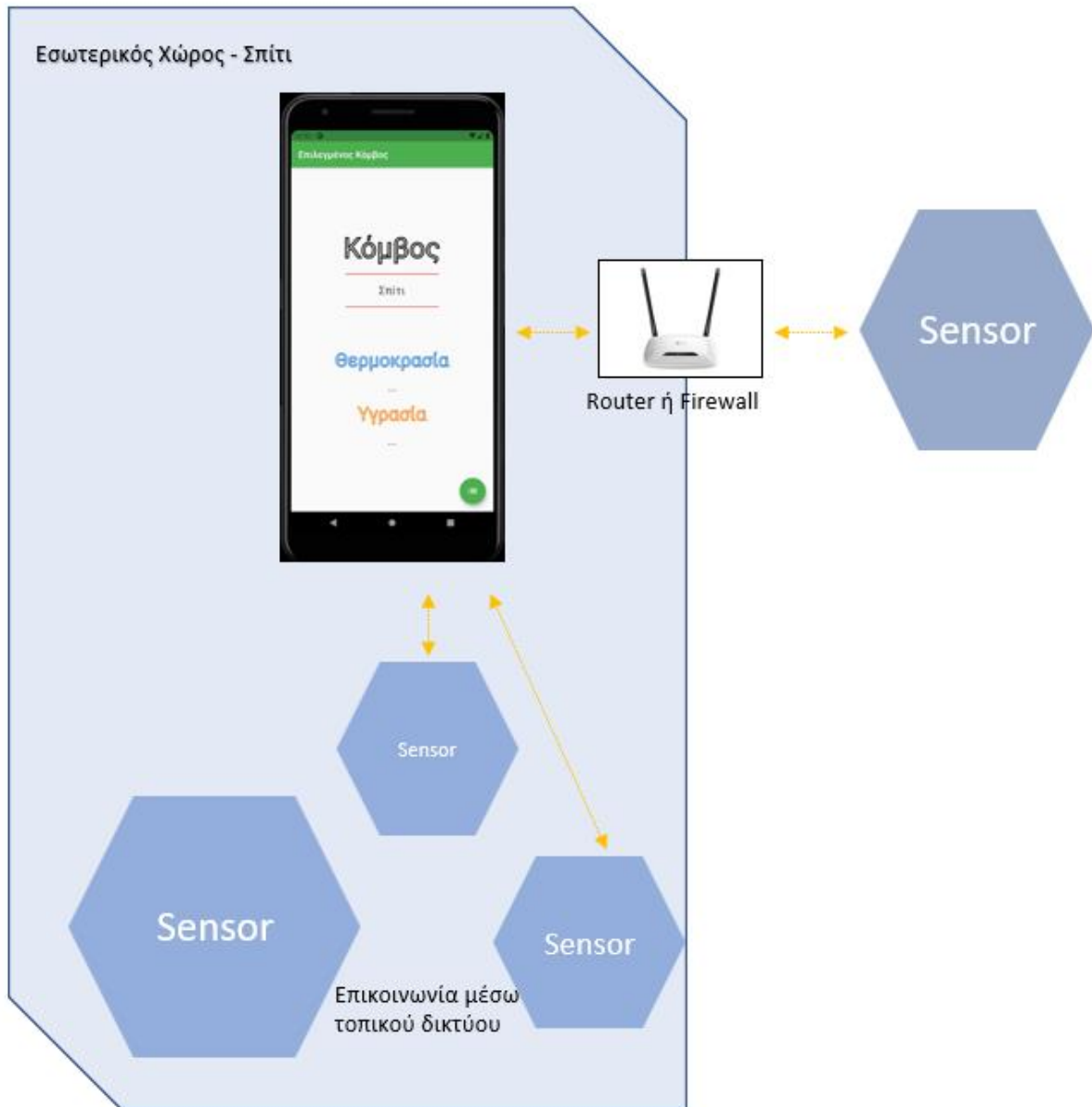
Ένας εναλλακτικός τρόπος για να γίνει επιτυχώς η επικοινωνία της εφαρμογής με το esp32 όταν δεν βρίσκονται στον ίδιο τοπικό δίκτυο είναι η επικοινωνία της εφαρμογής με το Thingspeak, χωρίς να υπάρχει άμεση επικοινωνία με το esp32. Αυτό παρουσιάζεται στην Εικόνα 2.5. Το Thingspeak παίρνει τις τιμές κάθε  $t$  δευτερόλεπτα και έτσι η εφαρμογή μπορεί να τις λαμβάνει από το server.

Στην εφαρμογή έχουν υλοποιηθεί και οι δύο τρόποι

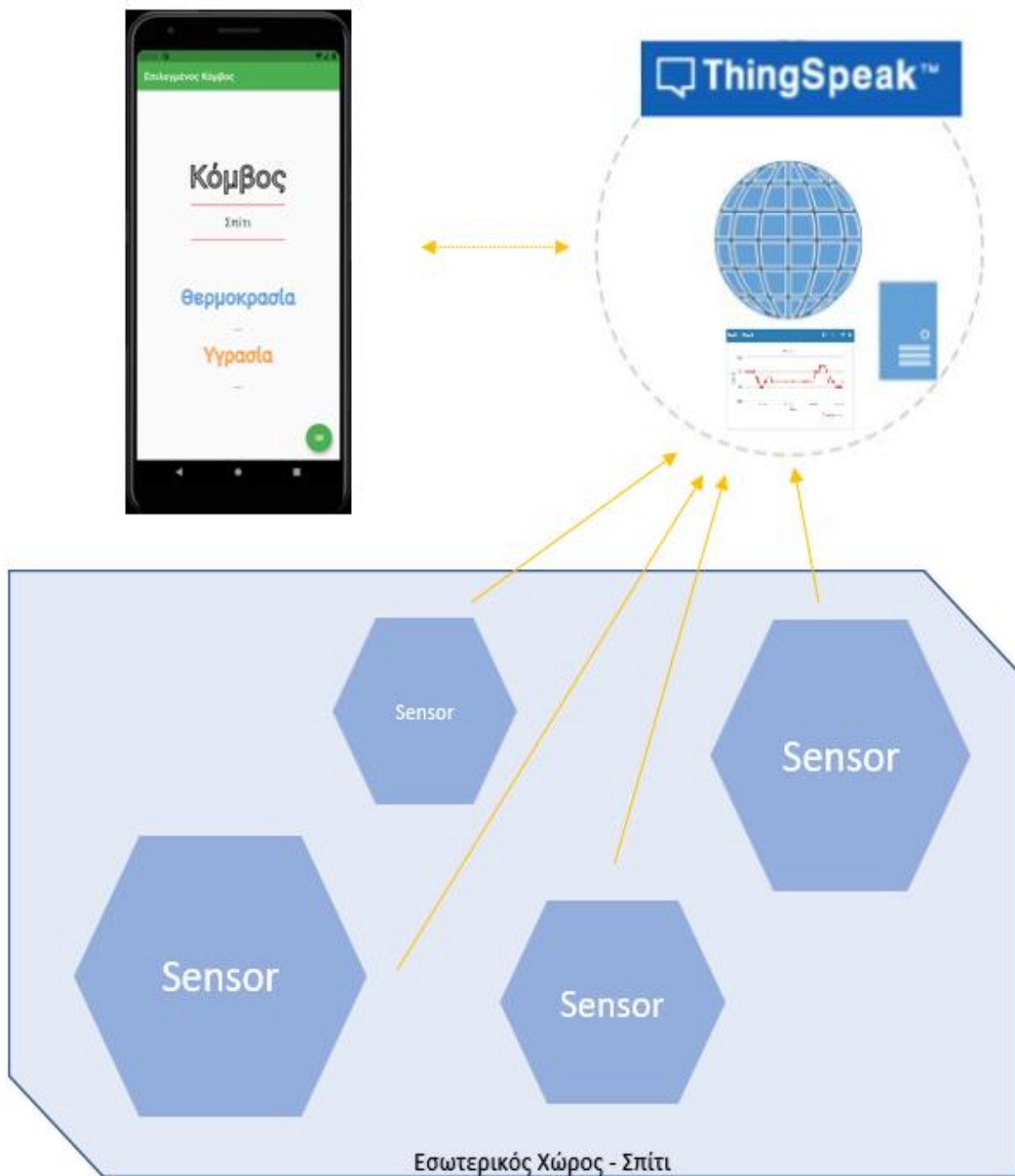


Εικόνα 2.6: Επικοινωνία της εφαρμογής με πολλά esp32-αισθητήρες σε εσωτερικό δίκτυο

Στην Εικόνα 2.6 παρουσιάζεται η δυνατότητα της εφαρμογής να επικοινωνεί με όσους κόμβους υπάρχουν σε ένα εσωτερικό χώρο στο ίδιο δίκτυο.

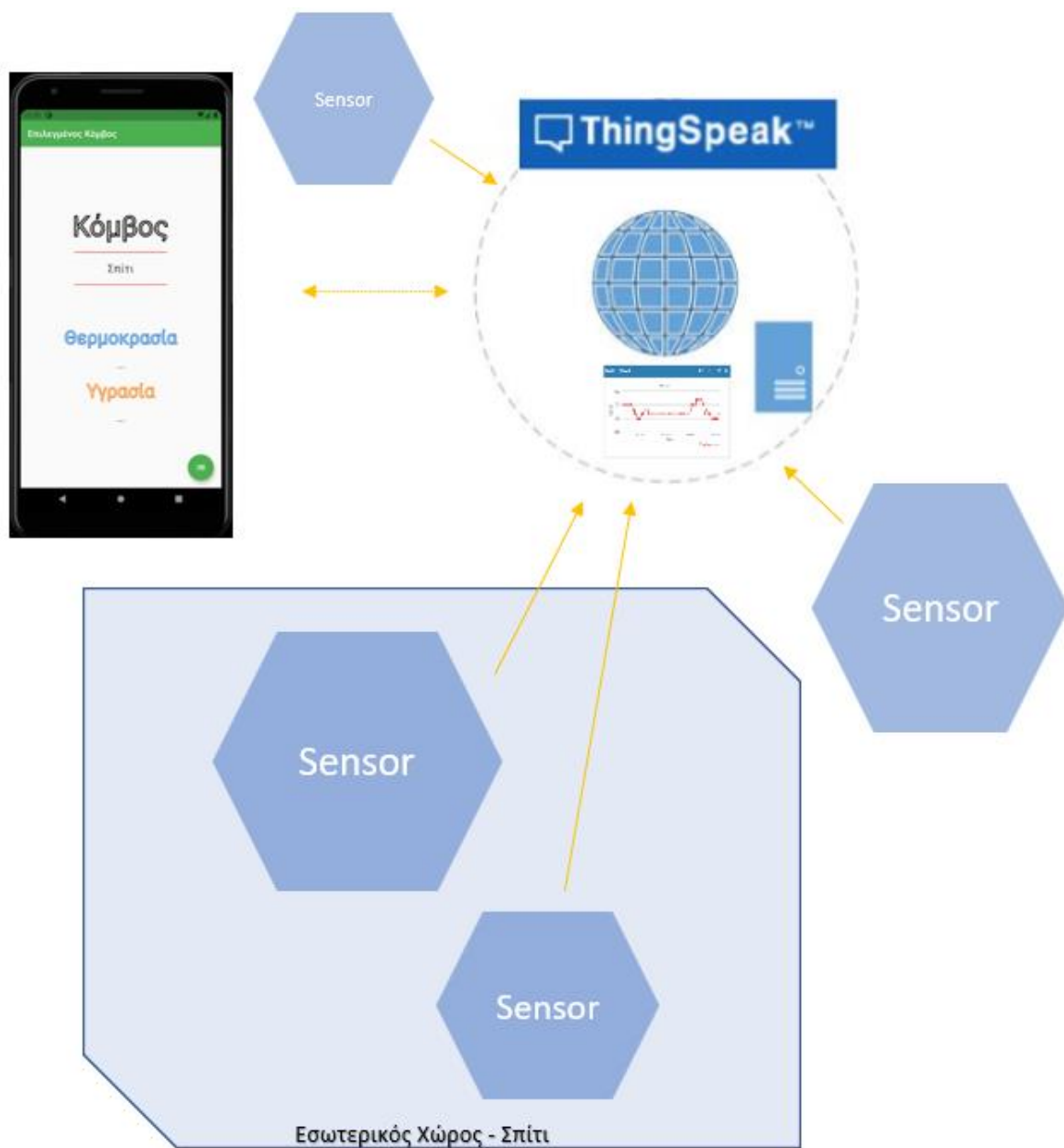


Εικόνα 2.7: Επικοινωνία εφαρμογής με ένα εκτός και με πολλούς κόμβους σε εσωτερικό δίκτυο  
Όποιος κόμβος βρίσκεται μέσα στο ίδιο τοπικό δίκτυο θα πρέπει να πραγματοποιηθεί port forwarding όπως φαίνεται στην Εικόνα 2.7.



Εικόνα 2.8: Επικοινωνία της εφαρμογής με πολλούς κόμβους όταν βρίσκεται εκτός του εσωτερικού δικτύου

Όπως παρουσιάζεται στην Εικόνα 2.8 για να επικοινωνεί η εφαρμογή με όσα esp32 υπάρχουν στον εσωτερικό χώρο από το διαδίκτυο θα πρέπει να το κάνουν μέσω του Thingspeak.



