



ΠΑΝΕΠΙΣΤΗΜΙΟ  
ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ  
UNIVERSITY OF WEST ATTICA

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ  
ΥΠΟΛΟΓΙΣΤΩΝ

ΤΕΛΙΚΗ ΕΡΓΑΣΙΑ

ΗΛΕΚΤΡΙΚΟΣ ΦΩΤΕΙΝΟΣ ΣΗΜΑΤΟΔΟΤΗΣ

**ΣΤΟΙΧΕΙΑ ΟΜΑΔΑΣ / ΕΡΓΑΣΙΑΣ**

---

**ΣΤΟΙΧΕΙΑ ΦΟΙΤΗΤΗ 1 :** ΑΘΑΝΑΣΙΟΥ ΒΑΣΙΛΕΙΟΣ ΕΥΑΓΓΕΛΟΣ – 19390005 – ΠΑΔΑ

**ΣΤΟΙΧΕΙΑ ΦΟΙΤΗΤΗ 2 :** ΜΑΝΤΖΟΥΚΑΣ ΑΓΓΕΛΟΣ-ΒΑΣΙΛΕΙΟΣ – 19390128 – ΠΑΔΑ

**ΣΤΟΙΧΕΙΑ ΦΟΙΤΗΤΗ 3 :** ΠΥΛΑΡΙΝΟΣ ΧΡΗΣΤΟΣ – 20390278 – ΠΑΔΑ

**ΥΠΕΥΘΥΝΗ ΕΡΓΑΣΤΗΡΙΟΥ :** ΓΑΡΟΦΑΛΑΚΗ ΖΑΧΑΡΕΝΙΑ

**ΥΠΕΥΘΥΝΟΣ ΘΕΩΡΙΑΣ :** ΚΑΡΚΑΖΗΣ ΠΑΝΑΓΙΩΤΗΣ

# ΔΙΑΔΙΚΤΥΟ ΤΩΝ ΑΝΤΙΚΕΙΜΕΝΩΝ

## ΠΕΡΙΕΧΟΜΕΝΑ

<b>1. Εισαγωγή</b>	<b>2</b>
1.1 Αναφορά στις βασικές έννοιες	2
<b>2. Ζητούμενα και Ανάλυση</b>	<b>3</b>
2.1 Ζητούμενο Α: Ηλεκτρικός φωτεινός σηματοδότης	3
2.1.1 Περιγραφή λειτουργιών .....	3
2.1.2 Χρήση του Arduino και πλατφόρμας ThingSpeak .....	3
2.1.3 Απαιτήσεις για προγραμματισμό και συνδεσιμότητα .....	4
2.2 Ζητούμενο Β: Αποστολή δεδομένων σε κανάλι	6
2.2.1 Περιγραφή στόχων και λειτουργιών .....	6
2.2.2 Ανάλυση προγραμματιστικών βημάτων .....	6
2.3 Ζητούμενο Γ: Ανάγνωση δεδομένων από κανάλι	7
2.3.1 Περιγραφή απαιτούμενων ρυθμίσεων και λειτουργιών .....	7
2.3.2 Ανάλυση προγραμματιστικών βημάτων .....	7
<b>3. Περιγραφή Κυκλώματος Υλικού</b>	<b>8</b>
3.1 Ανάλυση Συνδεσμολογίας	8
3.2 Χρησιμοποιημένα Εξαρτήματα και ο Ρόλος τους	8
3.3 Διάγραμμα Κυκλώματος	9
<b>4. Ρυθμίσεις Καναλιού ThingSpeak</b>	<b>10</b>
4.1 Επεξήγηση των μεταβλητών και των ρυθμίσεων	10
4.2 Περιγραφή της χρήσης οπτικών στοιχείων	11
<b>5. Προγραμματιστικός Κώδικας</b>	<b>14</b>
5.1 Ανάλυση του κώδικα που χρησιμοποιείται	14
5.2 Σχολιασμός και εξήγηση για το τι κάνει κάθε κομμάτι του κώδικα	15
5.3 Παρουσίαση αποτελεσμάτων	19
5.4 Προβλήματα που εντοπίστηκαν και λύσεις που δοκιμάστηκαν	23
<b>6. Συμπεράσματα</b>	<b>24</b>
6.1 Ανασκόπηση της εργασίας	24
6.2 Προτάσεις για βελτιώσεις	24

## 1. Εισαγωγή

Η εργασία αφορά την μελέτη ενός ηλεκτρικού φωτεινού σηματοδότη του οποίου η λειτουργία ορίζεται από τον μικροελεγκτή Arduino UNO και τα δεδομένα αποθηκεύονται σε πραγματικό χρόνο (real-time) στην πλατφόρμα ThingSpeak. Η υλοποίηση και η ανάπτυξη αυτής της IoT εφαρμογής απαιτεί ορισμένες προδιαγραφές οι οποίες είναι:

- **Κατανόηση των μικροελεγκτών και της αλληλεπίδρασής τους με ηλεκτρονικά στοιχεία:** Με λίγα λόγια, απαιτείται κατανόηση του υλικού εξοπλισμού που περιλαμβάνει αναλογικές και ψηφιακές εισόδους/εξόδους, όπως έχει ο μικροελεγκτής Arduino UNO και το WiFi module ESP-01.
- **Δικτύωση και ανταλλαγή δεδομένων:** Η σύνδεση του Arduino μέσω WiFi και η ανταλλαγή δεδομένων με κανάλια της πλατφόρμας ThingSpeak, τόσο για την απεικόνιση όσο και για την αποστολή/λήψη δεδομένων.
- **Γραφική απεικόνιση δεδομένων:** Η χρήση γραφικών στοιχείων για την παρουσίαση των δεδομένων και της κατάστασης του συστήματος.
- **Ανάγνωση και συγγραφή δεδομένων:** Η προγραμματιστική ικανότητα να διαβάζουμε και να στέλνουμε σε πραγματικό χρόνο δεδομένα.

### 1.1 Αναφορά στις βασικές έννοιες

Οι βασικές έννοιες που χρησιμοποιούνται στην εργασία του ηλεκτρικού φωτεινού σηματοδότη είναι οι εξής:

- **Μικροελεγκτές (Arduino):** Οι μικροελεγκτές είναι μικρές, ευέλικτες συσκευές που χρησιμοποιούνται για τον έλεγχο ηλεκτρονικών κυκλωμάτων. Στην εργασία αυτή, ο μικροελεγκτής Arduino UNO αναλαμβάνει την εναλλαγή των φωτεινών ενδείξεων ενός σηματοδότη και την επικοινωνία με την πλατφόρμα ThingSpeak.
- **Διαδίκτυο των Αντικειμένων (IoT):** Το IoT αναφέρεται στη σύνδεση φυσικών αντικειμένων στο διαδίκτυο, επιτρέποντας την ανταλλαγή δεδομένων μεταξύ αυτών και άλλων συστημάτων. Εδώ, οι ενδείξεις του φωτεινού σηματοδότη διαμοιράζονται και εμφανίζονται μέσω της πλατφόρμας ThingSpeak.
- **ThingSpeak:** Είναι μια πλατφόρμα IoT που επιτρέπει τη συλλογή, αποθήκευση, ανάλυση και απεικόνιση δεδομένων από συνδεδεμένες συσκευές σε πραγματικό χρόνο. Στην παρούσα εργασία χρησιμοποιείται για την αποστολή δεδομένων από το Arduino και την εμφάνιση των ενδείξεων.
- **Πρωτόκολλα και APIs:** Η επικοινωνία με το ThingSpeak γίνεται μέσω πρωτοκόλλων και APIs, τα οποία επιτρέπουν την ασφαλή ανταλλαγή δεδομένων.
- **ESP-01 (WiFi Module):** Ένα απλό και οικονομικό module που χρησιμοποιείται για τη σύνδεση του Arduino σε ασύρματο δίκτυο, επιτρέποντας την πρόσβαση στο διαδίκτυο και την επικοινωνία με το ThingSpeak.

## 2. Ζητούμενα και Ανάλυση

### 2.1 Ζητούμενο Α: Ηλεκτρικός φωτεινός σηματοδότης

#### 2.1.1 Περιγραφή λειτουργιών

Ο φωτεινός σηματοδότης θα λειτουργεί ακολουθώντας την κλασική αλληλουχία χρωμάτων (κόκκινο, πράσινο, πορτοκαλί), με το **Arduino** να ελέγχει τη χρονική διάρκεια κάθε ένδειξης: κόκκινο και πράσινο για 30 δευτερόλεπτα, πορτοκαλί για 20 δευτερόλεπτα. Αυτή η διαδικασία θα επαναλαμβάνεται συνεχώς χωρίς διακοπή.

Παράλληλα, η λειτουργία του σηματοδότη θα απεικονίζεται στην πλατφόρμα **ThingSpeak** μέσω διαδικτυακής σύνδεσης με το **ESP-01**. Έτσι, οι χρήστες θα μπορούν να παρακολουθούν την κατάσταση του σηματοδότη σε πραγματικό χρόνο, από απόσταση, μέσω του διαδικτύου.

