

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

# ΠΕΡΙΕΧΟΜΕΝΑ

[**1. Εισαγωγή 3**](#_heading=h.3znysh7)

[1.1 Στόχοι της εργασίας 3](#_heading=h.2et92p0)

[1.2 Αναφορά στις βασικές έννοιες (μικροελεγκτές, IoT, ThingSpeak κλπ.) 3](#_heading=h.tyjcwt)

[**2. Ζητούμενα και Ανάλυση 3**](#_heading=h.3dy6vkm)

[**2.1 Ζητούμενο Α: Ηλεκτρικός φωτεινός σηματοδότης 3**](#_heading=h.1t3h5sf)

[2.1.1 Περιγραφή λειτουργιών 3](#_heading=h.4d34og8)

[2.1.2 Χρήση του Arduino και πλατφόρμας ThingSpeak 3](#_heading=h.2s8eyo1)

[2.1.3 Απαιτήσεις για προγραμματισμό και συνδεσιμότητα 3](#_heading=h.17dp8vu)

[**2.2 Ζητούμενο Β: Αποστολή δεδομένων σε κανάλι άλλης εφαρμογής 3**](#_heading=h.3rdcrjn)

[2.2.1 Περιγραφή στόχων και λειτουργιών 3](#_heading=h.26in1rg)

[2.2.2 Ανάλυση προγραμματιστικών βημάτων 3](#_heading=h.lnxbz9)

[**2.3 Ζητούμενο Γ: Ανάγνωση δεδομένων από κανάλι άλλης εφαρμογής 3**](#_heading=h.35nkun2)

[2.3.1 Περιγραφή απαιτούμενων ρυθμίσεων και λειτουργιών 3](#_heading=h.1ksv4uv)

[2.3.2 Ανάλυση προγραμματιστικών βημάτων 3](#_heading=h.44sinio)

[**3. Περιγραφή Κυκλώματος Υλικού 4**](#_heading=h.2jxsxqh)

[3.1 Ανάλυση συνδεσμολογίας 4](#_heading=h.z337ya)

[3.2 Χρησιμοποιούμενα εξαρτήματα και ο ρόλος τους (π.χ. ESP-01, LEDs) 4](#_heading=h.3j2qqm3)

[3.3 Διάγραμμα κυκλώματος 4](#_heading=h.1y810tw)

[**4. Ρυθμίσεις Καναλιού ThingSpeak 4**](#_heading=h.4i7ojhp)

[4.1 Επεξήγηση των μεταβλητών και των ρυθμίσεων 4](#_heading=h.2xcytpi)

[4.2 Περιγραφή της χρήσης οπτικών στοιχείων 4](#_heading=h.1ci93xb)

[**5. Προγραμματιστικός Κώδικας 4**](#_heading=h.3whwml4)

[5.1 Ανάλυση του κώδικα που χρησιμοποιείται 4](#_heading=h.2bn6wsx)

[5.2 Σχολιασμός και εξήγηση για το τι κάνει κάθε κομμάτι του κώδικα 4](#_heading=h.qsh70q)

[**6. Συμπεράσματα 4**](#_heading=h.3as4poj)

[6.1 Ανασκόπηση της εργασίας 4](#_heading=h.1pxezwc)

[6.2 Προτάσεις για βελτιώσεις ή μελλοντική επέκταση του έργου 4](#_heading=h.49x2ik5)

# 1. Εισαγωγή

## 1.1 Στόχοι της εργασίας

Ο κύριος στόχος αυτής της εργασίας είναι η απόκτηση πρακτικών γνώσεων και δεξιοτήτων στον τομέα των μικροελεγκτών και των συστημάτων Διαδικτύου των Αντικειμένων (IoT). Συγκεκριμένα, οι στόχοι περιλαμβάνουν:

* **Κατανόηση των μικροελεγκτών και της αλληλεπίδρασής τους με ηλεκτρονικά στοιχεία**: Συμπεριλαμβάνονται θέματα όπως οι αναλογικές και ψηφιακές είσοδοι/έξοδοι, η σειριακή επικοινωνία, καθώς και η χρήση και σύνδεση αισθητήρων και εξόδων (π.χ. LED).
* **Ανάπτυξη εφαρμογής IoT**: Η εργασία απαιτεί την ανάπτυξη και υλοποίηση ενός ηλεκτρικού φωτεινού σηματοδότη που λειτουργεί με Arduino και την ενσωμάτωσή του στην πλατφόρμα ThingSpeak.
* **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 μεταξύ διαφορετικών καναλιών**: Ενσωμάτωση προγραμματιστικών δυνατοτήτων για τη συνεργασία διαφορετικών εφαρμογών IoT.

Η εργασία στοχεύει στην πλήρη υλοποίηση ενός παραδείγματος λειτουργικού IoT συστήματος όπως είναι αυτό του ηλεκτρικού φωτεινού σηματοδότη, προωθώντας τη χρήση σύγχρονων τεχνολογιών και πλατφορμών.

## 1.2 Αναφορά στις βασικές έννοιες

Οι βασικές έννοιες που χρησιμοποιούνται στην εργασία του ηλεκτρικού φωτεινού σηματοδότη περιλαμβάνουν:

* **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 δημιουργια 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 αναλυτικα το καθενα:

**Arduino UNO**

To **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 χρονοδιακόπτες που θα μετρούν τη διάρκεια κάθε φάσης του σηματοδότη (κόκκινο, πράσινο, πορτοκαλί). Αυτό επιτυγχάνεται με τη χρήση της εντολής delay() ή άλλων τεχνικών για τον προγραμματισμό χρονοκαθυστέρησης.
3. **WiFi Συνδεσιμότητα**: Ο Arduino πρέπει να συνδέεται σε WiFi δίκτυο μέσω του **ESP-01** WiFi module, 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:

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 οι υπόλοιπες τιμές είναι 0.
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 είτε μέσω **HTTP αιτημάτων**

**Διασύνδεση και Μετάδοση Δεδομένων**

Η σύνδεση του Arduino με την πλατφόρμα ThingSpeak επιτυγχάνεται μέσω του **ESP-01** WiFi module, το οποίο επιτρέπει τη μετάδοση των δεδομένων από τον μικροελεγκτή στο διαδικτυακό περιβάλλον. 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 τη συνεχή ενημέρωση των χρηστών.

### 2.1.3 Απαιτήσεις για προγραμματισμό και συνδεσιμότητα

Για την υλοποίηση του συστήματος 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 (>=30 δευτερόλεπτα για κόκκινο και πράσινο, >=20 δευτερόλεπτα για πορτοκαλί).
2. **Ρυθμίζει τις Χρονικές Καθυστερήσεις**: Η διάρκεια κάθε φάσης του σηματοδότη ελέγχεται με τη χρήση χρονοκαθυστέρησης, 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 πλησιέστερο **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.

## 2.2 Ζητούμενο Β: Αποστολή δεδομένων σε κανάλι άλλης εφαρμογής

### 2.2.1 Περιγραφή στόχων και λειτουργιών

Ο κύριος στόχος του Ζητούμενου είναι η ανάπτυξη μιας λειτουργικότητας που επιτρέπει την επικοινωνία μεταξύ δύο διαφορετικών καναλιών στην πλατφόρμα ThingSpeak. Η λειτουργικότητα αυτή αφορά την αποστολή δεδομένων από το κανάλι του φωτεινού σηματοδότη στο κανάλι μιας άλλης εφαρμογής και περιλαμβάνει τα εξής:

**Αποστολή δεδομένων σε απομακρυσμένο κανάλι**:

* + Ορίζεται η τιμή **0** στη μεταβλητή **Field 8** του καναλιού της άλλης εφαρμογής, χρησιμοποιώντας HTTP αιτήματα για την επικοινωνία με το API του ThingSpeak.
  + Η ενέργεια αυτή αποσκοπεί στη δυνατότητα αλληλεπίδρασης με άλλα IoT συστήματα, παρέχοντας πληροφορίες ή προκαθορισμένες τιμές.