Εικόνα 2.9: Η εφαρμογή μπορεί να παρακολουθεί esp32 εκτός και εντός δικτύου μέσω Thingspeak

Όσα esp32 έχουν σύνδεση στο διαδίκτυο και τη διεύθυνση του server Thingspeak να αποστέλουν τα δεδομένα τους στο Thingspeak..

## Κεφάλαιο 3ο: Οι γλώσσες προγραμματισμού, τα αναπτυξιακά εργαλεία και τα ηλεκτρονικά εξαρτήματα

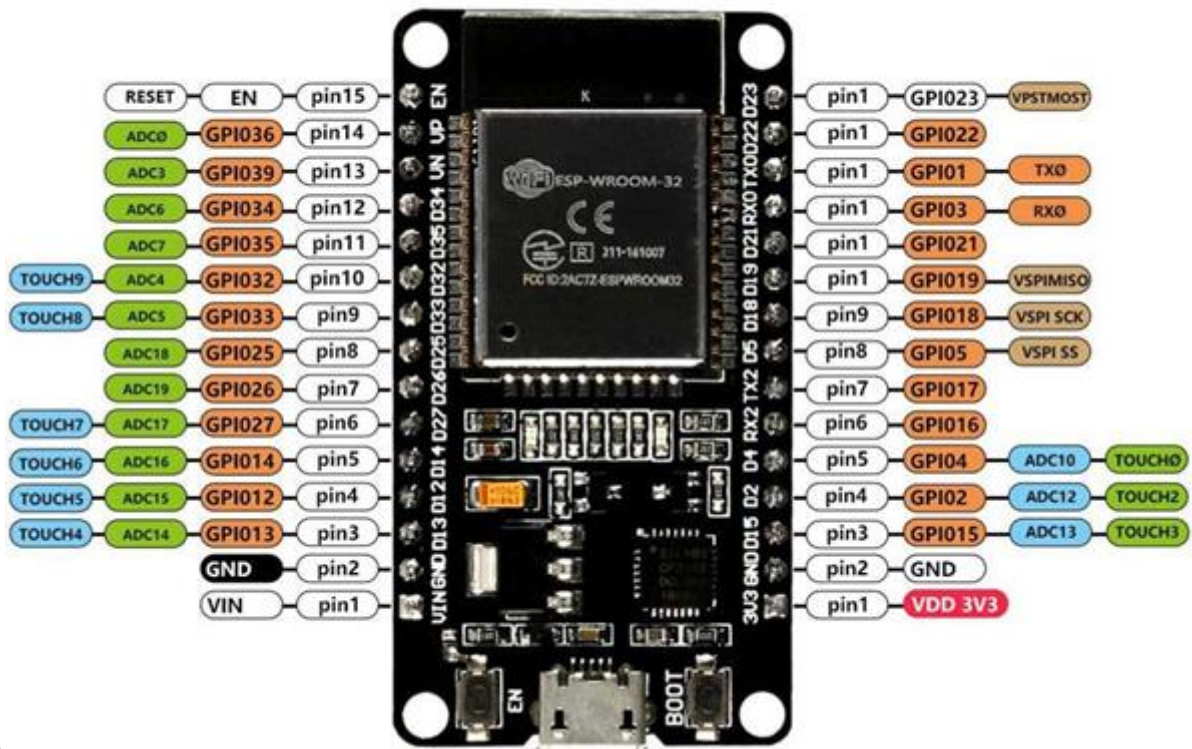
### 3.1 ESP32

Στην παρούσα εργασία θα χρησιμοποιηθεί το ESP32. Το ESP32 είναι μια σειρά συστημάτων χαμηλού κόστους, χαμηλής κατανάλωσης σε μικροελεγκτές chip με ενσωματωμένο Wi-Fi και Bluetooth διπλής λειτουργίας. Η σειρά ESP32 χρησιμοποιεί μικροεπεξεργαστή Tensilica Xtensa LX6 τόσο σε παραλλαγές διπλού πυρήνα (dual core) όσο και σε μονόπλευρα και περιλαμβάνει ενσωματωμένους διακόπτες κεραίας, ενισχυτή ισχύος, ενισχυτή ισχύος χαμηλού θορύβου, φίλτρα και μονάδες διαχείρισης ενέργειας. Το ESP32 δημιουργείται και αναπτύσσεται από την εταιρεία Espressif Systems και κατασκευάζεται από την TSMC χρησιμοποιώντας τη διαδικασία των 40 nm. Είναι διάδοχος του μικροελεγκτή ESP8266. Το ESP32 μπορεί να λειτουργεί αξιόπιστα σε βιομηχανικά περιβάλλοντα, με θερμοκρασία λειτουργίας που κυμαίνεται από  $-40^{\circ}\text{C}$  έως  $+125^{\circ}\text{C}$ . Με τη βοήθεια προηγμένων κυκλωμάτων βαθμονόμησης, το ESP32 μπορεί να απομακρύνει δυναμικά τις ατέλειες των εξωτερικών κυκλωμάτων και να προσαρμοστεί σε αλλαγές στις εξωτερικές συνθήκες. Σχεδιασμένο για κινητές συσκευές, φοριέται για ηλεκτρονικά και εφαρμογές IoT και επιτυγχάνει εξαιρετικά χαμηλή κατανάλωση ενέργειας με ένα συνδυασμό διαφόρων τύπων του ιδιόκτητου λογισμικού. Το ESP32 περιλαμβάνει επίσης state-of-the-art χαρακτηριστικά, όπως clock gating, διάφορες λειτουργίες ισχύος και δυναμική κλιμακωτή ισχύ. Το ESP32 έχει ενσωματωμένα διακόπτες κεραίας, ενισχυτή ισχύος, ενισχυτή χαμηλού θορύβου, φίλτρα και μονάδες διαχείρισης ενέργειας. Το ESP32 προσθέτει ανεκτίμητη λειτουργικότητα και ευελιξία στις εφαρμογές σας με ελάχιστες απαιτήσεις για τυπωμένο κύκλωμα (PCB) και μπορεί να εκτελεστεί ως ένα πλήρες αυτόνομο σύστημα ή ως υποτελής συσκευή σε ένα MCU κεντρικού υπολογιστή, μειώνοντας τις επικοινωνιακές διεπαφές επικοινωνίας στον κύριο επεξεργαστή εφαρμογών. Μπορεί να συνδεθεί με άλλα συστήματα για να παρέχει λειτουργίες Wi-Fi και Bluetooth μέσω των διεπαφών SPI / SDIO ή I2C / UART.

Κάποια από τα χαρακτηριστικά του είναι:

- Processors:
  - CPU: Xtensa dual-core (or single-core) 32-bit LX6 microprocessor, operating at 160 or 240 MHz and performing at up to 600 DMIPS
  - Ultra low power (ULP) co-processor
- Memory: 520 KiB SRAM, 448 KiB ROM

- Wireless connectivity:
  - Wi-Fi: 802.11 b/g/n
  - Bluetooth: v4.2 BR/EDR and BLE (shares the radio with Wi-Fi)
- Peripheral interfaces:
  - 34 × programmable GPIOs
  - 12-bit SAR ADC up to 18 channels
  - 2 × 8-bit DACs
  - 10 × touch sensors (capacitive sensing GPIOs)
  - 4 × SPI
  - 2 × I<sup>2</sup>S interfaces
  - 2 × I<sup>2</sup>C interfaces
  - 3 × UART
  - SD/SDIO/CE-ATA/MMC/eMMC host controller
  - SDIO/SPI slave controller
  - Ethernet MAC interface with dedicated DMA and IEEE 1588 Precision Time Protocol support
  - CAN bus 2.0
  - Infrared remote controller (TX/RX, up to 8 channels)
  - Motor PWM
  - LED PWM (up to 16 channels)
  - Hall effect sensor
  - Ultra low power analog pre-amplifier
- Security:
  - IEEE 802.11 standard security features all supported, including WPA, WPA/WPA2 and WAPI
  - Secure boot
  - Flash encryption
  - 1024-bit OTP, up to 768-bit for customers
  - Cryptographic hardware acceleration: AES, SHA-2, RSA, elliptic curve cryptography (ECC), random number generator (RNG)
- Power management:
  - Internal low-dropout regulator
  - Individual power domain for RTC
  - 5 μA deep sleep current
  - Wake up from GPIO interrupt, timer, ADC measurements, capacitive touch sensor interrupt



Εικόνα 3.1: esp32 με τους ακροδέκτες

Το ESP32 είναι μια σειρά συστημάτων χαμηλού κόστους, χαμηλής κατανάλωσης σε μικροελεγκτές chip με ενσωματωμένο Wi-Fi και Bluetooth διπλής λειτουργίας. Το εύρος τάσης λειτουργίας του ESP32 είναι 2,2V έως 3,6V, η πλακέτα διαθέτει ρυθμιστή τάσης LDO για να διατηρεί σταθερή την τάση στα 3,3V. Η έξοδος του ρυθμιστή βρίσκεται σε μία από τις πλευρές της πλακέτας και είναι 3.3V. Αυτός ο ακροδέκτης μπορεί να χρησιμοποιηθεί για την παροχή τάσης σε εξωτερικά εξαρτήματα.

Ένα από τα μεγαλύτερα χαρακτηριστικά που προσφέρει το ESP32 είναι ότι μπορεί μόνο να συνδεθεί με ένα υπάρχον δίκτυο WiFi και να λειτουργήσει ως Web Server, αλλά μπορεί επίσης να δημιουργήσει ένα δικό του δίκτυο, επιτρέποντας σε άλλες συσκευές να συνδεθούν απευθείας με αυτό και να αποκτήσουν πρόσβαση σε ιστοσελίδες. Αυτό είναι δυνατό επειδή το ESP32 μπορεί να λειτουργήσει σε δυο διαφορετικές λειτουργίες:

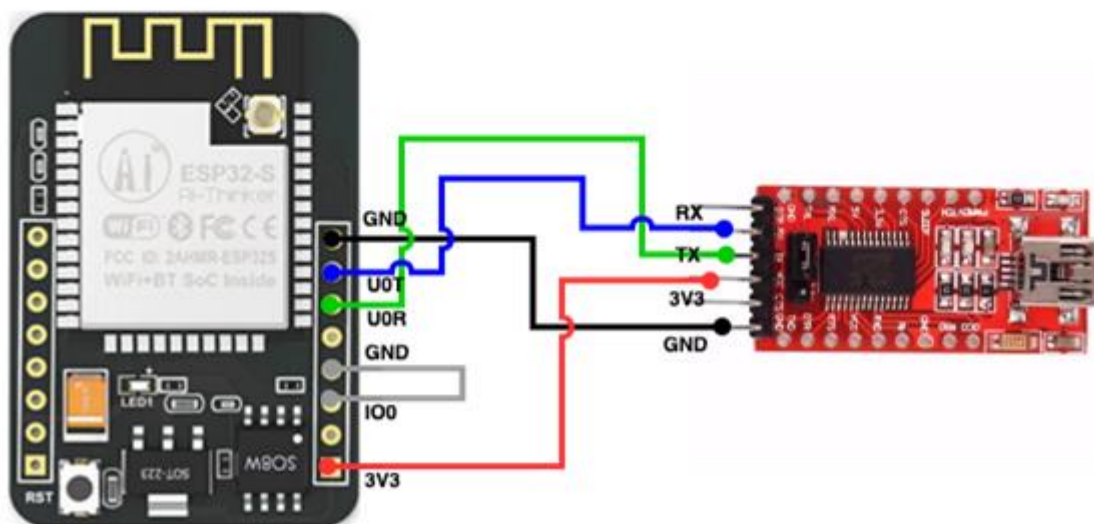
- Station mode
- Soft Access Point mode

### **Station mode (STA)**

Το ESP32 που συνδέεται με ένα υπάρχον WiFi ονομάζεται Station (STA). Στη λειτουργία STA, το ESP32 παίρνει IP από ασύρματο δρομολογητή στον οποίο είναι συνδεδεμένο. Με αυτή τη διεύθυνση IP, μπορεί να δημιουργήσει έναν Web Server και να δημιουργήσει ιστοσελίδες σε όλες τις συνδεδεμένες συσκευές στο υπάρχον δίκτυο WiFi.

### **Access Point (AP)**

Το ESP32 που δημιουργεί το δικό του WiFi δίκτυο και λειτουργεί ως διανομέας (hub) ακριβώς όπως το WiFi για έναν ή περισσότερους σταθμούς ονομάζεται Access Point (AP). Σε αντίθεση με τον δρομολογητή WiFi, δεν έχει διασύνδεση με ένα ενσύρματο δίκτυο. Έτσι, ένας τέτοιος τρόπος λειτουργίας ονομάζεται Soft Access Point (soft-AP). Επίσης ο μέγιστος αριθμός σταθμών που μπορούν να συνδεθούν σε αυτό περιορίζεται σε πέντε.



Εικόνα 3.2: esp32-CAM

[<https://homeassistant.jongriffith.com/Hardware-Sensors-Etc/ESP8266%20Boards/ESP32-CAM/>]

Το esp32 διαθέτει πολλά μοντέλα αλλά αυτό που επικεντρώνει το ενδιαφέρον τα τελευταία χρόνια είναι ένα μοντέλο που περιέχει κάμερα.

Το ESP32-CAM είναι μια πολύ μικρή μονάδα κάμερας με το τσιπ ESP32-S που κοστίζει περίπου 10€. Εκτός από την κάμερα OV2640 και πολλά GPIO για σύνδεση περιφερειακών, διαθέτει επίσης μια υποδοχή κάρτας microSD που μπορεί να είναι χρήσιμη για την αποθήκευση εικόνων που λαμβάνονται με την κάμερα ή για την αποθήκευση αρχείων για εξυπηρέτηση πελατών.