#### 2.1.2 Χρήση του Arduino και πλατφόρμας ThingSpeak

Για την δημιουργία του ηλεκτρικού φωτεινού σηματοδότη, η συνδυασμένη χρήση του μικροελεγκτή **Arduino** και της πλατφόρμας **ThingSpeak** εξασφαλίζει την παρακολούθηση και την απεικόνιση των δεδομένων του σηματοδότη σε πραγματικό χρόνο μέσω διαδικτύου. Πιο αναλυτικά το καθένα:

#### Arduino UNO

Το **Arduino UNO** είναι ο μικροελεγκτής που αναλαμβάνει τον έλεγχο των φωτεινών ενδείξεων του σηματοδότη. Ο προγραμματισμός του γίνεται μέσω της γλώσσας **Arduino C**, και θα περιλαμβάνει τις εξής βασικές λειτουργίες:

1. **Έλεγχος οπτικών στοιχείων στο ThingSpeak:** Ο σηματοδότης αποτελείται από τρία διαφορετικά οπτικά στοιχεία (κόκκινο, πράσινο, πορτοκαλί) στο ThingSpeak, τα οποία εναλλάσσονται με βάση τις απαιτήσεις του προγράμματος. Ο μικροελεγκτής θα ενεργοποιεί κάθε οπτικό στοιχείο για το αντίστοιχο χρονικό διάστημα
2. **Χρονοδιακόπτης:** Ο κώδικας πρέπει να περιλαμβάνει timers που θα μετρούν τη διάρκεια κάθε φάσης του σηματοδότη (κόκκινο, πράσινο, πορτοκαλί). Αυτό επιτυγχάνεται με τη χρήση της εντολής delay().
3. **WiFi Συνδεσιμότητα:** Ο Arduino πρέπει να συνδέεται σε WiFi δίκτυο μέσω του ESP-01, ώστε να μπορεί να στέλνει δεδομένα στο διαδίκτυο και να επικοινωνεί με την πλατφόρμα ThingSpeak.

#### ThingSpeak

Η πλατφόρμα **ThingSpeak** λειτουργεί ως ο διαδικτυακός χώρος στον οποίο αποστέλλονται και αποθηκεύονται τα δεδομένα του φωτεινού σηματοδότη. Παρέχει δυνατότητες παρακολούθησης και ανάλυσης των τιμών που λαμβάνει από τον Arduino:

## ΔΙΑΔΙΚΤΥΟ ΤΩΝ ΑΝΤΙΚΕΙΜΕΝΩΝ

1. **Κανάλι:** Στην πλατφόρμα δημιουργείται ένα κανάλι όπου καταγράφονται οι μεταβολές της κατάστασης του σηματοδότη. Τα πεδία του καναλιού αντιστοιχούν σε μεταβλητές που δείχνουν αν η ένδειξη του οπτικού στοιχείου είναι ενεργή.
2. **Μεταβλητές:** Κάθε χρώμα του σηματοδότη συνδέεται με μια μεταβλητή στο κανάλι του ThingSpeak. Όταν ένα οπτικό στοιχείο ανάβει, η αντίστοιχη μεταβλητή παίρνει την τιμή 1 για κόκκινο, 2 για πορτοκαλί, 3 για πράσινο, ενώ όταν παίρνει την τιμή 0, τότε σβήνει.
3. **Απεικόνιση:** Τα δεδομένα που στέλνει ο Arduino μπορούν να απεικονιστούν γραφικά μέσω της πλατφόρμας. Με αυτόν τον τρόπο, ο χρήστης μπορεί να παρακολουθεί σε πραγματικό χρόνο ποια χρώματα είναι ενεργά και για πόσο χρόνο.
4. **Επικοινωνία Δεδομένων:** Ο Arduino αποστέλλει τα δεδομένα στην πλατφόρμα μέσω HTTP αιτημάτων

### Διασύνδεση και Μετάδοση Δεδομένων

Η σύνδεση του Arduino με την πλατφόρμα ThingSpeak επιτυγχάνεται μέσω του **ESP-01**, το οποίο επιτρέπει τη μετάδοση των δεδομένων από τον μικροελεγκτή στο διαδίκτυο. Χρησιμοποιώντας HTTP αιτήματα, τα δεδομένα για την τρέχουσα κατάσταση του σηματοδότη αποστέλλονται αυτόματα στο ThingSpeak, όπου αποθηκεύονται και εμφανίζονται σε πραγματικό χρόνο.

Με αυτόν τον τρόπο, η λειτουργία του φωτεινού σηματοδότη μπορεί να παρακολουθείται και να καταγράφεται απομακρυσμένα, διασφαλίζοντας έτσι συνεχή ενημέρωση για την λειτουργία του σηματοδότη.

### **2.1.3 Απαιτήσεις για προγραμματισμό και συνδεσιμότητα**

Για την υλοποίηση του συστήματος του ηλεκτρικού φωτεινού σηματοδότη με τη χρήση του **Arduino** και της πλατφόρμας **ThingSpeak**, απαιτούνται συγκεκριμένες προγραμματιστικές παραδοχές. Αυτές οι απαιτήσεις περιλαμβάνουν την προετοιμασία του κώδικα για τον έλεγχο των οπτικών στοιχείων, την εξασφάλιση της σωστής συνδεσιμότητας στο διαδίκτυο και την αποστολή δεδομένων σε πραγματικό χρόνο.

### Προγραμματισμός του Arduino

Η βασική λειτουργία του σηματοδότη προϋποθέτει την ανάπτυξη ενός προγράμματος στο **Arduino**, το οποίο θα:

1. **Διαχειρίζεται τις Ενδείξεις:** Ο κώδικας πρέπει να ελέγχει την αλληλουχία των χρωμάτων του σηματοδότη (κόκκινο, πράσινο, πορτοκαλί) και να καθορίζει τη διάρκεια κάθε ένδειξης, όπως απαιτείται ( $\geq 30$  δευτερόλεπτα για κόκκινο και πράσινο,  $\geq 20$  δευτερόλεπτα για πορτοκαλί).

## ΔΙΑΔΙΚΤΥΟ ΤΩΝ ΑΝΤΙΚΕΙΜΕΝΩΝ

2. **Ρυθμίζει τις Χρονικές Καθυστερήσεις:** Η διάρκεια κάθε φάσης του σηματοδότη ελέγχεται με τη χρήση χρονοκαθυστερήσης, η οποία διασφαλίζει ότι τα οπτικά στοιχεία ανάβουν και σβήνουν με τον σωστό ρυθμό.
3. **Αποστέλλει Δεδομένα:** Ο προγραμματισμός περιλαμβάνει εντολές που επιτρέπουν στον Arduino να αποστέλλει δεδομένα στην πλατφόρμα ThingSpeak, ενημερώνοντας τις μεταβλητές που αντιπροσωπεύουν την κατάσταση κάθε οπτικού στοιχείου.

### Σύνδεση με την Πλατφόρμα ThingSpeak

Η επικοινωνία του Arduino με την πλατφόρμα **ThingSpeak** είναι το κλειδί για να βλέπουμε την κατάσταση του φωτεινού σηματοδότη. Για να επιτευχθεί αυτό, χρειάζεται:

1. **WiFi Σύνδεση:** Ο μικροελεγκτής πρέπει να συνδεθεί με το πιο κοντινό WiFi access point χρησιμοποιώντας το ESP-01 WiFi module. Η σύνδεση αυτή επιτρέπει στον Arduino να στέλνει δεδομένα στο διαδίκτυο, τα οποία στη συνέχεια εμφανίζονται στο κανάλι του ThingSpeak.
2. **Ασφαλής Μεταφορά Δεδομένων:** Η αποστολή των δεδομένων πρέπει να γίνει με ασφαλή και αξιόπιστο τρόπο, χρησιμοποιώντας HTTP requests.
3. **Προγραμματισμός Καναλιού στο ThingSpeak:** Το κανάλι στην πλατφόρμα πρέπει να ρυθμιστεί σωστά ώστε να δέχεται και να απεικονίζει τις μεταβολές της κατάστασης του σηματοδότη. Αυτό περιλαμβάνει τη δημιουργία πεδίων για τις μεταβλητές που αντιπροσωπεύουν τα οπτικά στοιχεία.

### Επικοινωνία και Συνδεσιμότητα

Η επιτυχής σύνδεση και η συνεχής λειτουργία του συστήματος απαιτούν αξιόπιστη επικοινωνία μεταξύ του Arduino και της πλατφόρμας. Για να διασφαλιστεί αυτό χρειάζεται:

1. **Σταθερή Σύνδεση στο Διαδίκτυο:** Ο μικροελεγκτής πρέπει να έχει μόνιμη πρόσβαση στο δίκτυο για την αδιάλειπτη αποστολή δεδομένων.
2. **Συνεχής Ενημέρωση Δεδομένων:** Ο κώδικας θα πρέπει να προβλέπει την τακτική αποστολή των τιμών στο ThingSpeak, ώστε οι χρήστες να βλέπουν τις ενημερώσεις σε πραγματικό χρόνο.

Η παραπάνω προσέγγιση εξασφαλίζει ότι το σύστημα θα λειτουργεί ομαλά, και η συνδεσιμότητα μεταξύ του Arduino και της πλατφόρμας θα παραμένει σταθερή καθ' όλη τη διάρκεια της λειτουργίας του φωτεινού σηματοδότη.

# ΔΙΑΔΙΚΤΥΟ ΤΩΝ ΑΝΤΙΚΕΙΜΕΝΩΝ

## 2.2 Ζητούμενο Β: Αποστολή δεδομένων σε κανάλι

### 2.2.1 Περιγραφή στόχων και λειτουργιών

Ο κύριος στόχος του ζητούμενου είναι η ανάπτυξη μιας λειτουργικότητας που επιτρέπει την αποστολή δεδομένων σε κανάλι στην πλατφόρμα ThingSpeak. Η λειτουργικότητα αυτή αφορά την αποστολή δεδομένων σ' ένα πεδίο που λαμβάνει τιμές ένδειξης για την λειτουργία του φαναριού, και περιλαμβάνει τα εξής:

#### Αποστολή δεδομένων:

- Ορίζεται η τιμή **0** στη μεταβλητή **Field 8** του καναλιού μας, χρησιμοποιώντας HTTP αιτήματα για την επικοινωνία με το API του ThingSpeak.
- Η ενέργεια αυτή αποσκοπεί στη δυνατότητα αποστολής δεδομένων σε κανάλι.