**Συγχρονισμός δεδομένων με το τοπικό κανάλι**:

* + Για να διασφαλιστεί η συνοχή των δεδομένων, η ίδια τιμή (**0**) που αποστέλλεται στο **Field 8** του καναλιού της άλλης εφαρμογής αποστέλλεται και στο **Field 8** του καναλιού του φωτεινού σηματοδότη μας.

**Διαμοιρασμός πληροφοριών μεταξύ συστημάτων IoT**:

* + Η λειτουργία αυτή προάγει τη συνεργασία μεταξύ διαφορετικών IoT εφαρμογών, επιτρέποντας την ανταλλαγή δεδομένων σε πραγματικό χρόνο.

**Ευελιξία για μελλοντική επέκταση**:

* + Η συγκεκριμένη υλοποίηση μπορεί να προσαρμοστεί για την αποστολή διαφορετικών δεδομένων ή την επικοινωνία με περισσότερα κανάλια, 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 σε κανάλι άλλης εφαρμογής, 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 για το άλλο κανάλι**:

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

**Αποστολή των τιμών στο τοπικό κανάλι**:

* Όταν ενημερώνεται το **Field 8** της άλλης εφαρμογής, η ίδια τιμή αποστέλλεται και στο **Field 8** του δικού μας καναλιού.

**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 άλλης εφαρμογής

### 2.3.1 Περιγραφή απαιτούμενων ρυθμίσεων και λειτουργιών

### 

Για την ανάγνωση των δεδομένων από το κανάλι του ThingSpeak 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.
  + 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

* **Διαμόρφωση σύνδεσης**
  + Για να διαμορφωθεί η σύνδεση αποστέλλονται οι κατάλληλες AT εντολές για την έναρξη της επικοινωνίας του Arduino UNO με το ThingSpeak

espData("AT+CIPMUX=1", 1000, true);  
espData("AT+CIPSTART=0,\"TCP\",\"" + myHOST + "\"," + myPORT, 1000, true);

* **Σύνταξη και Αποστολή αιτήματος HTTP GET**
  + Για να πάρουμε τα δεδομένα του καναλιού σε μορφή JSON αποστέλλουμε το παρακάτω GET URL

messager = "GET /channels/" + myCHANNEL + "/feeds.json?api\_key=" + myReadAPI + "&results=1";

* + Αποθηκεύουμε το μήκος του μηνύματος και στέλνουμε το αίτημα

espData("AT+CIPSEND=0,"+ String(messager.length() + 4), 1000, DEBUG);  
espSerial.find(">");  
espSerial.println(messager);

* **Λήψη και ανάλυση απάντησης**
  + Διαβάζουμε την απάντηση από το ESP-01 και την αποθηκεύουμε σε μια μεταβλητή

delay(1000);  
Serial.println(response);

* + Εντοπίζουμε και εξάγουμε την τιμή από το JSON χρησιμοποιώντας συναρτήσεις όπως **indexOf()** και **substring()**

**ret\_Len = response.length();  
pos = response.indexOf("[");  
res\_feeds = response.substring(pos, ret\_Len);  
Serial.print(res\_feeds);  
pos = res\_feeds.indexOf(myFIELDa);  
x01 = res\_feeds.substring(pos+9, pos+10);**

* **Χειρισμός Δεδομένων**
  + Ανάλογα με την τιμή που εξάγεται από το πεδίο, εκτέλεσε την αντίστοιχη λειτουργία, π.χ. άλλαξε τη φωτεινή ένδειξη ή ενεργοποίησε την ειδοποίηση

if (x01 == 1)  
{  
 Serial.println("ALERT: Traffic Light out of order.");  
 setTrafficLight("ORANGE");  
 delay(outOfOrderTimer);