Αυτό το ESP32-CAM δεν συνοδεύεται από υποδοχή USB, επομένως χρειάζεστε έναν προγραμματιστή FTDI για να ανεβάσετε κώδικα μέσω των ακροδεκτών του.

### 3.2 Αισθητήρας DHT11

Για λόγους πειραματισμού στο σύστημα χρησιμοποιήθηκε μόνο ο αισθητήρας μέτρησης θερμοκρασίας-υγρασίας όπου έχει ως βασικό περιφερειακό ψηφιακό αισθητήριο χαμηλής ακρίβειας ώστε να μην απαιτείται η χρήση των μετατροπέων αναλογικού σε ψηφιακό. Τονίζεται ότι δεν χρειάζεται η προσθήκη διαφορετικών αισθητήρων γιατί ο στόχος της εργασίας είναι η επικοινωνία της εφαρμογής στο κινητό και του Thingspeak με το esp32.

Επιλέχθηκε το DHT11 για μέτρηση θερμοκρασία και υγρασίας γιατί:

- Το DHT11 είναι ένας ενσωματωμένος αισθητήρας χαμηλού κόστους, ο οποίος χρησιμοποιείται για τη μέτρηση της θερμοκρασίας (σε κλίμακα από 0 έως 50 βαθμούς Κελσίου με ακρίβεια + -2 C) και υγρασία (σε εύρος από 20% έως 80% με ακρίβεια + -5%).
- Αποτελείται από έναν χωρητικό αισθητήρα υγρασίας που μετρά την υγρασία στον αέρα.
- Για μέτρηση θερμοκρασίας, έχει ενσωματωμένο θερμίστορ, το οποίο είναι μια αντίσταση και γρήγη συσκευή μέτρησης θερμοκρασίας NTC.
- Λειτουργεί με συστήματα μικροελεγκτή 3.3V και 5V.
- Αυτός ο αισθητήρας έχει εξαιρετική ποιότητα, γρήγορο χρόνο απόκρισης και ικανότητα κατά των παρεμβολών.
- Στο DHT11, οι συντελεστές βαθμονόμησης είναι ήδη αποθηκευμένοι στη μνήμη του προγράμματος του και απλώς πρέπει να καλέσουμε αυτούς τους συντελεστές βαθμονόμησης, ενώ οι εσωτερικοί αισθητήρες ανιχνεύουν το σήμα στη διαδικασία.
- Χρησιμοποιεί χαμηλή ισχύ για τη μετάδοση απλών σημάτων έως και 20 μέτρων.
- Διατίθεται σε ένα πακέτο που αποτελείται από 4 ακίδες με απόσταση 0,1" μεταξύ τους και μπορεί να παρέχεται ειδικό πακέτο ανάλογα με τη ζήτηση του χρήστη.
- Είναι αρκετά ακριβές στις μετρήσεις του σε σύγκριση με άλλους ακριβούς αισθητήρες, δηλαδή SHT10, DS18B20 κ.λπ.

Τα βασικά του χαρακτηριστικά είναι:

- Ultra low cost
- 3 to 5V power and I/O
- 2.5mA max current use during conversion (while requesting data)
- Good for 20-80% humidity readings with 5% accuracy
- Good for 0-50°C temperature readings  $\pm 2^{\circ}\text{C}$  accuracy
- No more than 1 Hz sampling rate (once every second)
- Body size 15.5mm x 12mm x 5.5mm
- 4 pins with 0.1" spacing



Εικόνα 3.3: DHT11 Αισθητήρας

### 3.3 Flutter

Το Flutter είναι ένα λογισμικό που χρησιμοποιείται για την ανάπτυξη εφαρμογών για Android, IOS, Windows, Mac, Linux και τον ιστό. Είναι ένα κιτ ανάπτυξης UI ανοιχτού κώδικα από την Google.

Το Flutter είναι μια εφαρμογή για κινητές συσκευές πολλαπλών πλατφορμών που χρησιμοποιεί μια κοινή βάση κώδικα τόσο για Android όσο και για iOS.

Οι εφαρμογές Flutter είναι η ενσωματωμένη γλώσσα προγραμματισμού της Google, Dart. Χρησιμοποιώντας αυτό το πλαίσιο οι προγραμματιστές μπορούν να κάνουν μια εφαρμογή με δημιουργικά σχέδια και γραφικά. Αυτό το πλαίσιο μειώνει την απαιτούμενη ποσότητα κώδικα, επομένως είναι εύκολη σε σύγκριση με άλλα πλαίσια.

Είναι λογισμικό ανοιχτού κώδικα, οπότε είναι ελεύθερο να δημιουργήσει οποιοδήποτε είδος εφαρμογής. Εφόσον συντηρείται και αναπτύσσεται από την Google, μπορείτε να αποκτήσετε εκπληκτικές λειτουργίες για την εφαρμογή σας.

Στο Flutter, όλα είναι widget. Τα Widgets είναι κομμάτια UI που μπορείτε να συνδυάσετε για να δημιουργήσετε μια ολοκληρωμένη εφαρμογή.

Η δημιουργία μιας εφαρμογής Flutter είναι σαν να χτίζετε ένα lego κομμάτι-κομμάτι.

Για προγραμματιστές, το Flutter σημαίνει ταχύτερη και πιο δυναμική ανάπτυξη εφαρμογών για κινητά. Ο προγραμματιστής μπορεί να κάνει αλλαγές στον κώδικα και να τις δει αμέσως στην εφαρμογή. Αυτή είναι η λεγόμενη επαναφόρτωση Hot, η οποία συνήθως διαρκεί μόνο κάτι δευτερόλεπτα και βοηθά τις ομάδες να προσθέσουν λειτουργίες, να διορθώσουν σφάλματα και να πειραματιστούν πιο γρήγορα.

Ένα επιπλέον σημαντικό πλεονέκτημα του flutter είναι ότι έχει μια ισχυρή κοινότητα.

Υπάρχουν πολλά πακέτα από την ομάδα του flutter καθώς και από την ομάδα του προγραμματιστών.

Το Flutter αποτελείται από όμορφα σχέδια UI που είναι φιλικά προς τον χρήστη. Η πολυεπίπεδη αρχιτεκτονική επιτρέπει πλήρη προσαρμογή, η οποία οδηγεί σε απίστευτα γρήγορη απόδοση και εκφραστικά και ευέλικτα σχέδια.

Τα widget του Flutter ενσωματώνουν όλες τις κρίσιμες διαφορές πλατφόρμας, όπως κύλιση, πλοήγηση, εικονίδια και γραμματοσειρές για να παρέχουν πλήρη εγγενή απόδοση σε iOS και Android.

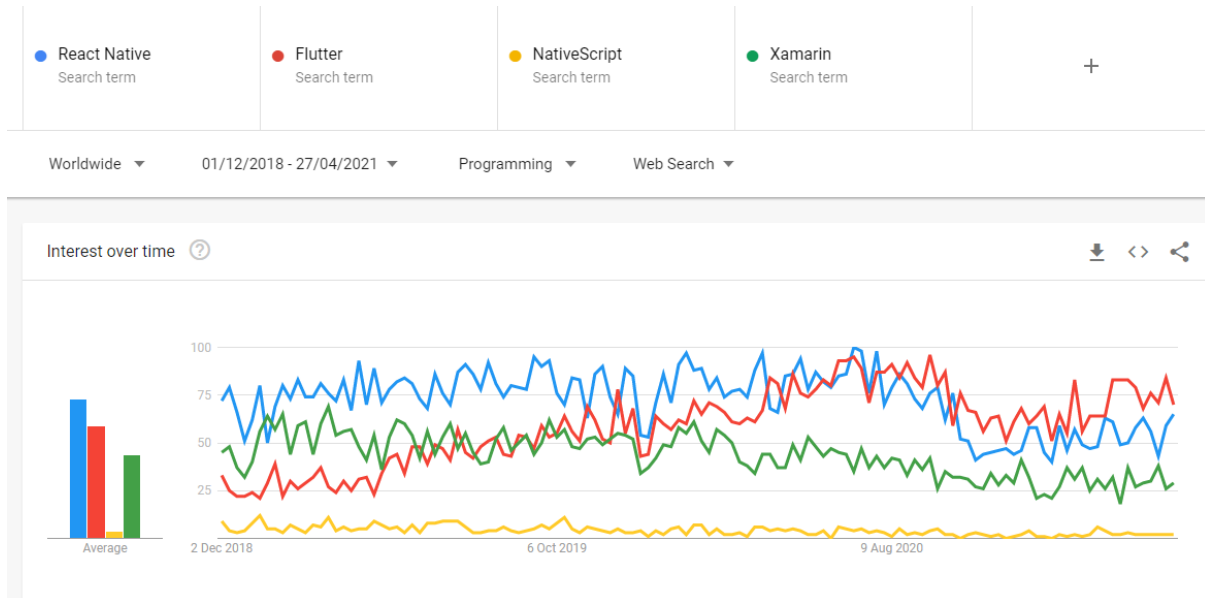
Μετά την κυκλοφορία της άλφα έκδοσης τον Μάιο του 2017 και την επίσημη κυκλοφορία της σταθερής έκδοσης τον Δεκέμβριο του 2018, δεν χρειάστηκε πολύς χρόνος ο Flutter για να κερδίσει απίστευτη δημοτικότητα, κρίνοντας από τον αριθμό των αστεριών στο GitHub (119k έναντι του React Native's 95k). Και συνεχίζει να αυξάνεται, μαρτυρώντας το ενδιαφέρον για το προϊόν.

Τον Μάρτιο του 2021, κυκλοφόρησε η δεύτερη γενιά του Flutter, φέρνοντας περισσότερες αναβαθμίσεις και βελτιώσεις για να προσφέρει καλύτερη εμπειρία για προγραμματιστές και χρήστες.

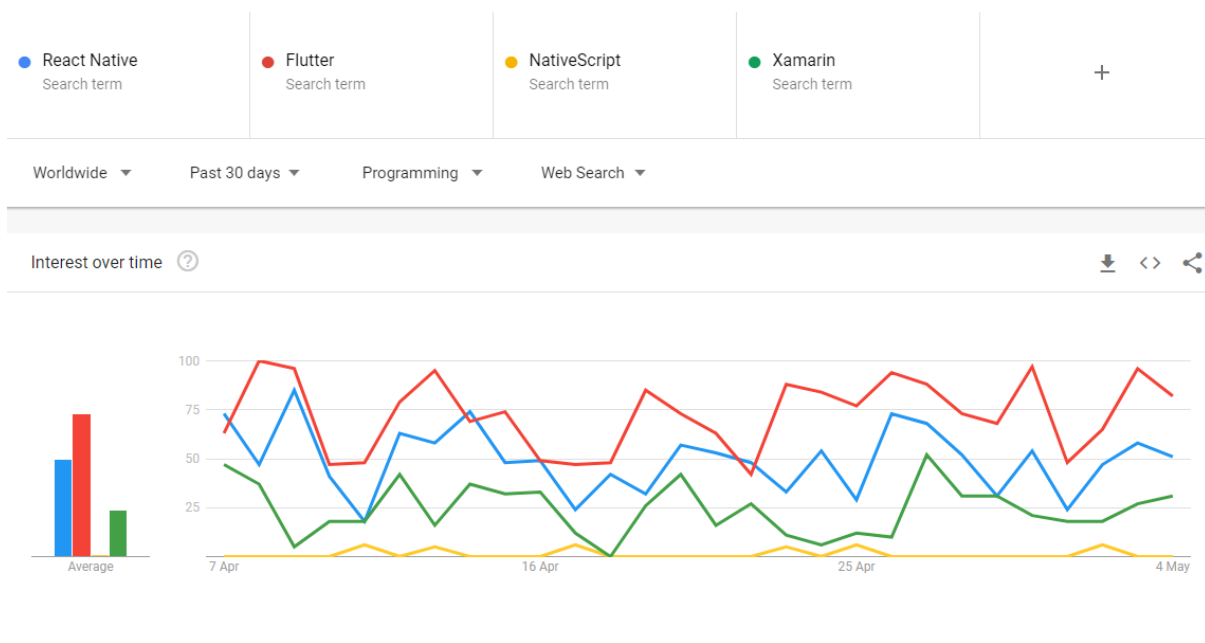
Το Flutter είναι η τεχνολογία ανοιχτού κώδικα της Google για τη δημιουργία εφαρμογών για κινητά, επιτραπέζιους υπολογιστές και ιστούς με μία μόνο βάση κώδικα. Σε αντίθεση με άλλες δημοφιλείς λύσεις, το Flutter δεν είναι πλαίσιο ή βιβλιοθήκη είναι ένα πλήρες kit ανάπτυξης λογισμικού - SDK.

Η βιβλιοθήκη είναι βασικά ένα επαναχρησιμοποιήσιμο κομμάτι κώδικα που εισάγετε στην εφαρμογή σας για να εκτελέσετε μια συγκεκριμένη κοινή λειτουργία.