#### Προτάσεις για βελτίωση:

- Η συγκεκριμένη υλοποίηση μπορεί να προσαρμοστεί για την αποστολή διαφορετικών δεδομένων ή την επικοινωνία με κανάλι άλλης εφαρμογής, υποστηρίζοντας την ανάπτυξη μεγαλύτερων και πιο σύνθετων IoT συστημάτων.

Η υλοποίηση της παραπάνω λειτουργίας γίνεται με τη χρήση του ESP-01 για τη σύνδεση στο WiFi και την αποστολή HTTP αιτημάτων μέσω του Arduino, ακολουθώντας τις οδηγίες του API της πλατφόρμας ThingSpeak.

### 2.2.2 Ανάλυση προγραμματιστικών βημάτων

Για την υλοποίηση της αποστολής δεδομένων στο κανάλι μας, τα κύρια βήματα περιλαμβάνουν:

#### Σύνδεση στο WiFi:

Η σύνδεση του Arduino με το δίκτυο WiFi μέσω του ESP-01 είναι απαραίτητη για την αποστολή δεδομένων στο ThingSpeak.

#### Εντολές και πληροφορίες για το κανάλι:

- Χρήση HTTP αιτημάτων (**GET**) για την αποστολή δεδομένων στο **Field 8** του καναλιού μας.
- Απαιτείται το API Key του καναλιού μας (το οποίο πρέπει να προστεθεί στην αντίστοιχη μεταβλητή του κώδικα **myWriteAPI**).

#### Επαλήθευση Λειτουργίας:

- Ο κώδικας ελέγχει την επιτυχία της αποστολής μέσω απαντήσεων από το ThingSpeak API.
- Καταγράφει μηνύματα στο σειριακό monitor για debugging.

## 2.3 Ζητούμενο Γ: Ανάγνωση δεδομένων από κανάλι

### 2.3.1 Περιγραφή απαιτούμενων ρυθμίσεων και λειτουργιών

Για την ανάγνωση των δεδομένων από το κανάλι Field 8 που περιλαμβάνει ενδείξεις για την λειτουργία του σηματοδότη απαιτούνται τα εξής:

- **Ρυθμίσεις ESP-01 και Wi-Fi**
  - Η σύνδεση με το δίκτυο Wi-Fi πρέπει να έχει ρυθμιστεί μέσω της εντολής **AT+CWJAP**, χρησιμοποιώντας τα κατάλληλα στοιχεία δικτύου (SSID και password)
  - Το ESP-01 πρέπει να είναι ρυθμισμένο σε λειτουργία client (**AT+CWMODE=1**) για να επικοινωνεί με τον διακομιστή του ThingSpeak
- **Διαμόρφωση API και καναλιού**
  - Απαιτείται το **Read API Key** του καναλιού, το οποίο επιτρέπει την πρόσβαση στα δεδομένα του.
  - Πρέπει να καθοριστεί ο αριθμός καναλιού (Channel ID) για το συγκεκριμένο κανάλι της εφαρμογής.
- **Λειτουργία HTTP GET**
  - Η επικοινωνία με το ThingSpeak API γίνεται μέσω αιτημάτων HTTP GET, όπου ζητούνται τα δεδομένα του καναλιού σε μορφή JSON (χρειαζόμαστε και το Channel ID).
  - Χρησιμοποιείται η εντολή **AT+CIPSTART** για τη δημιουργία TCP σύνδεσης με τον διακομιστή, και στη συνέχεια η εντολή **AT+CIPSEND** για την αποστολή του αιτήματος.
- **Ανάλυση Δεδομένων (JSON)**
  - Η απάντηση που λαμβάνεται από το κανάλι πρέπει να υποβληθεί σε επεξεργασία ώστε να εξαχθεί η απαραίτητη πληροφορία (π.χ. τα πεδία των καναλιών)

### 2.3.2 Ανάλυση προγραμματιστικών βημάτων

Για την υλοποίηση του ζητουμένου ακολουθήσαμε τα συγκεκριμένα βήματα:

- **Διαμόρφωση σύνδεσης**
  - Για να διαμορφωθεί η σύνδεση αποστέλλονται οι κατάλληλες AT εντολές για την έναρξη της επικοινωνίας του Arduino UNO με το ThingSpeak.
- **Σύνταξη και Αποστολή αιτήματος HTTP GET**
  - Για να πάρουμε τα δεδομένα του καναλιού σε μορφή JSON αποστέλλουμε το κατάλληλο HTTP request αποθηκεύουμε το μήκος του μηνύματος και στέλνουμε το αίτημα



# ΔΙΑΔΙΚΤΥΟ ΤΩΝ ΑΝΤΙΚΕΙΜΕΝΩΝ

- **Λήψη και ανάλυση απάντησης**
  - Διαβάζουμε την απάντηση από το ESP-01 και την αποθηκεύουμε σε μια μεταβλητή.
  - Εντοπίζουμε και εξάγουμε την τιμή από το JSON χρησιμοποιώντας συναρτήσεις όπως `indexOf()` και `substring()`
- **Χειρισμός Δεδομένων**
  - Ανάλογα με την τιμή που εξάγεται από το πεδίο, εκτελούμε την αντίστοιχη λειτουργία, όπου με τιμή 0 ο σηματοδότης παραμένει σε λειτουργία, ενώ με τιμή 1, τίθεται εκτός λειτουργίας ανάβοντας το πορτοκαλί φωτισμό.
- **Επαλήθευση ελέγχου της λειτουργίας του σηματοδότη**
  - Κάθε 10 λεπτά έχει ρυθμιστεί να αποστέλλεται στο κανάλι field8 η ένδειξη εκτός λειτουργίας για το φανάρι, ώστε να ελεγχθεί ότι η ρύθμιση λειτουργεί.
- **Κλείσιμο Σύνδεσης**
  - Κλείνουμε τη σύνδεση TCP μετά την ολοκλήρωση του αιτήματος

## 3. Περιγραφή Κυκλώματος Υλικού

Το κύκλωμα του φωτεινού σηματοδότη βασίζεται σε ένα Arduino UNO και χρησιμοποιεί το ESP-01 για σύνδεση στο διαδίκτυο, όπως φαίνεται στην εικόνα. Το κύκλωμα συνδυάζει τη φυσική απεικόνιση των φωτεινών σημάτων με τη δυνατότητα απομακρυσμένης παρακολούθησης μέσω της πλατφόρμας ThingSpeak.

### 3.1 Ανάλυση Συνδεσμολογίας

Η συνδεσμολογία βασίζεται σε ένα Arduino UNO που συνδέεται με το ESP-01 μέσω του breadboard. Η τροφοδοσία παρέχεται στο ESP-01 από το Arduino, ενώ η σειριακή επικοινωνία μεταξύ των δύο μονάδων επιτυγχάνεται μέσω των θυρών 6 και 7.

- Το ESP-01 συνδέεται με το breadboard και στη συνέχεια με το Arduino μέσω καλωδίων, επιτρέποντας την ανταλλαγή δεδομένων.
- Τα απαραίτητα σήματα τροφοδοσίας (VCC, GND) παρέχονται στο module από την πλακέτα Arduino.

### 3.2 Χρησιμοποιημένα Εξαρτήματα και ο Ρόλος τους

- **Arduino:** Ο μικροελεγκτής που είναι υπεύθυνος για τον έλεγχο των οπτικών στοιχείων και για τη διασύνδεση με την πλατφόρμα ThingSpeak.
- **ESP-01:** Το WiFi module που συνδέει τον Arduino στο δίκτυο και επιτρέπει την αποστολή δεδομένων στο ThingSpeak.

## ΔΙΑΔΙΚΤΥΟ ΤΩΝ ΑΝΤΙΚΕΙΜΕΝΩΝ

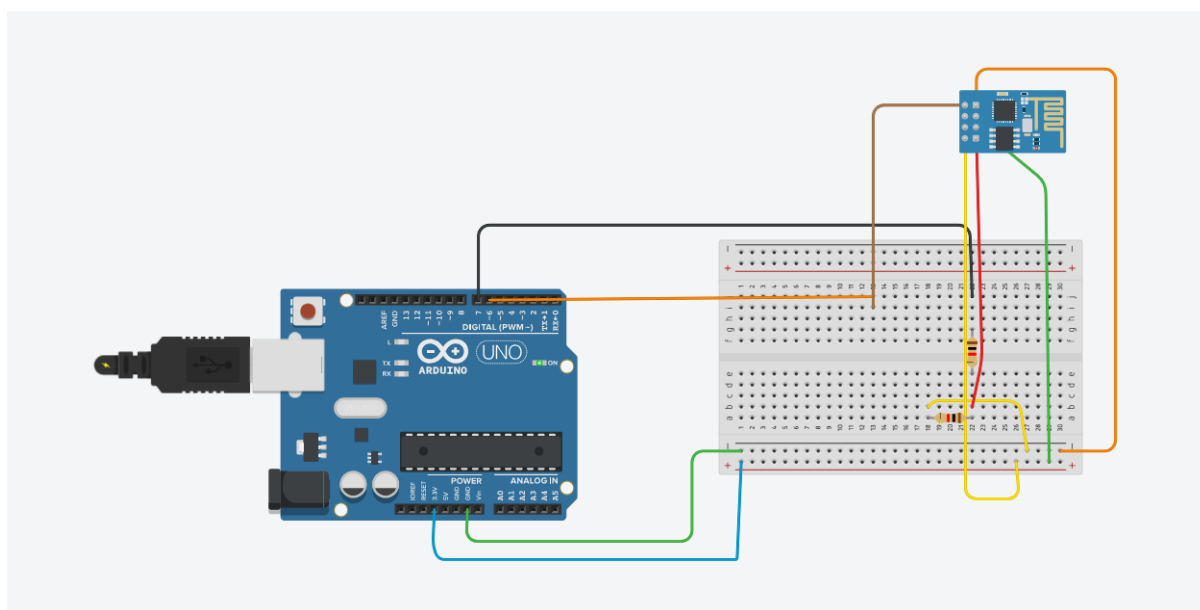
- **Αντιστάσεις:** Οι αντιστάσεις παίζουν σημαντικό ρόλο για τη σωστή λειτουργία και προστασία της σύνδεσης μεταξύ του Arduino Uno και του ESP-01 WiFi module. Οι δύο αντιστάσεις δημιουργούν έναν διαιρέτη τάσης για το σήμα TX (Transmit) από το Arduino προς το RX (Receive) pin του ESP-01. Ο ρόλος του είναι να μειώσει την τάση από 5V στα 3.3V για να προστατευτεί το ESP-01 από τάσεις μεγαλύτερες των 3.3 V που μπορούν να προκαλέσουν ζημιά
- **Σύνδεση μέσω καλωδίων:** Τα καλώδια χρησιμοποιούνται για τη σύνδεση του Arduino με το ESP-01, καθώς και για την παροχή τάσης και γείωσης στο κύκλωμα.