}

* **Κλείσιμο Σύνδεσης**
  + Κλείσε τη σύνδεση TCP μετά την ολοκλήρωση του αιτήματος

espData("AT+CIPCLOSE=0", 1000, DEBUG);

# 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 Χρησιμοποιούμενα Εξαρτήματα και ο Ρόλος τους

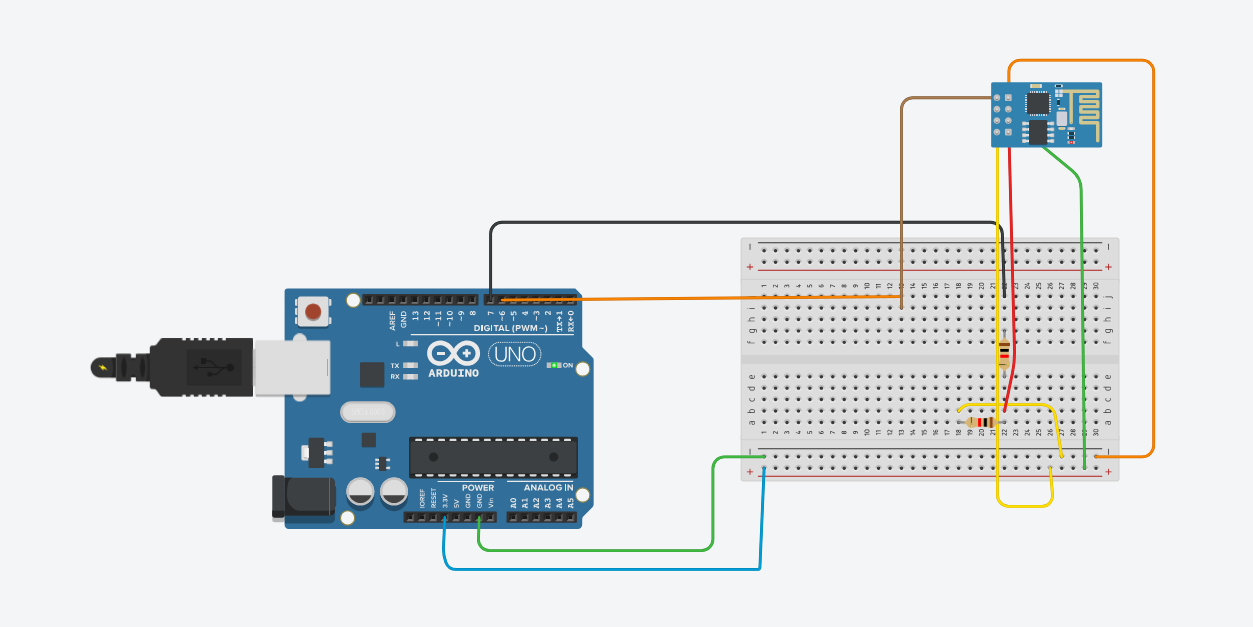
* **Arduino:** Ο μικροελεγκτής που είναι υπεύθυνος για τον έλεγχο των οπτικών στοιχείων και για τη διασύνδεση με την πλατφόρμα ThingSpeak.
* **ESP-01:** Το WiFi module που συνδέει τον Arduino στο δίκτυο και επιτρέπει την αποστολή δεδομένων στο ThingSpeak.
* **Αντιστάσεις:** Οι αντιστάσεις παίζουν σημαντικό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