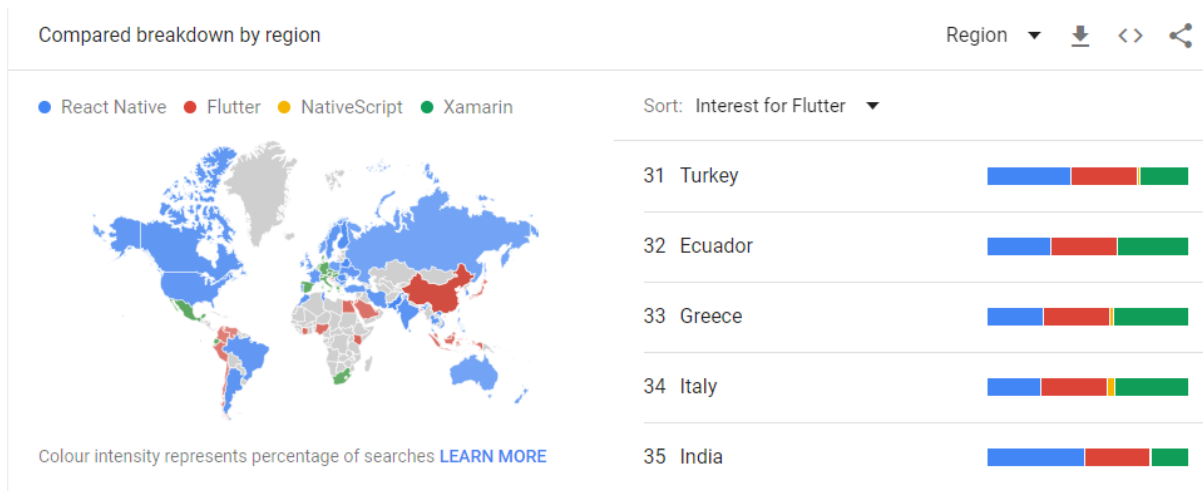
Ένα πλαίσιο είναι μια δομή που σας παρέχει μια αρχιτεκτονική σκελετού για την κατασκευή λογισμικού. Είναι ένα σύνολο εργαλείων που χρησιμεύουν ως θεμέλιο για την εφαρμογή σας, απαιτώντας να συμπληρώσετε τα κενά με τον κωδικό σας για να ολοκληρώσετε ολόκληρη τη δομή και να αποκτήσετε την επιθυμητή λειτουργικότητα.



Εικόνα 3.4: Συγκριτικά αποτελέσματα τεχνολογιών για υλοποίηση εφαρμογών σε κινητά – Περίοδος 2018-2021



Εικόνα 3.5: Συγκριτικά αποτελέσματα τεχνολογιών για υλοποίηση εφαρμογών σε κινητά – Περίοδος Απρίλιος 2021

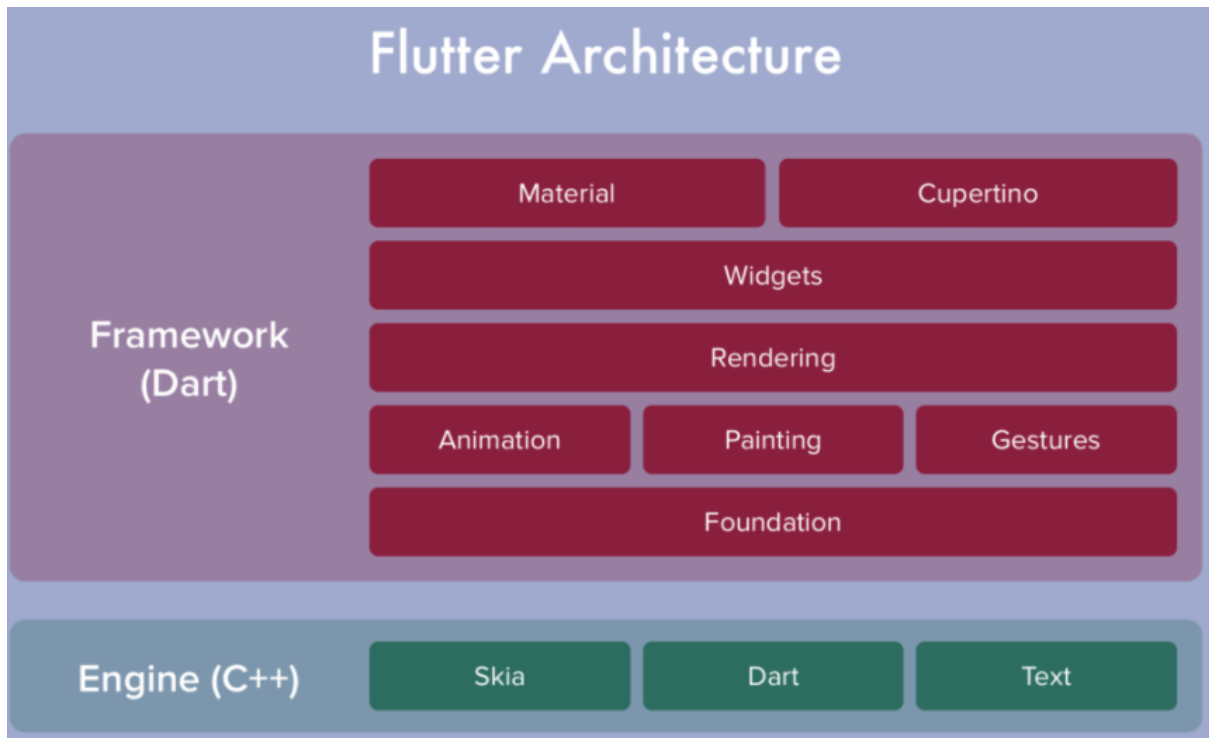


Εικόνα 3.6: Συγκριτικά αποτελέσματα τεχνολογιών για υλοποίηση εφαρμογών σε κινητά – Περιοχή Ελλάδα – Περίοδος 2018-2021

Ένα SDK έχει πολύ ευρύτερο πεδίο, καθώς είναι μια συλλογή εργαλείων, συμπεριλαμβανομένων βιβλιοθηκών, τεκμηρίωσης, API, μερικές φορές πλαισίων και άλλων, παρέχοντας όλα όσα χρειάζεστε για την ανάπτυξη λογισμικού. Και αυτό ισχύει για το Flutter - περιέχει ήδη όλα τα απαραίτητα για τη δημιουργία εφαρμογών πολλαπλών πλατφορμών.

Άλλες τεχνολογίες όπως το Xamarin, το React Native, ή το NativeScript χρησιμοποιούνται επίσης για την ανάπτυξη εφαρμογών που λειτουργούν σε πολλές πλατφόρμες. Έχουμε μια λεπτομερή σύγκριση αυτών των εργαλείων μεταξύ πλατφορμών, όπως φαίνεται στις Εικόνες 3.4, 3.5, 3.6.

Δεδομένου ότι το Flutter είναι ένα πλήρες SDK, περιλαμβάνει μια μηχανή απόδοσης, έτοιμα widget, API δοκιμών και ενοποίησης κ.λπ.



Εικόνα 3.7: Flutter Framework

Τα τρία κύρια αρχιτεκτονικά στρώματα του flutter είναι:

- μια ενσωμάτωση που χρησιμοποιεί μια γλώσσα για συγκεκριμένη πλατφόρμα και κάνει την εφαρμογή να λειτουργεί σε οποιοδήποτε λειτουργικό σύστημα .
- μια μηχανή γραμμένη σε C / C ++ που παρέχει χαμηλού επιπέδου εφαρμογή των βασικών API του Flutter. Αυτό περιλαμβάνει γραφικά (μέσω της βιβλιοθήκης γραφικών Skia 2D), διάταξη κειμένου, αρχείο / δίκτυο I / O, υποστήριξη προσβασιμότητας, αρχιτεκτονική προσθηκών και Dart runtime και compile toolchain. και
- ένα πλαίσιο βασισμένο στη γλώσσα προγραμματισμού Dart. Η εφαρμογή του είναι προαιρετική, αλλά παρέχει ένα πλούσιο σύνολο βιβλιοθηκών που μπορούν να διακριθούν σε επίπεδα: βασικές θεμελιώδεις τάξεις, επίπεδο απόδοσης, επίπεδο widget και βιβλιοθήκες Υλικού / Cupertino.

Η γλώσσα που βρίσκεται στα θεμέλια του Flutter είναι η Dart. Είναι μια βελτιστοποιημένη για πελάτες, αντικειμενοστρεφής γλώσσα προγραμματισμού που αναπτύχθηκε από την Google. Το Dart μπορεί να μεταγλωττιστεί σε εγγενή κώδικα για κινητά και επιτραπέζιους υπολογιστές, καθώς και σε JavaScript. Αυτό που είναι υπέροχο είναι ότι λόγω αυτής της άμεσης συλλογής δεν απαιτεί επιπλέον γέφυρα για επικοινωνία με την πλατφόρμα, όπως για παράδειγμα το React Native. Αυτό βελτιώνει σημαντικά τον χρόνο εκκίνησης και τη συνολική απόδοση της εφαρμογής.

Ένα άλλο κρίσιμο μέρος του Flutter είναι τα widget του. Μέσα στο Flutter SDK, χρησιμεύουν ως δομικά στοιχεία που μπορούν να καλύψουν σχεδόν όλες τις πτυχές της ανάπτυξης. Το Flutter προσφέρει όχι μόνο ένα ευρύ φάσμα έτοιμων widget, αλλά σας επιτρέπει επίσης να προσαρμόσετε τα υπάρχοντα ή να δημιουργήσετε τα δικά σας. Διαβάστε παρακάτω για περισσότερες λεπτομέρειες σχετικά με τα widget Flutter. Το Flutter προσφέρει επίσης ένα αυτοματοποιημένο σύνολο εργαλείων δοκιμών, συγκεκριμένα, για τρεις τύπους δοκιμών: δοκιμή μονάδας, δοκιμή widget και δοκιμή ενδοποίησης. Η Google παρέχει μαθήματα για αυτές τις δοκιμές. Επιπλέον, το Flutter υποστηρίζει το μοντέλο Continuous Delivery μέσω fastlane, μια δωρεάν πλατφόρμα που γεφυρώνει το Flutter με δημοφιλή εργαλεία όπως το Gitlab.

Ο εντοπισμός σφαλμάτων στο Flutter πραγματοποιείται με τη βοήθεια του Flutter DevTools. Χρησιμοποιούνται για τον έλεγχο της διάταξης, την ανάλυση της απόδοσης, την εφαρμογή εντοπισμού σφαλμάτων κ.λπ.

Αρχικά, το Flutter σχεδιάστηκε ως SDK εστιασμένο σε κινητά για τη δημιουργία εγγενών εφαρμογών Android και iOS με μία βάση κώδικα. Ωστόσο, η Google έχει εργαστεί πολύ για να αναπτύξει και να το βελτιώσει και με την τελευταία σημαντική αναβάθμισή τους ήρθε η μακροχρόνια δυνατότητα δημιουργίας εφαρμογών για σχεδόν οποιαδήποτε πλατφόρμα.

Με το Flutter 2, μπορείτε να χρησιμοποιήσετε την ίδια βάση κώδικα για την αποστολή εγγενών εφαρμογών σε πέντε λειτουργικά συστήματα:

- iOS,
- Android,
- Windows,
- macOS και
- Linux

καθώς και εμπειρίες ιστού που στοχεύουν προγράμματα περιήγησης

όπως

- το Chrome,
- το Firefox,
- το Safari ή
- το Edge.

Το Flutter μπορεί ακόμη και να ενσωματωθεί σε αυτοκίνητα, τηλεοράσεις και έξυπνες οικιακές συσκευές, παρέχοντας την πιο διαδεδομένη και φορητή εμπειρία για έναν περιβάλλοντα περιβάλλοντα υπολογιστών.

Την τελευταία χρονιά η Google ξεκίνησε συνεργασία της με τη Microsoft, την Toyota και την Canonical με στόχο να φέρει το Flutter σε επιτραπέζιους υπολογιστές, πτυσσόμενα και ενσωματωμένες συσκευές.

Για να επιδείξουν αυτήν την προσαρμοστικότητα, δημιούργησαν μια εφαρμογή Flutter Folio που τρέχει σε πολλές πλατφόρμες από μία βάση κώδικα.

### 3.4 MicroPython

Η MicroPython είναι μια εκ νέου υλοποίηση της γλώσσας προγραμματισμού Python που στοχεύει σε μικροελεγκτές και ενσωματωμένα συστήματα όπως το ESP32 ή το ESP8266.

Ο προγραμματισμός στην MicroPython μοιάζει πολύ με τον προγραμματισμό με την Python: όλα τα γλωσσικά χαρακτηριστικά της Python βρίσκονται επίσης στη MicroPython, εκτός από μερικές εξαιρέσεις. Επειδή οι μικροελεγκτές και τα ενσωματωμένα συστήματα είναι πολύ πιο περιορισμένα από τους υπολογιστές, το MicroPython δεν συνοδεύεται από την πλήρη τυπική βιβλιοθήκη από προεπιλογή.



Εικόνα 3.8: MicroPython χρησιμοποιείται ευρέως για esp32 και esp8266

Εάν κάποιος ήδη γνωρίζει πώς να προγραμματίζει στην Python, ο προγραμματισμός στην MicroPython είναι ο ίδιος. Απλώς πρέπει να γνωρίζει ότι η MicroPython χρησιμοποιείται για περιορισμένες συσκευές. Επομένως, πρέπει να διατηρηθεί ο κώδικας όσο πιο απλός γίνεται.

Παρακάτω παρουσιάζονται τα βασικά της σύνταξης γλώσσας προγραμματισμού Python που ισχύουν επίσης για το MicroPython, όπως:

Mathematical operators

Relational operators

Data types

print() function

Conditional statements

While and for loops

User defined functions

Classes and objects

Modules

Η MicroPython είναι μια εκ νέου υλοποίηση του Python 3 που στοχεύει σε μικροελεγκτές και ενσωματωμένα συστήματα.