### 3.3 Διάγραμμα Κυκλώματος

Το διάγραμμα του κυκλώματος μπορεί να περιγραφεί ως εξής:

- Το Arduino UNO συνδέεται με το ESP-01 μέσω καλωδίων για την παροχή ρεύματος και τη σειριακή επικοινωνία.
- Η συνδεσμολογία γίνεται σε ένα breadboard, όπου τα καλώδια συνδέουν τις αντίστοιχες θύρες μεταξύ των δύο εξαρτημάτων.
- Η πλακέτα Arduino παρέχει την τροφοδοσία και τον έλεγχο στο ESP-01 για τη σύνδεση στο διαδίκτυο.

Το κύκλωμα μπορεί να επεκταθεί με την προσθήκη των LED για την απεικόνιση των ενδείξεων του σηματοδότη, όπως περιγράφηκε στα προηγούμενα βήματα.



**Εικόνα 1.** Το κύκλωμα του Arduino UNO με το ESP8266 Wi-Fi module στο Tinkercad

## 4. Ρυθμίσεις Καναλιού ThingSpeak

Σε αυτή την ενότητα περιγράφονται οι ρυθμίσεις του καναλιού ThingSpeak, καθώς και η χρήση των οπτικών στοιχείων και των διαγραμμάτων για την απεικόνιση της λειτουργίας του συστήματος φωτεινού σηματοδότη. Παρακάτω παρατίθενται οι λεπτομέρειες των ρυθμίσεων και των οπτικών στοιχείων.

### 4.1 Επεξήγηση των μεταβλητών και των ρυθμίσεων

The screenshot shows the 'Channel Settings' page for a ThingSpeak channel named 'Iot-Lab'. The channel ID is 2749755, the author is mwa0000021829519, and the access is Public. The page has tabs for Private View, Public View, Channel Settings (selected), Sharing, API Keys, and Data Import / Export. The 'Channel Settings' section includes a 'Percentage Complete' of 50%, a 'Channel ID' of 2749755, a 'Name' field with 'Iot-Lab', and a 'Description' field with 'This is a channel for academic use for the Lab of Internet of Things'. There are eight 'Field' settings, each with a name field and a checkbox. Fields 1, 2, 3, and 8 are enabled (checkbox checked) and have names 'field1', 'field2', 'field3', and 'field8' respectively. Fields 4, 5, 6, and 7 are disabled (checkbox unchecked) and have empty name fields. To the right, there is a 'Help' section explaining channel data storage and a 'Channel Settings' section with detailed instructions for each setting: Percentage complete, Channel Name, Description, Field#, Metadata, Tags, Link to External Site, and Show Channel Location (Latitude, Longitude, Elevation).

Εικόνα 2. Μεταβλητές και ρυθμίσεις στο Thingspeak.

- **Channel ID:** Το κανάλι έχει τον μοναδικό αριθμό 2749755 και ονομάζεται Iot-Lab.
- **Author:** Το κανάλι δημιουργήθηκε από τον χρήστη με τον αναγνωριστικό αριθμό mwa0000021829519 και είναι διαθέσιμο ως Public για ακαδημαϊκή χρήση στο πλαίσιο του Εργαστηρίου Διαδικτύου των Αντικειμένων (IoT).

Το κανάλι έχει ορίσει οκτώ (8) πεδία, με τα τρία πρώτα να χρησιμοποιούνται για την παρακολούθηση των χρωμάτων του φωτεινού σηματοδότη και το Field 8 να χρησιμοποιείται ως Alert Signal.

- **Field 1 (field1):** Χρησιμοποιείται για την ένδειξη του κόκκινου σήματος (Red Signal).
- **Field 2 (field2):** Χρησιμοποιείται για την ένδειξη του πορτοκαλί σήματος (Orange Signal).

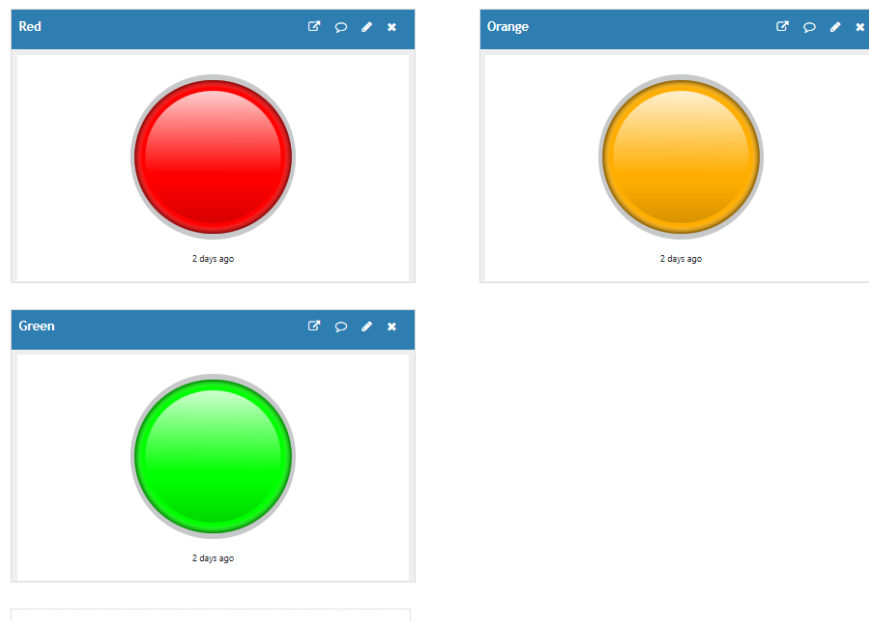
## ΔΙΑΔΙΚΤΥΟ ΤΩΝ ΑΝΤΙΚΕΙΜΕΝΩΝ

- **Field 3 (field3):** Χρησιμοποιείται για την ένδειξη του πράσινου σήματος (Green Signal).
- **Field 8 (field8):** Χρησιμοποιείται για την αποστολή και την παρακολούθηση του Alert Signal (Σήμα ειδοποίησης), που αποτελεί κανάλι παρακολούθησης για την λειτουργία του σηματοδότη.

Τα παραπάνω πεδία ενεργοποιούνται και ενημερώνονται μέσω του ESP-01, το οποίο είναι συνδεδεμένο με το Arduino, το οποίο με τη σειρά του ελέγχει τον φωτεινό σηματοδότη.

### Channel Stats

Created: 17 days ago  
Last entry: 2 days ago  
Entries: 287



**Εικόνα 3.** Φωτεινοί σηματοδότες στα πεδία fields.

## 4.2 Περιγραφή της χρήσης οπτικών στοιχείων

Για την καλύτερη κατανόηση και παρακολούθηση των δεδομένων, το κανάλι περιλαμβάνει διαγράμματα που αποτυπώνουν την κατάσταση του φωτεινού σηματοδότη σε πραγματικό χρόνο:

# ΔΙΑΔΙΚΤΥΟ ΤΩΝ ΑΝΤΙΚΕΙΜΕΝΩΝ

Red Options ? X

Name Red

Condition If

Field 1

is equal to

1

turn Lamp ON

Update Interval 30 second(s)

Color

Save Cancel

Εικόνα 4. Ρυθμίσεις για το κόκκινο σήμα.

- **Red Options Widget:**
  - **Όνομα:** Κόκκινο
  - **Κατάσταση:** Εάν το Field 1 είναι ίσο με 1, τότε το κόκκινο φως ανάβει.
  - **Χρόνος ενημέρωσης:** Το widget ενημερώνεται κάθε 30 δευτερόλεπτα.
  - **Χρώμα:** Το χρώμα που εμφανίζεται είναι κόκκινο για εύκολη αναγνώριση.

Orange Options ? X

Name Orange

Condition If

Field 2

is equal to

2

turn Lamp ON

Update Interval 20 second(s)

Color

Save Cancel

Εικόνα 5. Ρυθμίσεις για το πορτοκαλί σήμα.

- **Orange Options Widget:**
  - **Όνομα:** Πορτοκαλί

## ΔΙΑΔΙΚΤΥΟ ΤΩΝ ΑΝΤΙΚΕΙΜΕΝΩΝ

- **Κατάσταση:** Εάν το Field 2 είναι ίσο με 2, τότε το πορτοκαλί φως ανάβει.
- **Χρόνος ενημέρωσης:** Το widget ενημερώνεται κάθε 20 δευτερόλεπτα.
- **Χρώμα:** Το χρώμα που εμφανίζεται είναι πορτοκαλί.

Green Options ? X

Name

Condition If

▼

turn Lamp ON

Update Interval  second(s)

Color

Εικόνα 6. Ρυθμίσεις για το πράσινο σήμα.

- **Green Options Widget:**
  - **Name:** Green
  - **Κατάσταση:** Εάν το Field 3 είναι ίσο με 3, τότε το πράσινο φως ανάβει.
  - **Χρόνος ενημέρωσης:** Το widget ενημερώνεται κάθε 30 δευτερόλεπτα.
  - **Χρώμα:** Το χρώμα που εμφανίζεται είναι πράσινο.