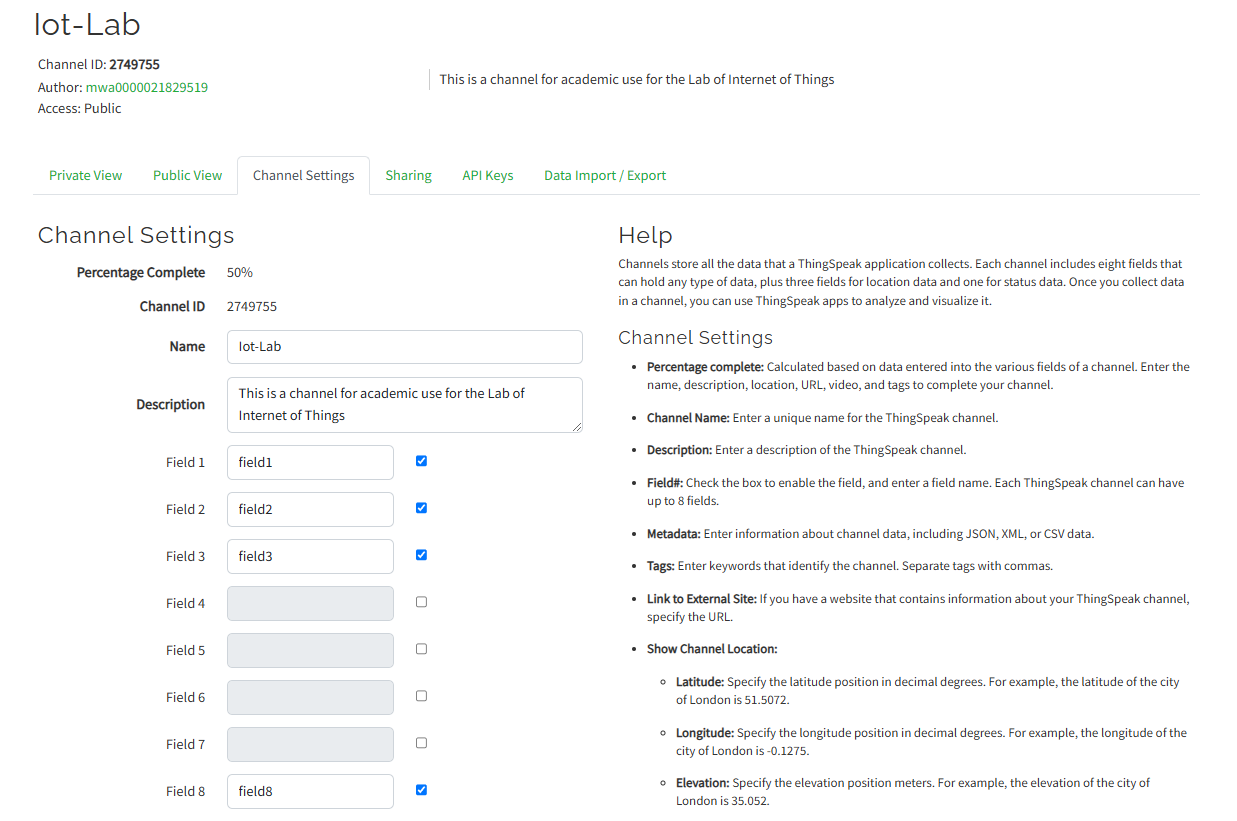


**Εικόνα 1.** Το κύκλωμα του Arduino UNO με το ESP8266 Wi-Fi module στο Tinkercad

# 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



**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 σηματοδότη (**Red Signal**).
* **Field 2 (field2)**: Χρησιμοποιείται για την ένδειξη του πορτοκαλί σηματοδότη (**Orange Signal**).
* **Field 3 (field3)**: Χρησιμοποιείται για την ένδειξη του πράσινου σηματοδότη (**Green Signal**).
* **Field 8 (field8)**: Χρησιμοποιείται για την αποστολή και την παρακολούθηση του **Alert Signal** (Σήμα ειδοποίησης).

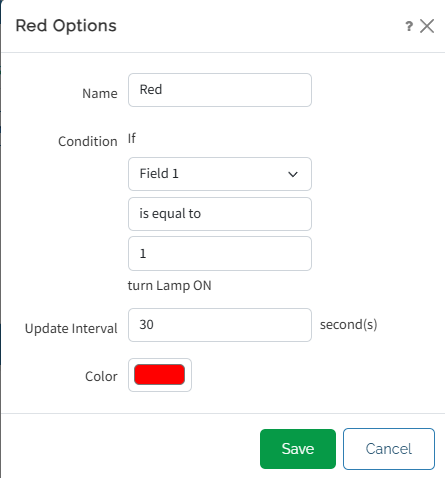
Τα παραπάνω πεδία ενεργοποιούνται και ενημερώνονται μέσω του ESP-01, το οποίο είναι συνδεδεμένο με το Arduino, το οποίο με τη σειρά του ελέγχει τον φωτεινό σηματοδότη.