### **Python εναντίον MicroPython**

Εκτός από μερικές εξαιρέσεις, τα γλωσσικά χαρακτηριστικά της Python είναι επίσης διαθέσιμα στην MicroPython. Η μεγαλύτερη διαφορά μεταξύ Python και MicroPython είναι ότι η MicroPython σχεδιάστηκε για να λειτουργεί υπό περιορισμένες συνθήκες.

Εξαιτίας αυτού, η MicroPython δεν συνοδεύεται από την πλήρη τυπική βιβλιοθήκη. Περιλαμβάνει μόνο ένα μικρό υποσύνολο της τυπικής βιβλιοθήκης Python. Ωστόσο, περιλαμβάνει λειτουργικές μονάδες για πρόσβαση σε υλικό χαμηλού επιπέδου - αυτό σημαίνει ότι υπάρχουν βιβλιοθήκες για εύκολη πρόσβαση και αλληλεπίδραση με τους GPIO.

Επιπλέον, συσκευές με δυνατότητες Wi-Fi όπως το ESP8266 και το ESP32 περιλαμβάνουν λειτουργικές μονάδες για την υποστήριξη συνδέσεων δικτύου.

### **Γιατί MicroPython;**

Η Python είναι μια από τις πιο ευρέως χρησιμοποιούμενες, απλές και εύχρηστες γλώσσες προγραμματισμού. Έτσι, η εμφάνιση του MicroPython καθιστά εξαιρετικά εύκολο και απλό τον προγραμματισμό ψηφιακών ηλεκτρονικών. Εάν κάποιος δεν έχει προγραμματίσει ποτέ ψηφιακά ηλεκτρονικά, το MicroPython είναι ένα καλό σημείο εκκίνησης.

Ο στόχος της MicroPython είναι να κάνει τον απλό προγραμματισμό ψηφιακών ηλεκτρονικών όσο το δυνατόν πιο απλό, ώστε να μπορεί να χρησιμοποιηθεί από οποιονδήποτε. Επί του παρόντος, η MicroPython χρησιμοποιείται από ερασιτέχνες, ερευνητές, καθηγητές, εκπαιδευτικούς, ακόμη και σε εμπορικά προϊόντα.

Η MicroPython λειτουργεί σε πολλές διαφορετικές συσκευές όπως:

ESP32

ESP8266

PyBoard

Micro: Bit

Adafruit Circuit Playground Express

WiPy - Pycom

Το ESP32 είναι ο διάδοχος του ESP8266. Έτσι, προς το παρόν, δεν είναι διαθέσιμες όλες οι δυνατότητες στη MicroPython για να τις αξιοποιήσετε στο έπακρο στο ESP32

Οι πλακέτες ESP32 και ESP8266 είναι παρόμοιες και δεν υπάρχει σχεδόν καμία διαφορά προγραμματισμού τους χρησιμοποιώντας την MicroPython. Αυτό σημαίνει ότι οτιδήποτε γραφεί για το ESP8266 θα πρέπει επίσης να εκτελείται χωρίς αλλαγές ή ελάχιστες αλλαγές στο ESP32 (αλλάζοντας κυρίως την αντιστοίχιση pin).

### **3.5 Thingspeak**

Το Διαδίκτυο των πραγμάτων, ή το IoT, είναι ένα σύστημα αλληλοσυνδεδεμένων υπολογιστικών συσκευών, μηχανικών και ψηφιακών μηχανών, αντικειμένων, ζώων ή ατόμων που διαθέτουν μοναδικά αναγνωριστικά και τη δυνατότητα μεταφοράς δεδομένων μέσω ενός δικτύου χωρίς να απαιτείται από άνθρωπο σε άνθρωπο ή ανθρώπου με υπολογιστή αλληλεπίδραση.

Ένα πράγμα στο Διαδίκτυο των πραγμάτων μπορεί να είναι ένα άτομο με εμφύτευμα καρδιακής παρακολούθησης, ένα αυτοκίνητο που διαθέτει ενσωματωμένους αισθητήρες για να ειδοποιεί τον οδηγό όταν η πίεση των ελαστικών είναι χαμηλή ή οποιοδήποτε άλλο φυσικό ή τεχνητό αντικείμενο στο οποίο μπορεί να εκχωρηθεί διεύθυνση πρωτοκόλλου διαδικτύου (IP) και μπορεί να μεταφέρει δεδομένα μέσω δικτύου.

Όλο και περισσότερο, οργανισμοί χρησιμοποιούν το IoT για να λειτουργούν πιο αποτελεσματικά και για να προσφέρουν βελτιωμένη εξυπηρέτηση πελατών, να βελτιώνουν τη λήψη αποφάσεων και να αυξάνουν την αξία της επιχείρησης.

Το Διαδίκτυο των πραγμάτων βοηθά τους ανθρώπους να ζουν και να εργάζονται πιο έξυπνα, καθώς και να αποκτήσουν πλήρη έλεγχο της ζωής τους. Εκτός από την προσφορά έξυπνων συσκευών για αυτοματοποίηση σπιτιών, το IoT είναι απαραίτητο για τις επιχειρήσεις. Το IoT παρέχει στις επιχειρήσεις μια ματιά σε πραγματικό χρόνο για το πώς λειτουργούν πραγματικά τα συστήματά τους, παρέχοντας πληροφορίες για τα πάντα, από την απόδοση των μηχανημάτων έως τις λειτουργίες αλυσίδας εφοδιασμού και εφοδιαστικής.

Το IoT επιτρέπει στις εταιρείες να αυτοματοποιήσουν τις διαδικασίες και να μειώσουν το κόστος εργασίας. Περιορίζει επίσης τα απόβλητα και βελτιώνει την παροχή υπηρεσιών, καθιστώντας λιγότερο δαπανηρή την κατασκευή και την παράδοση αγαθών, καθώς και τη διαφάνεια στις συναλλαγές των πελατών.

Ως εκ τούτου, το IoT είναι μια από τις πιο σημαντικές τεχνολογίες της καθημερινής ζωής και θα συνεχίσει να αυξάνει την χρήση του καθώς περισσότερες επιχειρήσεις συνειδητοποιούν τις δυνατότητες των συνδεδεμένων συσκευών για να γίνουν ανταγωνιστικές.

Μια πλατφόρμα εφαρμογών IoT που μπορεί να παρέχει μια ευρεία ποικιλία δυνατοτήτων, παρακολούθησης, ανάλυσης και ενεργειών είναι το ThingSpeak.

## **ThingSpeak**

Το ThingSpeak είναι μια πλατφόρμα που προσφέρει διάφορες υπηρεσίες που βοηθούν στην κατασκευή εφαρμογών IoT. Παρέχει δυνατότητες συλλογής δεδομένων σε πραγματικό χρόνο, δίνοντας την οπτική δυνατότητα για το χρήστη ώστε να βλέπει τα δεδομένα που σύλλεξε με τη μορφή γραφημάτων. Επίσης, του δίνει τη δυνατότητα να δημιουργήσει προσθήκες και εφαρμογές για λειτουργία μαζί με υπηρεσίες ιστού, κοινωνικά δίκτυα και άλλα τρίτου τύπου εφαρμογές.

Το κύριο στοιχείο του ThingSpeak είναι ένα Κανάλι - Channel. Ένα κανάλι αποθηκεύει τα δεδομένα που στέλνουμε στο ThingSpeak και έχει τα παρακάτω στοιχεία:

- 8 πεδία για την αποθήκευση δεδομένων αριθμών ή χαρακτήρων - Αυτά μπορούν να χρησιμοποιηθούν για την αποθήκευση των δεδομένων από έναν αισθητήρα.
- 3 πεδία τοποθεσίας - Χρησιμοποιούνται για την αποθήκευση του πλάτους, του μήκους και του υψομέτρου.
- 1 πεδίο κατάστασης - Ένα πεδίο για την περιγραφή κατάσταση των δεδομένων που είναι αποθηκευμένα στο συγκεκριμένο κανάλι.

Στη συνέχεια θα περιγράψει η δημιουργία και η χρήση ενός λογαριασμού στο ThingSpeak.

Κατά τη μετάβαση στο <https://thingspeak.com/login?skipSSOCheck=true> κάνουμε κλικ στο κουμπί «Create One» και τις απαραίτητες λεπτομέρειες και δημιουργούμε τον λογαριασμό μας.

The image shows a screenshot of the ThingSpeak website's login page on the left and a system architecture diagram on the right. The website header includes the ThingSpeak logo, navigation links for Channels, Apps, and Support, and options for Commercial Use and How to Buy. The main content area contains text about free and commercial accounts and a link to paid license options. The login form includes an email input field, a 'Next' button, and a 'Create one!' link. The diagram illustrates the data flow: 'SMART CONNECTED DEVICES' send data to a cloud labeled 'DATA AGGREGATION AND ANALYTICS ThingSpeak™'. This cloud then feeds into a 'MATLAB' environment for 'ALGORITHM DEVELOPMENT SENSOR ANALYTICS'.

Εικόνα 3.9: Δημιουργία λογαριασμού

## Create MathWorks Account

### Email Address

mimikos@gmail.com ✓

**i** To access your organization's MATLAB license, use your school or work email.

### Location

Zimbabwe ▼

### First Name

Stamatis ✓

### Last Name

Mimikos ✓

Continue

Cancel

Εικόνα 3.10: Στοιχεία Χρήστη - Thingspeak

Στη συνέχεια ο χρήστης μπορεί να δημιουργήσει ένα νέο κανάλι

## My Channels

New Channel

Search by tag



Name	Created	Updated
mimikoschannel	2020-10-08	2021-04-28 12:08
<a href="#">Private</a> <a href="#">Public</a> <a href="#">Settings</a> <a href="#">Sharing</a> <a href="#">API Keys</a> <a href="#">Data Import / Export</a>		

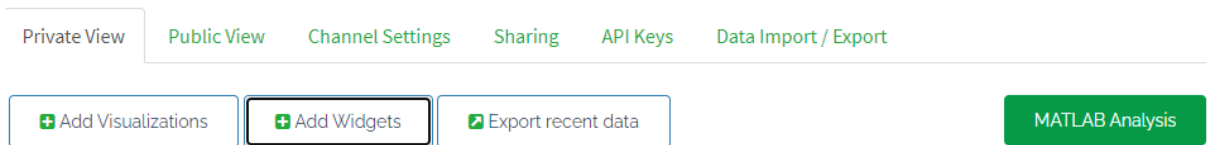
Εικόνα 3.11: Δημιουργία Καναλιού και Ρυθμίσεις

Μπορεί να βάλει μέχρι και 8 πεδία δηλαδή 8 αισθητήρες.

<b>Name</b>	<input type="text" value="mimikoschannel"/>
<b>Description</b>	<input type="text"/>
<b>Field 1</b>	<input type="text" value="sensorTemp"/> <input checked="" type="checkbox"/>
<b>Field 2</b>	<input type="text" value="sensorHum"/> <input checked="" type="checkbox"/>
<b>Field 3</b>	<input type="text"/> <input type="checkbox"/>
<b>Field 4</b>	<input type="text"/> <input type="checkbox"/>
<b>Field 5</b>	<input type="text"/> <input type="checkbox"/>
<b>Field 6</b>	<input type="text"/> <input type="checkbox"/>
<b>Field 7</b>	<input type="text"/> <input type="checkbox"/>
<b>Field 8</b>	<input type="text"/> <input type="checkbox"/>
<b>Metadata</b>	<input type="text"/>
<b>Tags</b>	<input type="text"/>

Εικόνα 3.12: Χαρακτηριστικά Καναλιού Thingspeak

Access: Private

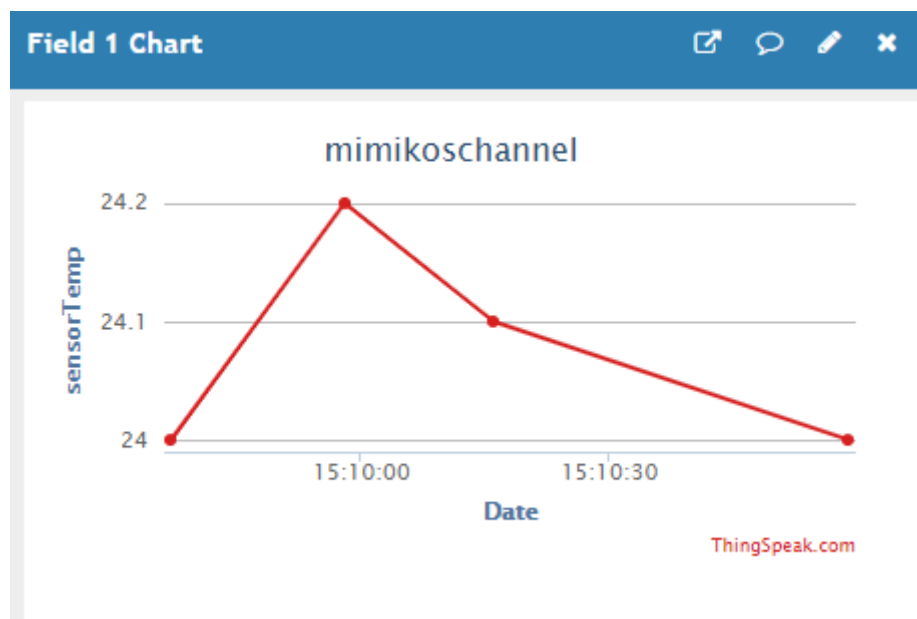


Εικόνα 3.13: Thingspeak – Προσθήκη πρόσθετων

Μπορεί να βάλει διάφορα πρόσθετα-widgets για την αναπαράσταση τις τιμές των πεδίων όπως φαίνεται στις Εικόνες 3.13 και 3.14.