- **Διαγράμματα μεταβλητών:**

Υπάρχουν τέσσερα διαγράμματα που αποτυπώνουν τα δεδομένα των πεδίων:

- **Field 1 Chart (Red Signal):** Απεικονίζει το ιστορικό των τιμών για το κόκκινο φως, το οποίο δείχνει πότε το φως είναι αναμμένο (τιμή 1).
- **Field 2 Chart (Orange Signal):** Απεικονίζει το ιστορικό των τιμών για το πορτοκαλί φως, το οποίο δείχνει πότε το φως είναι αναμμένο (τιμή 2).
- **Field 3 Chart (Green Signal):** Απεικονίζει το ιστορικό των τιμών για το πράσινο φως, το οποίο δείχνει πότε το φως είναι αναμμένο (τιμή 3).
- **Field 8 Chart (Alert Signal):** Δείχνει την κατάσταση του Alert Signal, που χρησιμοποιείται ως ειδοποίηση από το σύστημα (τιμή 0 ή 1) για την λειτουργία του φαναριού.

# ΔΙΑΔΙΚΤΥΟ ΤΩΝ ΑΝΤΙΚΕΙΜΕΝΩΝ



Εικόνα 7. Τα διαγράμματα που αποτυπώνουν τα δεδομένα των πεδίων.

## 5. Προγραμματιστικός Κώδικας

### 5.1 Ανάλυση του κώδικα που χρησιμοποιείται

Ο κώδικας υλοποιεί ένα σύστημα ελέγχου φωτεινού σηματοδότη μέσω του Arduino UNO που επικοινωνεί στο διαδίκτυο με το ESP-01. Το σύστημα επικοινωνεί με την πλατφόρμα ThingSpeak, επιτρέποντας την ανάγνωση και την εγγραφή δεδομένων μέσω πεδίων καναλιών (fields). Περιλαμβάνει λειτουργίες για την διαχείριση των οπτικών στοιχείων του σηματοδότη, καθώς και δυνατότητα παρακολούθησης της λειτουργίας του σηματοδότη μέσω ενός πεδίου καναλιού που ανάλογα με τις τιμές που διαβάζει θέτει εκτός λειτουργίας τον σηματοδότη για κάποιο χρονικό διάστημα (alert field).

Τα βασικά χαρακτηριστικά του κώδικα είναι τα εξής:

- **Επικοινωνία μέσω ESP-01:** Χρησιμοποιείται για την αποστολή και λήψη δεδομένων από το ThingSpeak.
- **Διαχείριση φωτεινού σηματοδότη:** Ρυθμίζει τα χρώματα του φωτεινού σηματοδότη (κόκκινο, πράσινο, πορτοκαλί) με καθορισμένες χρονικές καθυστερήσεις.
- **Κατάσταση εκτός λειτουργίας (out of order):** Ελέγχει και ενημερώνει το πεδίο field8 ώστε να αλλάζει η λειτουργία του φαναριού σε πορτοκαλί αν χρειαστεί.

## ΔΙΑΔΙΚΤΥΟ ΤΩΝ ΑΝΤΙΚΕΙΜΕΝΩΝ

- **Περιοδικές ενημερώσεις:** Κάθε 10 λεπτά (600.000 ms) ενημερώνεται το πεδίο field8 στο ThingSpeak με την τιμή 1 για να ελεγχθεί η λειτουργία κατάστασης «out of order».

### 5.2 Σχολιασμός και εξήγηση για το τι κάνει κάθε κομμάτι του κώδικα

Ακολουθεί ανάλυση των βασικών τμημάτων του κώδικα:

#### Διαμόρφωση αρχικών παραμέτρων

```
String mySSID = "linksys_1";
String myPWD = "";
String myHOST = "api.thingspeak.com";
String myPORT = "80";
String myWriteAPI = "6UKE7N1W16R0TIOC";
String myReadAPI = "MCX66QIC4S10R75T";
String myCHANNEL = "2749755";
```

Ορίζονται οι βασικές παράμετροι σύνδεσης στο WiFi και το ThingSpeak.

- **mySSID** □ όνομα του WiFi δικτύου
- **myPWD** □ κωδικός πρόσβασης για το WiFi δίκτυο
- **myHOST** □ όνομα του host που θα συνδεθεί το Arduino (ThingSpeak)
- **myPORT** □ πόρτα που θα γίνει ο διάυλος επικοινωνίας
- **myWriteAPI** □ API κλειδί όπου αποκτούμε πρόσβαση για αποστολή δεδομένων
- **myReadAPI** □ API κλειδί όπου αποκτούμε πρόσβαση για ανάγνωση δεδομένων
- **myCHANNEL** □ ID καναλιού που θα γίνει η ανταλλαγή πληροφορίας

#### Ρυθμίσεις ESP-01

```
espData("AT+RST", 1000, DEBUG);
espData("AT+CWMODE=1", 1000, DEBUG);
espData("AT+CWJAP=\"" + mySSID + "\",\"" + myPWD + "\"", 1000, DEBUG);

while(!espSerial.find("WIFI GOT IP"))
{
    Serial.print(".");
    delay(1000);
    Serial.print(".");
    delay(1000);
    Serial.print(".");
    delay(1000);
}
```



## ΔΙΑΔΙΚΤΥΟ ΤΩΝ ΑΝΤΙΚΕΙΜΕΝΩΝ

```
Serial.println("Connected!");  
delay(1000);
```

Ορίζονται οι βασικές ρυθμίσεις σύνδεσης του ESP-01 με το Arduino UNO

- **AT+RST** □ Το ESP-01 επανεκκινείται για προσπάθεια επανασύνδεσης
- **AT+CWMODE=1** □ Ορισμός σε λειτουργία client για σύνδεση
- **AT+CWJAP="mySSID","myPWD"** □ Σύνδεση στο WiFi δίκτυο με όνομα mySSID και κωδικό πρόσβασης myPWD

Η συνάρτηση espData() είναι υπεύθυνη για την αποστολή AT εντολών στο ESP-01 μέσω σειριακής επικοινωνίας και την ανάγνωση της ανταπόκρισης του.

```
String espData(String command, const int timeout, boolean debug)  
{  
    Serial.print("AT Command ==> ");  
    Serial.println(command);  
  
    response = "";  
    espSerial.println(command);  
    long int time = millis();  
    while ((time + timeout) > millis())  
    {  
        while (espSerial.available())  
        {  
            char c = espSerial.read();  
            response += c;  
        }  
    }  
    if (debug)  
    {  
        // Serial.print(response);  
    }  
  
    return response;  
}
```

### Διαχείριση οπτικών στοιχείων (Ζητούμενο Α4)

```
setTrafficLight("RED");  
delay(DELAY_RED);  
setFieldValue(fieldRed, myWriteAPI, 0);  
  
setTrafficLight("GREEN");  
delay(DELAY_GREEN);  
setFieldValue(fieldGreen, myWriteAPI, 0);
```

## ΔΙΑΔΙΚΤΥΟ ΤΩΝ ΑΝΤΙΚΕΙΜΕΝΩΝ

```
setTrafficLight("ORANGE");  
delay(DELAY_ORANGE);  
setFieldValue(fieldOrange, myWriteAPI, 0);
```

Η συνάρτηση setTrafficLight() καθορίζει ποιο χρώμα θα ενεργοποιηθεί:

- **RED** ➤ Όταν στέλνουμε την τιμή 1 στο field1, τότε ανάβει το κόκκινο φως
- **ORANGE** ➤ Όταν στέλνουμε την τιμή 2 στο field2, τότε ανάβει το πορτοκαλί φως
- **GREEN** ➤ Όταν στέλνουμε την τιμή 3 στο field3, τότε ανάβει το πράσινο φως

Οι ενδείξεις ανάβουν με μεσολάβηση καθορισμένων χρονικών καθυστερήσεων που ορίζονται στις αντίστοιχες μεταβλητές. Συγκεκριμένα, 30 sec. για το κόκκινο και το πράσινο και 20 sec για το πορτοκαλί. Με αποστολή της τιμής 0 στα αντίστοιχα πεδία, σβήνουμε την ένδειξη.

Η τιμή αποστέλλεται μέσω της ρουτίνας setFieldValue() που θα αναλυθεί αργότερα.

```
void setTrafficLight(String color)  
{  
    String field;  
    if (color == "RED")  
    {  
        field = fieldRed;  
        sendVal = 1;  
    }  
    else if (color == "ORANGE")  
    {  
        field = fieldOrange;  
        sendVal = 2;  
    }  
    else if (color == "GREEN")  
    {  
        field = fieldGreen;  
        sendVal = 3;  
    }  
    else return;  
  
    setFieldValue(field, myWriteAPI, sendVal);  
    Serial.println("Traffic Light is set to " + color);  
}
```

### [Αποστολή δεδομένων στο ThingSpeak \(Ζητούμενο B\)](#)

```
setFieldValue(fieldAlert, myWriteAPI, 0);
```

## ΔΙΑΔΙΚΤΥΟ ΤΩΝ ΑΝΤΙΚΕΙΜΕΝΩΝ

```
Serial.println("ALERT Field set to 0.");
```

Η συνάρτηση `setFieldValue()` χρησιμοποιεί το κατάλληλο HTTP GET request για να αποστείλει δεδομένα στο πεδίο `field8` που είναι υπεύθυνο για την παρακολούθηση της λειτουργίας του σηματοδότη. Μέσω TCP συνδέεται στο API του ThingSpeak και αποστέλλει την τιμή 0 στο πεδίο `field8` που σηματοδοτεί την κατάσταση λειτουργίας του σηματοδότη. Με την βοήθεια της ρουτίνας `espData()` αποστέλλονται οι κατάλληλες AT εντολές για την αποστολή. Συγκεκριμένα αποστέλλονται:

- **AT+CIPMUX=1** □ Ενεργοποίηση λειτουργίας πολλαπλών συνδέσεων στο ESP-01
- **AT+CIPSTART=0, "TCP", "myHOST", "myPORT"** □ Εκκίνηση TCP σύνδεσης με τον απομακρυσμένο διακομιστή (στην περίπτωση μας στο ThingSpeak)
- **AT+CIPSEND=0, "request.length"** □ Προετοιμασία του ESP-01 για την αποστολή δεδομένων μέσω μιας ανοιχτής TCP σύνδεσης
- **AT+CIPCLOSE=0** □ Τερματισμός της TCP σύνδεσης

```
void setFieldValue(String field, String writeAPI, int value)
{
    sendData = "GET /update?api_key=" + writeAPI + "&" + field + "=" +
String(value);
    espData("AT+CIPMUX=1", 1000, DEBUG);
    espData("AT+CIPSTART=0,\"TCP\", \"\" + myHOST + "\",\" + myPORT, 1000,
DEBUG);
    espData("AT+CIPSEND=0,\" + String(sendData.length() + 4), 1000, DEBUG);
    espSerial.find(">");
    espSerial.println(sendData);
    Serial.println("Value to be sent: ");
    Serial.println(value);

    espData("AT+CIPCLOSE=0", 1000, true);
    delay(10000);
}
```

### Ανάγνωση δεδομένων από το ThingSpeak (Ζητούμενο Γ1)

```
x01 = getFieldValue(fieldAlert); // Ανάγνωση τιμής του πεδίου field8
Serial.println("Value of ALERT field:" +x01);

if (x01.equals("1"))
{
    Serial.println("ALERT: Traffic Light out of order, setting the ORANGE
signal.");
    setTrafficLight("ORANGE");
}
```

## ΔΙΑΔΙΚΤΥΟ ΤΩΝ ΑΝΤΙΚΕΙΜΕΝΩΝ

Η συνάρτηση `getFieldValue()` χρησιμοποιεί το κατάλληλο HTTP GET request για να διαβάσει δεδομένα από πεδίο του ThingSpeak. Αναζητά τη θέση του πεδίου στην απόκριση JSON και επιστρέφει την αντίστοιχη τιμή. Το πεδίο που διαβάζει είναι το `field8` που είναι το πεδίο που ελέγχει την λειτουργία του φαναριού. Εάν, η τιμή που διαβάζει είναι 1, τότε θέτει με την ρουτίνα `setTrafficLight()` το φανάρι εκτός λειτουργίας ανάβοντας την πορτοκαλί ένδειξη. Με την βοήθεια της ρουτίνας `espData()` αποστέλλονται οι κατάλληλες AT εντολές για την ανάγνωση. Συγκεκριμένα αποστέλλονται:

- **AT+CIPMUX=1** □ Ενεργοποίηση λειτουργίας πολλαπλών συνδέσεων στο ESP-01
- **AT+CIPSTART=0, "TCP", "myHOST", "myPORT"** □ Εκκίνηση TCP σύνδεσης με τον απομακρυσμένο διακομιστή (στην περίπτωση μας στο ThingSpeak)
- **AT+CIPSTO=10** □ Ρυθμίζει τον χρόνο timeout για αδρανείς TCP συνδέσεις
- **AT+CIPSEND=0, "request.length"** □ Προετοιμασία του ESP-01 για την αποστολή δεδομένων μέσω μιας ανοιχτής TCP σύνδεσης
- **AT+CIPCLOSE=0** □ Τερματισμός της TCP σύνδεσης

### Περιοδική ενημέρωση της τιμής του πεδίου field8 (Ζητούμενο Γ2)

```
currentMillis = millis();
if (currentMillis - previousMillis >= timer)
{
    // Θέτουμε την τιμή 1 στο πεδίο field8
    previousMillis = currentMillis;
    Serial.println("Setting the ALERT field to 1.");
    setFieldValue(fieldAlert, myWriteAPI, 1);

    // Θέτουμε το φανάρι σε κατάσταση εκτός λειτουργίας
    Serial.println("ALERT: Traffic Light out of order, setting the ORANGE
signal.");
    setTrafficLight("ORANGE");

    // Για 1 λεπτό το φανάρι τίθεται εκτός λειτουργίας
    delay(duration);

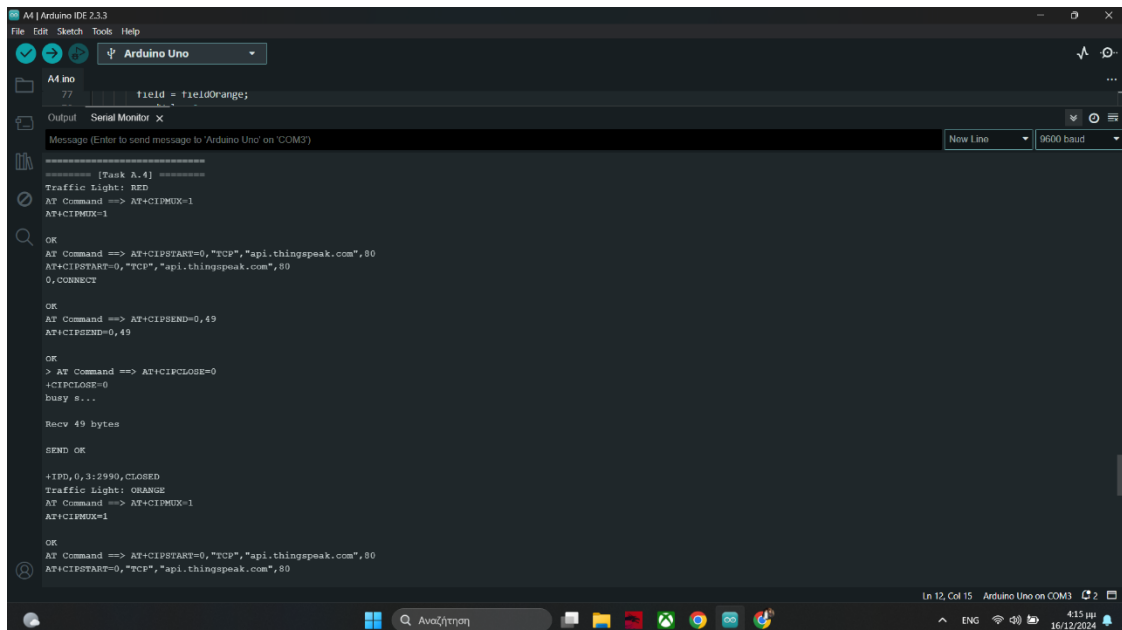
    // Το φανάρι τίθεται ξανά σε λειτουργία
    setFieldValue(fieldAlert, myWriteAPI, 0);
}
```

Κάθε 10 λεπτά (600.000 ms) η τιμή του `field8` ενημερώνεται με 1 και διαρκεί για 1 λεπτό. Αυτό γίνεται για να ελέγξουμε αν διαβάζεται σωστά η τιμή του `field8` και τίθεται εκτός λειτουργίας το φανάρι ανάβοντας την πορτοκαλί ένδειξη. Η αποστολή της τιμής γίνεται με τη βοήθεια της ρουτίνας `setFieldValue()`.

## 5.3 Παρουσίαση αποτελεσμάτων

Για όσο διάστημα είχαμε τον εξοπλισμό στον χώρο του εργαστηρίου πήραμε τα εξής αποτελέσματα:

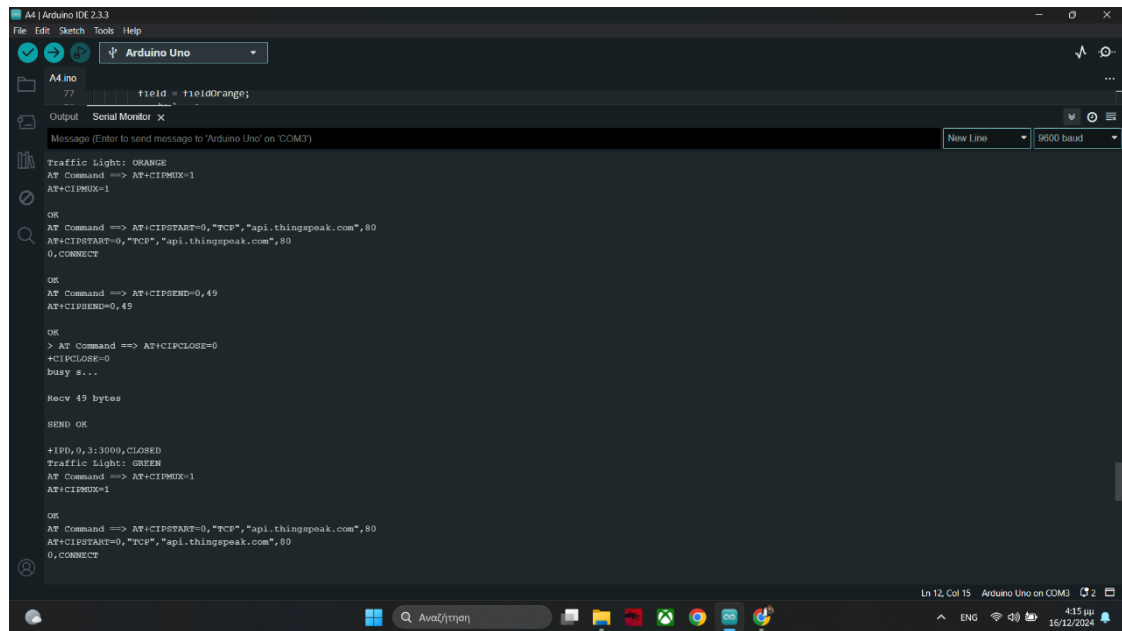
- Την κανονική λειτουργία του φαναριού, όπου εναλλάσσονται μεταξύ κόκκινου, πράσινου και πορτοκαλί για καθορισμένα διαστήματα.
- Κατάσταση λειτουργίας (Alert) όπου το πεδίο ενημερώνεται αυτόματα κάθε 10 λεπτά και αν διαβαστεί η τιμή 1, τότε το φανάρι τίθεται εκτός λειτουργίας.
- Οι εντολές AT και οι αποκρίσεις του ESP-01 για την επικοινωνία του με το Arduino UNO.



```
Arduino IDE 2.3.3
File Edit Sketch Tools Help
Arduino Uno
77 field = fieldOrange;
Output Serial Monitor x
Message (Enter to send message to 'Arduino Uno' on 'COM3')
New Line 9600 baud
===== [Task A:4] =====
Traffic Light: RED
AT Command ==> AT+CIWMUX=1
AT+CIWMUX=1
OK
AT Command ==> AT+CIPSTART=0,"TCP","api.thingspeak.com",80
AT+CIPSTART=0,"TCP","api.thingspeak.com",80
0,CONNECT
OK
AT Command ==> AT+CIPSEND=0,49
AT+CIPSEND=0,49
OK
> AT Command ==> AT+CIPCLOSE=0
+CIPCLOSE=0
busy ...
Recv 49 bytes
SEND OK
+IPD,0,3:2990,CLOSED
Traffic Light: ORANGE
AT Command ==> AT+CIWMUX=1
AT+CIWMUX=1
OK
AT Command ==> AT+CIPSTART=0,"TCP","api.thingspeak.com",80
AT+CIPSTART=0,"TCP","api.thingspeak.com",80
```