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

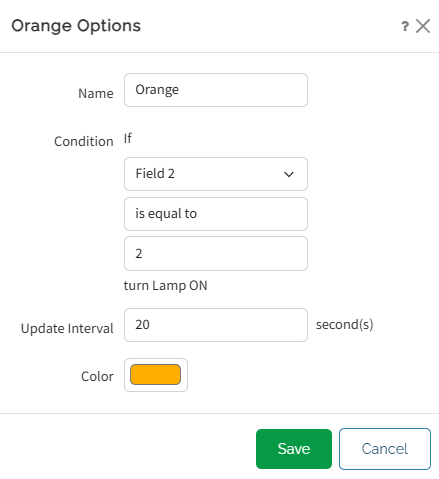
## 4.2 Περιγραφή της χρήσης οπτικών στοιχείων

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



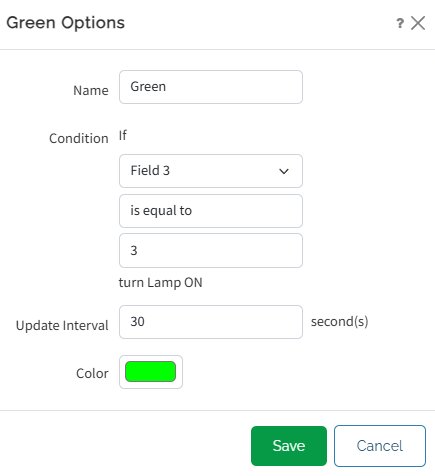
**Εικόνα 4.** Ρυθμίσεις για τον κόκκινο σηματοδότη.

* **Red Options Widget**:
  + **Name**: Red
  + **Condition**: Εάν το Field 1 είναι ίσο με 1, τότε το κόκκινο φως ανάβει.
  + **Update Interval**: Το widget ενημερώνεται κάθε **30 δευτερόλεπτα**.
  + **Χρώμα**: Το χρώμα που εμφανίζεται είναι κόκκινο για εύκολη αναγνώριση.



**Εικόνα 5.** Ρυθμίσεις για τον πορτοκαλί σηματοδότη.

* **Orange Options Widget**:
  + **Name**: Orange
  + **Condition**: Εάν το Field 2 είναι ίσο με 2, τότε το πορτοκαλί φως ανάβει.
  + **Update Interval**: Το widget ενημερώνεται κάθε **20 δευτερόλεπτα**.
  + **Χρώμα**: Το χρώμα που εμφανίζεται είναι πορτοκαλί.



**Εικόνα 6.** Ρυθμίσεις για τον πράσινο σηματοδότη.

* **Green Options Widget**:
  + **Name**: Green
  + **Condition**: Εάν το Field 3 είναι ίσο με 3, τότε το πράσινο φως ανάβει.
  + **Update Interval**: Το widget ενημερώνεται κάθε **30 δευτερόλεπτα**.
  + **Χρώμα**: Το χρώμα που εμφανίζεται είναι πράσινο.
* **Field Charts**:

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).



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

# 5. Προγραμματιστικός Κώδικας tiger

## 5.1 Ανάλυση του κώδικα που χρησιμοποιείται

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

**5.3 Παρουσίαση αποτελεσμάτων**

# 6. Συμπεράσματα chat gpt

## 6.1 Ανασκόπηση της εργασίας

## 6.2 Προτάσεις για βελτιώσεις ή μελλοντική επέκταση του έργου

## 



Σας ευχαριστούμε (ΜΠΟΡΕΙ ΚΑΙ ΟΧΙ) για την προσοχή σας.