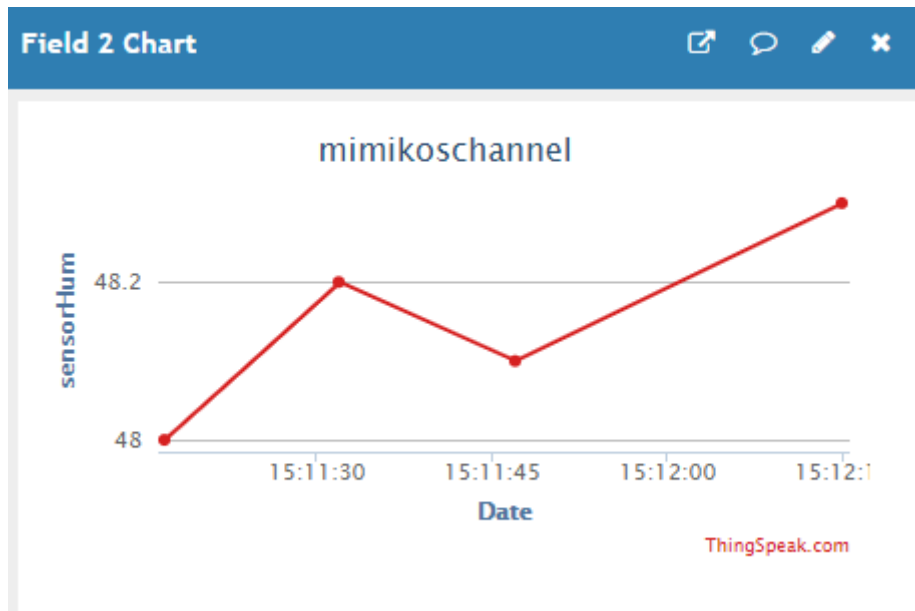


Εικόνα 3.14: Πρόσθετα-Widgets



Εικόνα 3.15: Εμφάνιση τιμών για τον αισθητήρα Θερμοκρασίας

Στις Εικόνες 3.15 και 3.16 παρουσιάζονται τα γραφήματα των τιμών των αισθητήρων σε σχέση με τον χρόνο που καταγράφηκαν.



Εικόνα 3.16: Εμφάνιση τιμών για τον αισθητήρα Υγρασίας

## Write API Key

Key

Generate New Write API Key

## Read API Keys

Key

Note

Save Note

Delete API Key

Add New Read API Key

Εικόνα 3.17: Τα κλειδιά του API για εγγραφή και ανάγνωση αισθητήρων από το κανάλι

Στην Εικόνα 3.17 παρουσιάζεται η σελίδα που θα μπει ο χρήστης ενός καναλιού και να ρυθμίσει τα api keys που απαιτούνται για να κάνει αποστολή και λήψη των τιμών των πεδίων του καναλιού.

Παρακάτω παρουσιάζεται ο κώδικας για αποστολή και λήψη των τιμών:

Να στείλουμε δεδομένα σε ένα αισθητήρα field1

στο κανάλι channel1 με Write API key: channel1writekey

Καλούμε την:

GET

[https://api.thingspeak.com/update?api\\_key=channel1writekey&field1=24](https://api.thingspeak.com/update?api_key=channel1writekey&field1=24)

Ή δύο αισθητήρες

GET

[https://api.thingspeak.com/update?api\\_key=channel1writekey&field1=24&field2=47](https://api.thingspeak.com/update?api_key=channel1writekey&field1=24&field2=47)

Να λάβουμε δεδομένα για έναν αισθητήρα field1

στο κανάλι channel1 με Read API key: channel1readkey

Καλούμε την:

GET

[https://api.thingspeak.com/channels/channel1/fields/1.json?api\\_key=channel1readkey&results=1](https://api.thingspeak.com/channels/channel1/fields/1.json?api_key=channel1readkey&results=1)

## Κεφάλαιο 4ο: Οι εφαρμογές στο κινητό και στο esp32

### 4.1 Εισαγωγή - Πλοήγηση

Στο κεφάλαιο αυτό θα παρουσιαστεί το σύστημα που υλοποιήθηκε όσον αφορά τα προγράμματα στο esp32 και η εφαρμογή στο κινητό με διαγράμματα.



Εικόνα 4.1: Κεντρική Οθόνη με τιμές

Όπως φαίνεται στην εικόνα 4.1 παρουσιάζεται η πρώτη οθόνη όταν ο χρήστης την ανοίγει την εφαρμογή στο κινητό. Στη οθόνη αυτή μπορούμε να διακρίνουμε το όνομα που έχει ο αισθητήρας-κόμβος και οι τιμές υγρασίας και θερμοκρασίας στον συγκεκριμένο κόμβο.



Εικόνα 4.2: Οθόνη για επιλογή κόμβου

Στις εικόνες 4.2 και 4.3 παρουσιάζονται οι οθόνες όταν ο χρήστης στην κεντρική οθόνη πατήσει το κουμπί κάτω δεξιά. Στην οθόνη αυτή μπορεί να διαλέξει κάποιον κόμβο που υπάρχει στη λίστα.

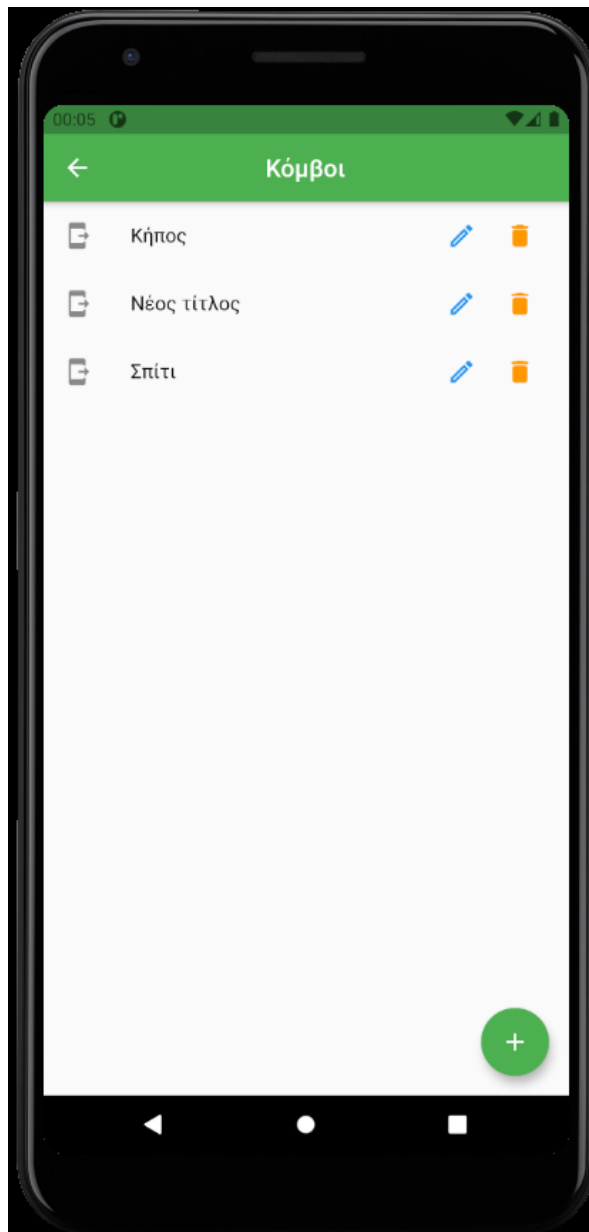
Οι επιλογές που έχει ο χρήστης είναι:

- ✓ Δημιουργία νέου κόμβου
- ✓ Διαγραφή κόμβου
- ✓ Επεξεργασία στοιχείων κόμβου

- Επιλογή αποθηκευμένου κόμβου για προβολή τιμών στην πρώτη κεντρική



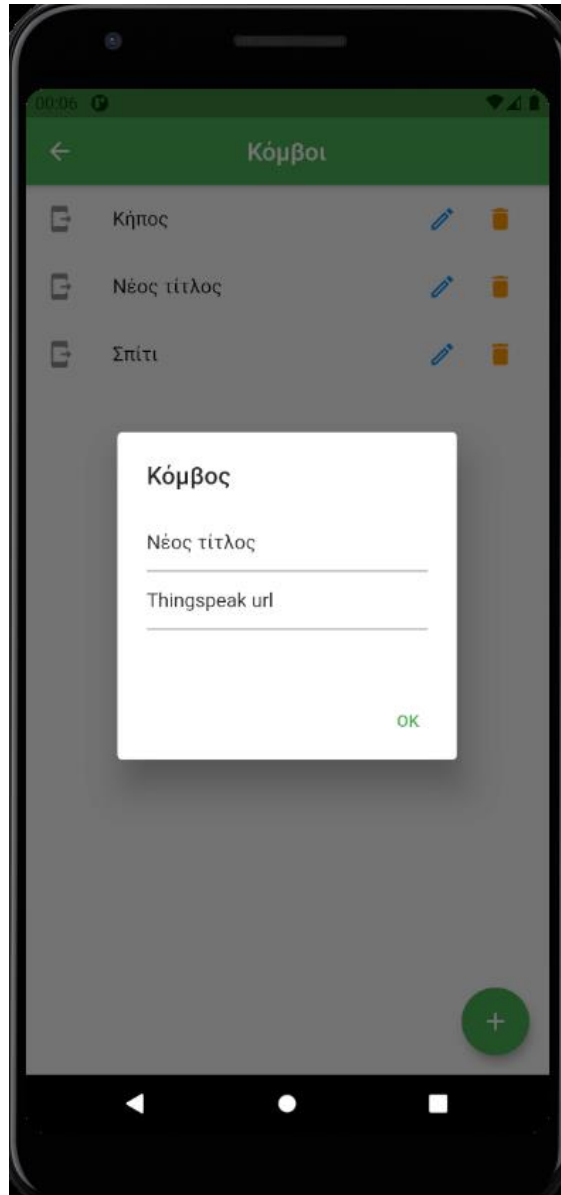
Εικόνα 4.3: Οθόνη για τις επιλογές στη διαχείριση κόμβων



Εικόνα 4.4: Δημιουργία νέου κόμβου

Όταν ο χρήστης πατήσει το σταυρό κάτω δεξιά επιλέγει την προσθήκη νέου κόμβου. (Εικόνα 4.4)

Δημιουργείται νέος κόμβος με ένα όνομα και ο χρήστης έχει τη δυνατότητα να το αλλάξει και να κάνει επεξεργασία των στοιχείων του.



Εικόνα 4.5: Επεξεργασία κόμβου

Όταν ο χρήστης επιλέξει έναν κόμβο μπορεί να τροποποιήσει τα στοιχεία του κόμβου, όπως Τίτλο και Διεύθυνση, όπως φαίνεται στην εικόνα 4.5



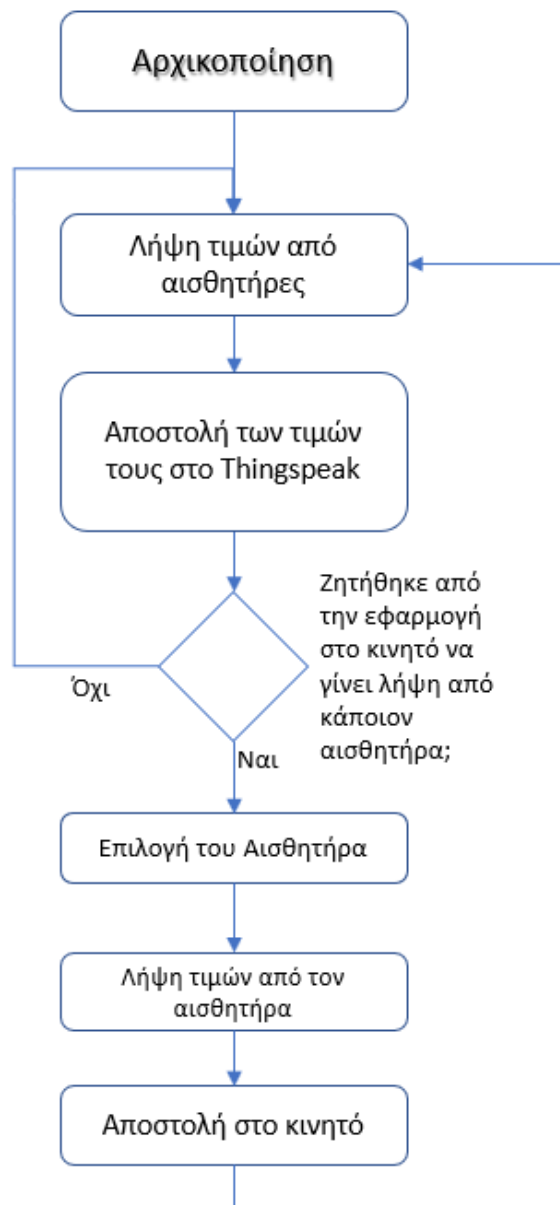
Εικόνα 4.6: Οθόνη με τιμές από τον αισθητήρα όταν δεν ανταποκρίνεται