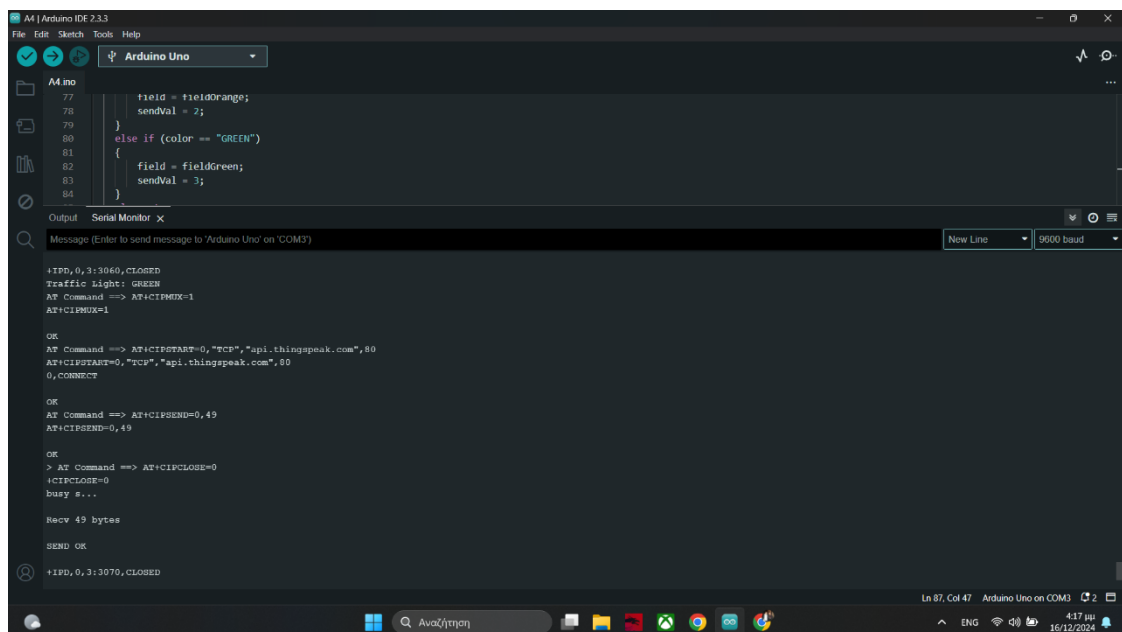
Εικόνα 8. Ζητούμενο A4

# ΔΙΑΔΙΚΤΥΟ ΤΩΝ ΑΝΤΙΚΕΙΜΕΝΩΝ



```
Arduino IDE 2.3.3
File Edit Sketch Tools Help
ψ Arduino Uno
A4.ino
77 field = fieldOrange;
Output Serial Monitor X
Message (Enter to send message to 'Arduino Uno' on 'COM3')
New Line 9600 baud
Traffic Light: ORANGE
AT Command ==> AT+CIPIPMUX=1
AT+CIPIPMUX=1
OK
AT Command ==> AT+CIPISTART=0,"TCP","api.thingspeak.com",80
AT+CIPISTART=0,"TCP","api.thingspeak.com",80
0,CONNECT
OK
AT Command ==> AT+CIPISEND=0,49
AT+CIPISEND=0,49
OK
> AT Command ==> AT+CIPICLOSE=0
+CIPICLOSE=0
busy s...
Recv 49 bytes
SEND OK
+IPD,0,3:3000,CLOSED
Traffic Light: GREEN
AT Command ==> AT+CIPIPMUX=1
AT+CIPIPMUX=1
OK
AT Command ==> AT+CIPISTART=0,"TCP","api.thingspeak.com",80
AT+CIPISTART=0,"TCP","api.thingspeak.com",80
0,CONNECT
Ln 12, Col 15 Arduino Uno on COM3 415 μs 16/12/2024
```

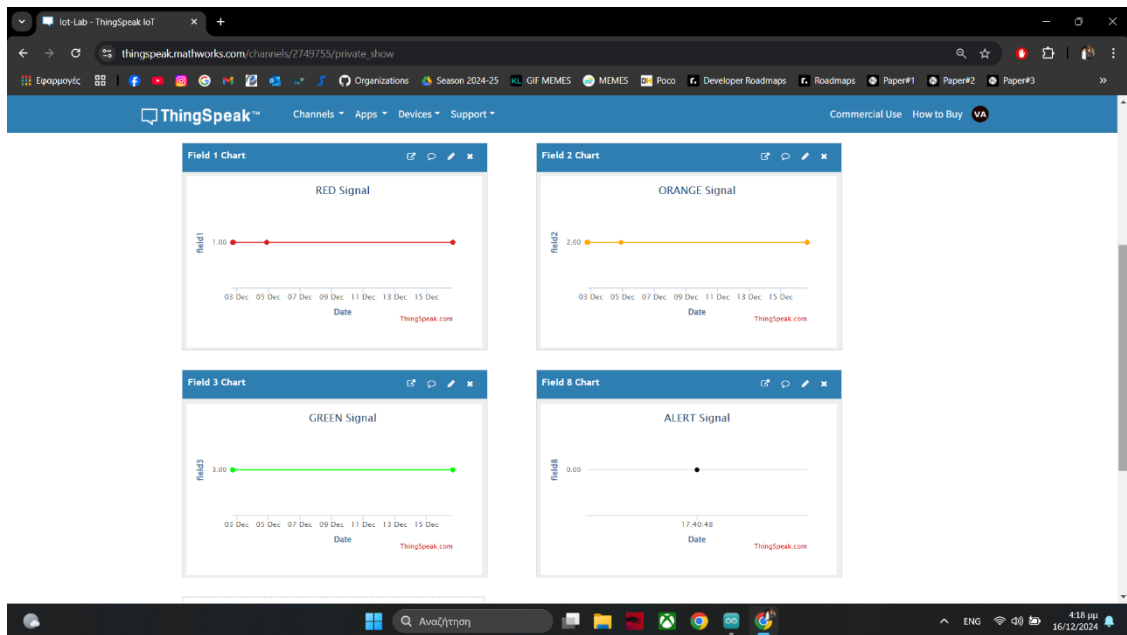
Εικόνα 9. Ζητούμενο Α4 (συνέχεια)



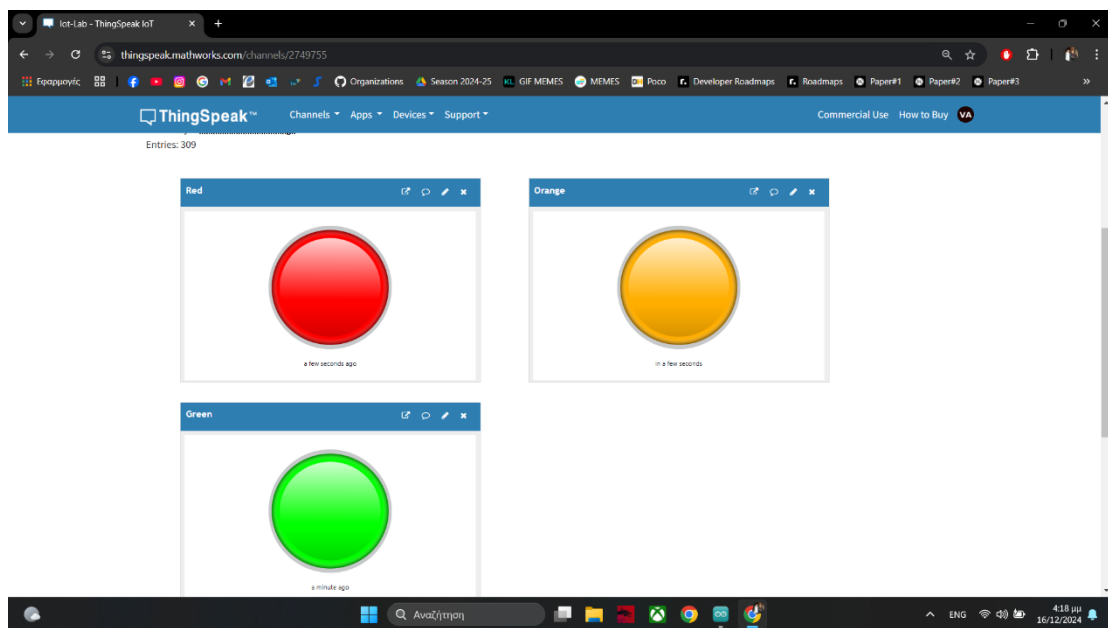
```
Arduino IDE 2.3.3
File Edit Sketch Tools Help
ψ Arduino Uno
A4.ino
77 field = fieldOrange;
78 sendVal = 2;
79 }
80 else if (color == "GREEN")
81 {
82 field = fieldGreen;
83 sendVal = 3;
84 }
Output Serial Monitor X
Message (Enter to send message to 'Arduino Uno' on 'COM3')
New Line 9600 baud
+IPD,0,3:3060,CLOSED
Traffic Light: GREEN
AT Command ==> AT+CIPIPMUX=1
AT+CIPIPMUX=1
OK
AT Command ==> AT+CIPISTART=0,"TCP","api.thingspeak.com",80
AT+CIPISTART=0,"TCP","api.thingspeak.com",80
0,CONNECT
OK
AT Command ==> AT+CIPISEND=0,49
AT+CIPISEND=0,49
OK
> AT Command ==> AT+CIPICLOSE=0
+CIPICLOSE=0
busy s...
Recv 49 bytes
SEND OK
+IPD,0,3:3070,CLOSED
Ln 87, Col 47 Arduino Uno on COM3 417 μs 16/12/2024
```