Όταν ο χρήστης επιλέξει έναν κόμβο και προσπαθεί να επικοινωνήσει με το esp32 και αυτό δεν ανταποκρίνεται για λόγους κακής σύνδεσης ή λειτουργίας τότε στις θέσεις των τιμών εμφανίζονται τελίτσες

## 4.2 Επεξήγηση και διαγράμματα των εφαρμογών

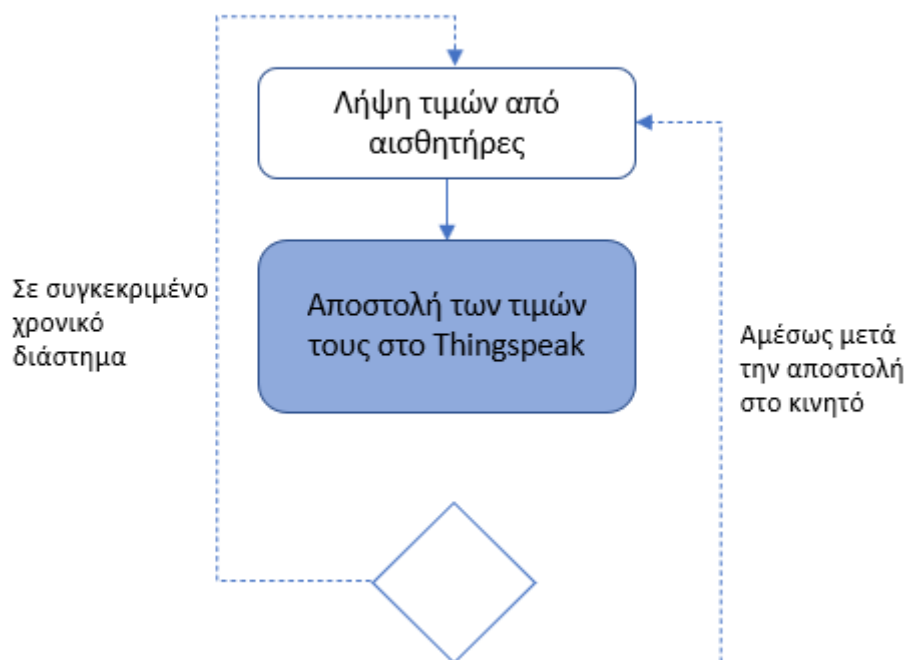
### 4.2.1 Το πρόγραμμα στο esp32

Στο esp32 έχει υλοποιηθεί ένα πρόγραμμα με δύο βασικές λειτουργίες σε micropython για την εκτέλεση τριών βασικών λειτουργιών: την επικοινωνία και την ανάγνωση τιμών από τον αισθητήρα, την λήψη σήματος από την εφαρμογή από το κινητό για την αποστολή των δεδομένων του αισθητήρα και την αποστολή των δεδομένων του αισθητήρα στο Thingspeak.



Εικόνα 4.7: Esp32 – Συνολική Εφαρμογή

Στην εικόνα 4.7 παρουσιάζεται το διάγραμμα λειτουργίας για τη συνολική εφαρμογή. Αρχικά γίνεται η αρχικοποίηση, στη συνέχεια γίνεται λήψη των τιμών των αισθητήρων και αποστολή των τιμών στο Thingspeak ανα χρονικό διάστημα.



Εικόνα 4.8: Esp32 – Διάγραμμα ροής για το αποστολή τιμών στο Thingspeak

Για να αποστείλει το esp32 τιμές στο ThingSpeak πρέπει να καλέσει το πρόγραμμα ένα συγκεκριμένο url σε όπως το παρακάτω:

```
https://api.thingspeak.com/update.json?api_key=CHANNELWRITEKEY&field1=24&field2=47
```

Παρακάτω παρουσιάζεται ο κώδικας που υλοποιεί αυτή τη διαδικασία.

```
def getDHT():  
    d = dht.DHT11(machine.Pin(DHT11_PIN))  
    d.measure()  
    t = d.temperature()
```

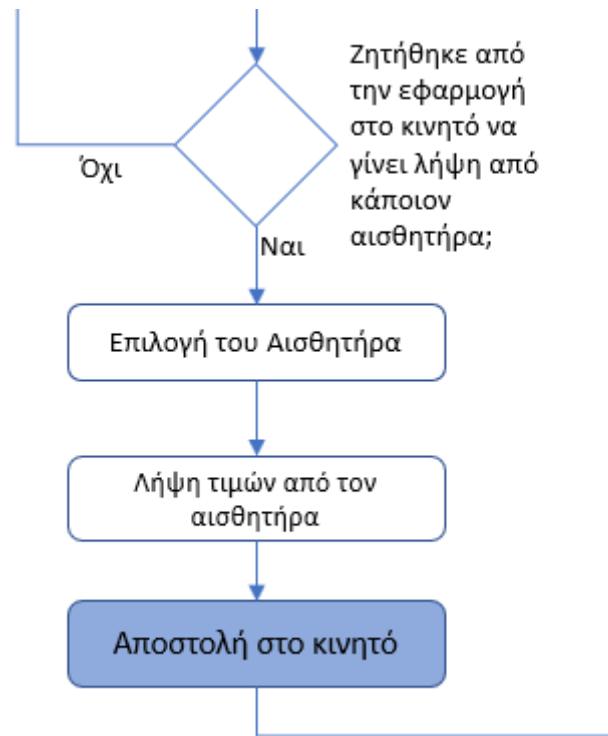
```
h = d.humidity()

print('temperature = %.2f % t)
print('humidity = %.2f % h)

global WRITE_KEY
if not WRITE_KEY:
    print('Problem with key')
    return

s = socket.socket()
ai = socket.getaddrinfo(API_HOST, API_PORT)

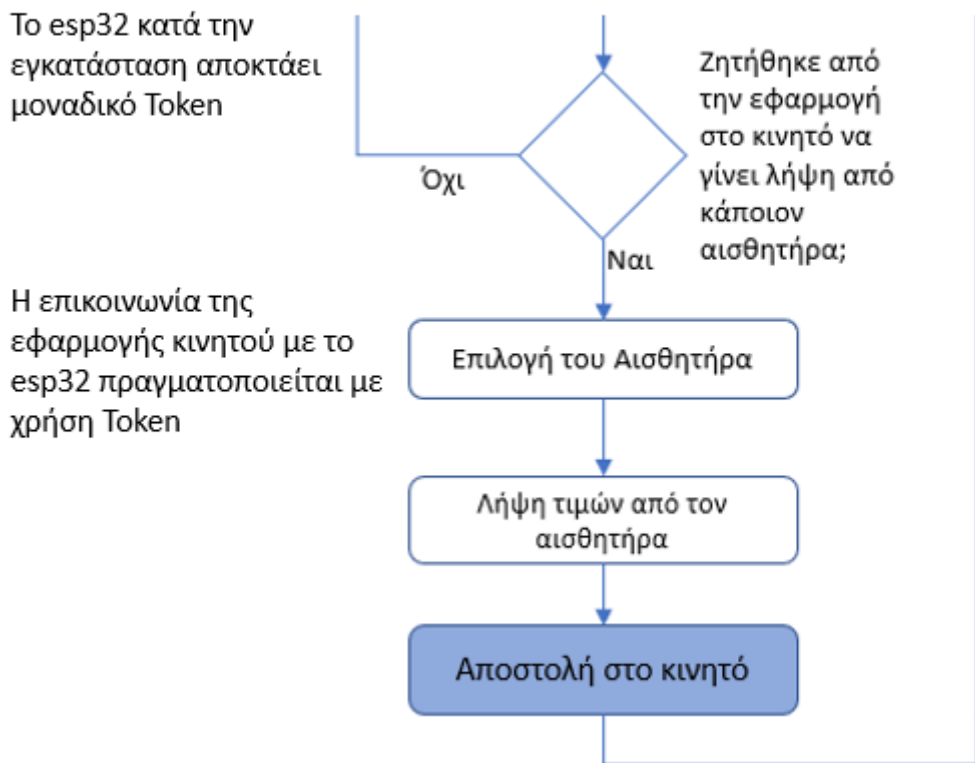
addr = ai[0][-1]
s.connect(addr)
s = ssl.wrap_socket(s)
data = 'field1=%.2f&field2=%.2f % (t, h)
http_data = POST_TEMPLATE % (WRITE_KEY, len(data), data)
s.write(http_data.encode())
s.close()
```



Εικόνα 4.9: Esp32 – Διάγραμμα ροής για το αποστολή τιμών στο Κινητό αφού απαιτήθηκε

Στην εικόνα 4.9 παρουσιάζεται το διάγραμμα ροής για τη δεύτερη λειτουργία. Γίνεται συνέχεια έλεγχος αν απαιτήθηκε ανάγνωση τιμών από το κινητό. Αν ναι τότε πραγματοποιείται η ανάγνωση των τιμών των αισθητήρων που επιλέχθηκε από την εφαρμογή στο κινητό και γίνεται αποστολή των τιμών σε αυτό.

<http://192.168.1.120:3000/sensor/?token=24680>



Εικόνα 4.10: Esp32 – Διαδικασία με Token

Στην Εικόνα 4.10 παρουσιάζεται το διάγραμμα ροής τον τρόπο επικοινωνίας της εφαρμογής στο κινητό με το esp32 με χρήση token. Όταν η εφαρμογή στο κινητό θέλει να ζητήσει από έναν κόμβο πραγματοποιηθεί εύρεση και σύνδεση σε συγκεκριμένο url που έχει οριστεί από τον προγραμματιστή και την γνωρίζει ο χρήστης. Στο url που βάζει-καλεί πρέπει να γράψει και ένα μοναδικό token που είναι μοναδικό για κάθε esp32.

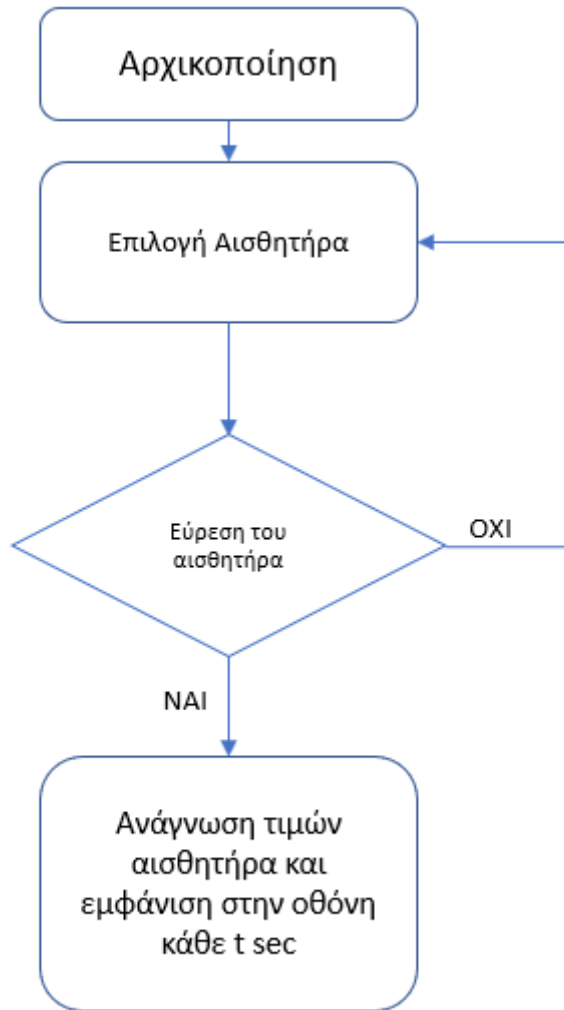
Στον παρακάτω κώδικα παρουσιάζεται ο τρόπος με τον οποίο ελέγχεται στο esp32 το token. Αν το esp32 δεν γνωρίζει το token τότε δεν μπορεί να εξυπηρετήσει την εφαρμογή στο κινητό.

```
while True:  
    conn, addr = s.accept()  
    request = conn.recv(1024)  
    request = str(request)
```

```
token = request.find('/?token=1357')

if token == 1357:
    d = dht.DHT22(machine.Pin(DHT22_PIN))
    d.measure()
    t = d.temperature()
    h = d.humidity()
    print('temperature = %.2f % t)
    print('humidity   = %.2f % h)

    m = {"t": t, "h": h}
    data = json.dumps(m)
    conn.sendall(bytes(data,encoding="utf-8"))
else
    conn.send('HTTP/1.1 404 Not Found\n')
conn.close()
```



Εικόνα 4.11: Διάγραμμα ροής για την εφαρμογή στο κινητό

Στην Εικόνα 4.11 παρουσιάζεται το διάγραμμα ροής για την εφαρμογή στο κινητό. Ο χρήστης επιλέγει τον κόμβο με τους αισθητήρες που θέλει να παρακολουθήσει και πατάει το κουμπί. Στη συνέχεια επιστρέφει την αρχική και κεντρική οθόνη και βλέπει τις τιμές από τους αισθητήρες.

Όσον αφορά την ασφάλεια πρόσβασης στο σύστημα esp32-αισθητήρα όποιος έχει πρόσβαση στο τοπικό δίκτυο που ανήκουν μπορεί να τοποθετήσει τη διεύθυνση του κόμβου και να μπορεί να λάβει οι τιμές από τους αισθητήρες. Αν όμως δεν βάλει το σωστό token δεν θα καταφέρει να πάρει τη σωστή απάντηση-δεδομένα. Κάθε esp32-κόμβος θα έχει ένα μοναδικό token

Όταν ο χρήστης τοποθετεί τη διεύθυνση του esp32-κόμβου στην εφαρμογή στο κινητό θα πρέπει να τοποθετεί και το token του κάθε esp32. Έτσι διασφαλίζεται ότι κανένας μη εξουσιοδοτημένος χρήστης δεν θα μπορεί να έχει πρόσβαση στον esp32 όταν είναι συνδεδεμένος στο τοπικό δίκτυο.

Από την άλλη όταν ο χρήστης δεν είναι συνδεδεμένος στο ίδιο τοπικό δίκτυο με την εφαρμογή στο κινητό θα πρέπει να χρησιμοποιήσει την υπηρεσία Thingspeak. Καλεί στη κατάλληλη διεύθυνση της υπηρεσίας και διαβάσει τις τιμές από το κανάλι που περιέχει δεδομένα του κόμβου.

Ο χρήστης πρέπει να γνωρίζει το κανάλι και το api key για να έχει πρόσβαση. Έτσι επιτυγχάνεται η διασφάλιση ότι κάποιος ανεπιθύμητος-μη εξουσιοδοτημένος χρήστης δεν μπορεί να διαβάσει τα δεδομένα από έναν esp32-κόμβο. Κάθε esp32-κόμβος στέλνει σε συγκεκριμένο χρονικό πλαίσιο τα δεδομένα από τους αισθητήρες στο Thingspeak με χρήση κλειδιών api keys και έτσι εξασφαλίζεται ότι χωρίς αυτά τα κλειδιά δεν μπορεί κάποιος να τοποθετήσει τιμές στο Thingspeak.

Επίσης είναι αξιοσημείωτο να ειπωθεί ότι η επικοινωνία σε τοπικό δίκτυο δεν χρησιμοποιεί Secure SSL πρωτόκολλο αλλά η επικοινωνία με το Thingspeak χρησιμοποιεί SSL σε κάθε request.

Όσον αφορά την ασφάλεια χρήσης της εφαρμογής στο κινητό να τονισθεί ότι η εφαρμογή έχει δημιουργηθεί με Flutter για Android κινητό και οποιοσδήποτε έχει την εφαρμογή στο κινητό του και γνωρίζει τη διεύθυνση ενός κόμβου ή του Thingspeak μπορεί να τα καλέσει από αυτό.

## Κεφάλαιο 5ο: Συμπεράσματα και προτάσεις βελτίωσης

Στην εργασία αυτή πραγματοποιήθηκε η μελέτη και η ανάπτυξη εφαρμογής σε κινητό τηλέφωνο για την παρακολούθηση περιβαλλοντικών συνθηκών με esp32. Η εφαρμογή android επικοινωνεί με το esp32 μέσω ασύρματου τοπικού δικτύου για την παρακολούθηση θερμοκρασίας και υγρασίας σε έναν εσωτερικό ή εξωτερικό χώρο. Ο χρήστης μπορεί μέσω της φορητής συσκευής μικρού κόστους και καταπόνησης και να επιβλέπει στο κινητό του τις τιμές των αισθητήρων σε πολύ μικρό χρόνο. Η χρησιμότητα της εφαρμογής είναι η παρακολούθηση της θερμοκρασίας/υγρασίας σε διάφορα σημεία ενός σπιτιού, αποθήκη, γενικά ενός χώρου. Ο χρήστης μπορεί να προσθέσει περισσότερους κόμβους με αισθητήρια για να παρακολουθήσει ενιαία από την εφαρμογή. Επιπλέον, προστέθηκε η δυνατότητα επίβλεψης των τιμών των αισθητήρων διαδικτυακά με τη βοήθεια της υπηρεσίας Thingspeak. Κάθε αισθητήρας-κόμβος ταυτόχρονα στέλνει τα δεδομένα του διαδικτυακά μέσω καναλιού στο Thingspeak που έχει οριστεί και ο χρήστης μέσω της εφαρμογής που υλοποιήθηκε έχει τη δυνατότητα να τα παρακολουθεί σε πραγματικό χρόνο από το κινητό του.

Συμπερασματικά ο χρήστης μπορεί με την εφαρμογή android στο κινητό του να παρακολουθεί τους αισθητήρες σε έναν χώρο. Αυτό μπορεί να το κάνει όταν το esp32 που έχει τους αισθητήρες είναι συνδεδεμένο στο ίδιο τοπικό δίκτυο με το κινητό.

Για να ξεπεράσει το εμπόδιο του port forwarding και να υπάρχει η δυνατότητα παρακολούθησης των κόμβων μέσω διαδικτύου χρησιμοποιήθηκε η υπηρεσία Thingspeak

Στις παραπάνω δυνατότητες προστίθεται η προσθήκη, η επεξεργασία και η διαγραφή νέων κόμβων μέσω της εφαρμογής.

Όσον αφορά την ασφάλεια του συστήματος η χρήση token σε κάθε requests που γίνεται τοπικά διασφαλίζει την αυθεντικοποίηση του χρήστη.

Μια από τις πιο σημαντικές βελτιώσεις είναι η ταυτοποίηση χρήστη στην εφαρμογή στο κινητό ώστε να μην μπορεί να την χρησιμοποιήσει κάποιος άλλος. Επίσης, προτείνεται η χρήση websockets για την επικοινωνία μεταξύ συσκευών όταν βρίσκονται πίσω από κάποιο δρομολογητή ή firewall.

## ΒΙΒΛΙΟΓΡΑΦΙΑ

- [1] Rap Payne, Beginning App Development with Flutter
- [2] <https://flutter.dev/docs/development/ui/widgets-intro>. *flutter.dev*. Retrieved 2021-05-01
- [3] [https://www.espressif.com/sites/default/files/documentation/esp32\\_datasheet\\_en.pdf](https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf), "ESP32 Datasheet" (PDF). Espressif Systems. 2017-03-06. Retrieved 2021-05-01
- [4] C. G. C. Carducci, A. Monti, M. H. Schraven, M. Schumacher and D. Mueller, "Enabling ESP32-based IoT Applications in Building Automation Systems," *2019 II Workshop on Metrology for Industry 4.0 and IoT (MetroInd4.0&IoT)*, 2019, pp. 306-311, doi: 10.1109/METROI4.2019.8792852.
- [5] O. Barybin, E. Zaitseva and V. Brazhnyi, "Testing the Security ESP32 Internet of Things Devices," *2019 IEEE International Scientific-Practical Conference Problems of Infocommunications, Science and Technology (PIC S&T)*, 2019, pp. 143-146, doi: 10.1109/PICST47496.2019.9061269.
- [6] N. Nikolov and O. Nakov, "Research of Secure Communication of Esp32 IoT Embedded System to.NET Core Cloud Structure using MQTTS SSL/TLS," *2019 IEEE XXVIII International Scientific Conference Electronics (ET)*, 2019, pp. 1-4, doi: 10.1109/ET.2019.8878636.
- [7] A. Praveen *et al.*, "Conference Room Booking Application using Flutter," *2020 International Conference on Communication and Signal Processing (ICCSP)*, 2020, pp. 0348-0350, doi: 10.1109/ICCSP48568.2020.9182183.
- [8] S. Boukhary and E. Colmenares, "A Clean Approach to Flutter Development through the Flutter Clean Architecture Package," *2019 International Conference on Computational Science and Computational Intelligence (CSCI)*, 2019, pp. 1115-1120, doi: 10.1109/CSCI49370.2019.00211.
- [9] J. Y. Kim, H. Lee, J. Son and J. Park, "Smart home web of objects-based IoT management model and methods for home data mining," *2015 17th Asia-Pacific Network Operations and Management Symposium (APNOMS)*, 2015, pp. 327-331, doi: 10.1109/APNOMS.2015.7275448.
- [10] V. D. Vaidya and P. Vishwakarma, "A Comparative Analysis on Smart Home System to Control, Monitor and Secure Home, based on technologies like GSM, IOT, Bluetooth and PIC Microcontroller with ZigBee Modulation," *2018 International Conference on Smart City and Emerging Technology (ICSCET)*, 2018, pp. 1-4, doi: 10.1109/ICSCET.2018.8537381.
- [11] S. ur Rehman and V. Gruhn, "An approach to secure smart homes in cyber-physical systems/Internet-of-Things," *2018 Fifth International Conference on Software Defined Systems (SDS)*, 2018, pp. 126-129, doi: 10.1109/SDS.2018.8370433.
- [12] S. Ghosh, "Smart homes: Architectural and engineering design imperatives for smart city building codes," *2018 Technologies for Smart-City Energy Security and Power (ICSESP)*, 2018, pp. 1-4, doi: 10.1109/ICSESP.2018.8376676.

## ΠΑΡΑΡΤΗΜΑ Α

Στο παράρτημα αυτό αναφέρονται τα βασικά κομμάτια του κώδικα που χρησιμοποιήθηκε.

### Flutter - Main

```
import 'package:flutter/material.dart';
import 'dart:async';
import 'package:http/http.dart' as http;
import 'dart:convert' as convert;
import 'package:intl/intl.dart';
import 'package:shared_preferences/shared_preferences.dart';
import 'package:appsensorsiot/nodelist.dart';
```

```
void main() {
  runApp(MyApp());
}
```

```
class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      debugShowCheckedModeBanner: false,
      title: 'Mimikos Sensor Nodes',
      theme: ThemeData(
        primarySwatch: Colors.green,
      ),
      home: MyHomePage(title: 'Mimikos Sensor Nodes'),
    );
  }
}
```

```
class MyHomePage extends StatefulWidget {
  MyHomePage({Key key, this.title}) : super(key: key);

  final String title;

  @override
  _MyHomePageState createState() => _MyHomePageState();
}
```

```
class _MyHomePageState extends State<MyHomePage> {
  int _counter = 0;
  String temp = '...';
  String hum = '...';
  var client = http.Client();
  String titlevalue = "...";
}
```

```

String urlvalue = "...";
int mode = 0;

@override
void initState() {
  readsettings();
  setState(() {});
  Timer.periodic(Duration(seconds: 5), (timer) {
    setState(() {
      if (mode == 1) getdataFromThingspeak();
      if (mode == 2) getdataFromLocal();
    });
  });
}

void readsettings() async {
  print("readsettings");
  SharedPreferences prefs = await SharedPreferences.getInstance();
  titlevalue = prefs.getString('selectednodetitle') ?? '24.5 C';
  urlvalue = prefs.getString('selectednodeurl') ?? '48 %';
  setState(() {
    titlevalue = titlevalue;
    urlvalue = urlvalue;
  });
  int hashttp = urlvalue.indexOf('http');
  int findthingspeak = urlvalue.indexOf('thingspeak');
  if (findthingspeak != -1) {
    mode = 1;
    return;
  }

  if (hashttp == -1) {
    mode = 0;
  } else {
    mode = 2;
    //getdataFromLocal();
  }
}

void getdataFromThingspeak() async {
  print("getdataFromThingspeak");
  var response = await client.get(urlvalue);
  var data = convert.jsonDecode(response.body);
  var feeds = data['feeds'][0];
  temp = feeds['field1'];
  hum = feeds['field2'];
  var f = NumberFormat("###.0#", "en_US");
  setState(() {

```

```

        temp = f.format(double.parse(temp));
        hum = f.format(double.parse(hum));
    });
}

```

```

void getDataFromLocal() async {
    print("getDataFromLocal");
    var response = await client.get(urlvalue);
    var data = convert.jsonDecode(response.body);
    temp = data['temp'].toString();
    hum = data['hum'].toString();
    var f = NumberFormat("###.0#", "en_US");
    setState(() {
        temp = f.format(double.parse(temp));
        hum = f.format(double.parse(hum));
    });
}

```

`@override`

```

Widget build(BuildContext context) {
    return Scaffold(
        appBar: AppBar(
            title: Text('Επιλεγμένος Κόμβος'),
        ),
        body: Center(
            child: Column(
                mainAxisAlignment: MainAxisAlignment.center,
                children: <Widget>[
                    Text(
                        'Κόμβος',
                        style: TextStyle(
                            fontSize: 60,
                            foreground: Paint()
                                ..style = PaintingStyle.stroke
                                ..strokeWidth = 3
                                ..color = Colors.black,
                        ),
                    ),
                    SizedBox(
                        height: 20,
                    ),
                    SizedBox(
                        height: 2,
                        width: 200,
                        child: const DecoratedBox(
                            decoration: const BoxDecoration(color: Colors.red),
                        ),
                    ),
                ],
            ),
        ),
    );
}

```

```

    SizedBox(
      height: 20,
    ),
    Text(
      '$titlevalue',
      style: Theme.of(context).textTheme.headline5,
    ),
    SizedBox(
      height: 20,
    ),
    SizedBox(
      height: 2,
      width: 200,
      child: const DecoratedBox(
        decoration: const BoxDecoration(color: Colors.red),
      ),
    ),
    SizedBox(
      height: 20,
    ),
    SizedBox(
      height: 70,
    ),
    Text(
      'Θερμοκρασία',
      style: TextStyle(
        fontSize: 40,
        foreground: Paint()
          ..style = PaintingStyle.stroke
          ..strokeWidth = 2
          ..color = Colors.blue[700],
      ),
    ),
    SizedBox(
      height: 20,
    ),
    Text(
      '$temp', // + ' C',
      style: Theme.of(context).textTheme.headline5,
    ),
    SizedBox(
      height: 20,
    ),
    Text(
      'Υγρασία',
      style: TextStyle(
        fontSize: 40,
        foreground: Paint()

```