Εικόνα 10. Ζητούμενο Α4 (συνέχεια)

# ΔΙΑΔΙΚΤΥΟ ΤΩΝ ΑΝΤΙΚΕΙΜΕΝΩΝ

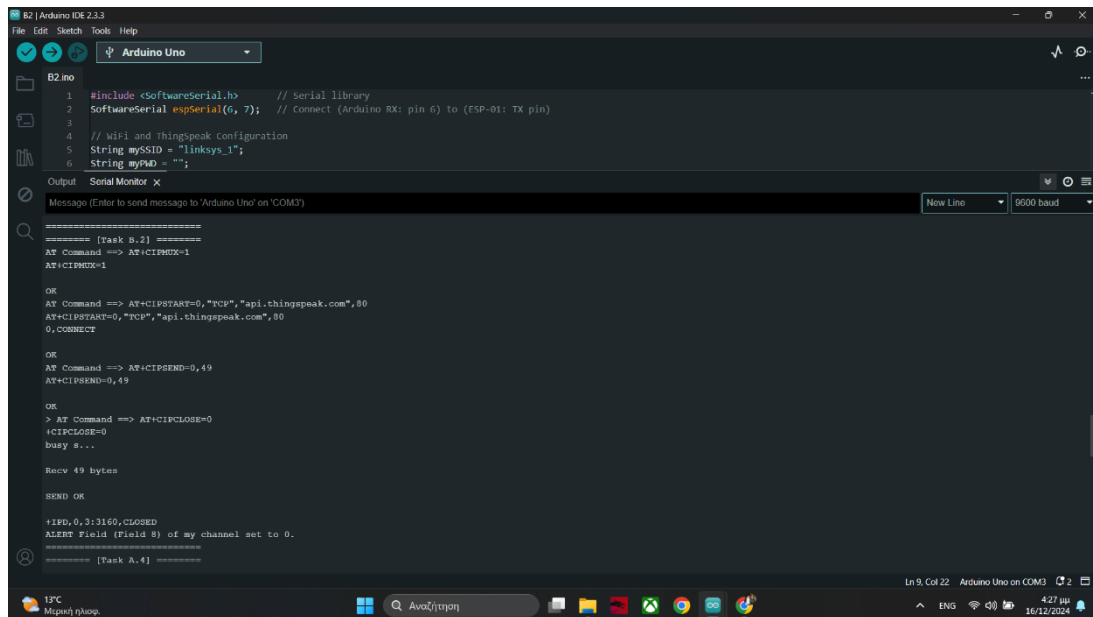


Εικόνα 11. Τα δεδομένα στα πεδία των οπτικών στοιχείων στο ThingSpeak

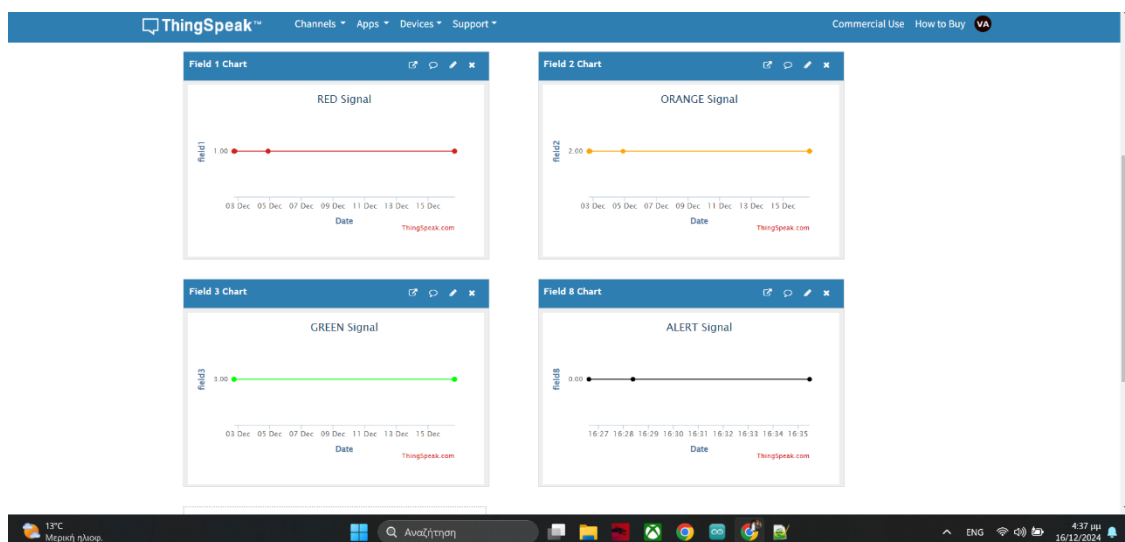


Εικόνα 12. Τα οπτικά στοιχεία στο ThingSpeak

## ΔΙΑΔΙΚΤΥΟ ΤΩΝ ΑΝΤΙΚΕΙΜΕΝΩΝ



**Εικόνα 13.** Ζητούμενο Β



**Εικόνα 14.** Τα δεδομένα του πεδίου κατάστασης λειτουργίας (alert) στο ThingSpeak

## 5.4 Προβλήματα που εντοπίστηκαν και λύσεις που δοκιμάστηκαν

Το κύριο πρόβλημα που αντιμετωπίσαμε ήταν στο ζητούμενο Γ1 όπου διαβάσαμε την τιμή του πεδίου field8 που παρακολουθεί την κατάσταση λειτουργίας του σηματοδότη. Ενώ, από το ζητούμενο Β στέλναμε προκαθορισμένα την τιμή 0 που σηματοδοτεί την κατάσταση έγκυρης λειτουργίας, η απόκριση JSON δεν επέστρεφε καμία τιμή.

Δοκιμάσαμε να αυξήσουμε τα delays ανάμεσα στα HTTP GET αιτήματα που κάναμε στα προηγούμενα ζητούμενα, μήπως υπήρχε θέμα στην ενημέρωση της τιμής, ωστόσο δεν



## ΔΙΑΔΙΚΤΥΟ ΤΩΝ ΑΝΤΙΚΕΙΜΕΝΩΝ

άλλαξε το αποτέλεσμα. Η τιμή 0 όπως φαίνεται και στα αποτελέσματα υπήρχε κανονικά στα δεδομένα του πεδίου field8. Αλλάξαμε επίσης και την συνθήκη

```
if (x01 == 1)
```

σε

```
if (x01.equals("1"))
```

μήπως έφταιγε ότι η τιμή αποθηκευόταν ως String. Εν τέλει, από το τελευταίο εργαστήριο δεν καταφέραμε να διορθώσουμε το πρόβλημα αυτό, οπότε το αφήσαμε όπως έχει, λόγω απουσίας εξοπλισμού.

## 6. Συμπεράσματα

### 6.1 Ανασκόπηση της εργασίας

Η εργασία αποτελεί σημαντικό βήμα για την ανάπτυξη ενός IoT συστήματος και παρακολούθησης των δεδομένων σε πραγματικό χρόνο (real-time). Πέτυχε όλους τους σκοπούς που προαναφέραμε και ανέδειξε την δυναμική του Internet of Things. Αποκτήσαμε την κατάλληλη εμπειρία και από την ανάπτυξη και μέσα από τις αστοχίες, όπου προβληματιστήκαμε και δοκιμάσαμε λύσεις.

### 6.2 Προτάσεις για βελτιώσεις

Οι βελτιώσεις που θα προτείναμε θα ήταν στον περιορισμό των HTTP αιτημάτων, ώστε να μεσολαβεί ένα χρονικό διάστημα όπου θα ανανεώνονται οι τιμές στα πεδία και να έχουμε ορθά αποτελέσματα για την λειτουργία του σηματοδότη. Όσον αφορά άλλα θέματα επεκτασιμότητας, θα μπορούσαμε να χρησιμοποιήσουμε HTTPS αιτήματα για κρυπτογράφηση, ρουτίνα επανασύνδεσης σε περιπτώσεις διακοπής της WiFi σύνδεσης και διάφορα άλλα τα οποία δεν είναι στα πλαίσια της εργασίας, αλλά καλό θα ήταν να εφαρμοστούν για την βελτίωση της αποδοτικότητας του συστήματος μας.

# ΔΙΑΔΙΚΤΥΟ ΤΩΝ ΑΝΤΙΚΕΙΜΕΝΩΝ



Σας ευχαριστούμε για την προσοχή σας.

