

ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ
ΗΛΕΚΤΡΟΝΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

«Μελέτη και εκτίμηση ποσότητας σωματιδίων σε σωλήνα
με χρήση χρωματικού αισθητήρα»



Φοιτητής

ΑΝΑΣΤΑΣΙΟΣ ΣΑΒΒΑΙΔΗΣ 52214m

Επιβλέπων

Δρ. Τσιακμάκης Κυριάκος

Οκτώβριος 2024

Μελέτη και εκτίμηση ποσότητας σωματιδίων σε σωλήνα με χρήση χρωματικού αισθητήρα

Κωδικός: 23301

Φοιτητής: ΑΝΑΣΤΑΣΙΟΣ ΣΑΒΒΑΙΔΗΣ

Εισηγητής: Δρ Κυριάκος Τσιακμάκης

Ημερομηνία ανάληψης Π.Ε. 31-10-2023

Ημερομηνία περάτωσης Π.Ε. 28-09-2024

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 διπλωματική εργασία, στο Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του ΔΙ.ΠΑ.Ε.

*Η παρούσα εργασία αποτελεί πνευματική ιδιοκτησία του φοιτητή **Σαββαΐδη Αναστάσιου** 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 διπλωματικής 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 παρούσα εργασία επικεντρώνεται στη μελέτη και εκτίμηση της ποσότητας σωματιδίων σε σωλήνα, με τη χρήση ενός χρωματικού αισθητήρα. Ο αισθητήρας AS7261 χρησιμοποιείται για τη μη επεμβατική μέτρηση των χαρακτηριστικών των σωματιδίων, όπως το χρώμα, με σκοπό την εκτίμηση της ποσότητάς τους. Αναπτύχθηκε ειδικός κώδικας σε C++ με χρήση `izc` για την επικοινωνία του υπολογιστή με τον αισθητήρα και ένας server για την αποστολή, αποθήκευση και προβολή των δεδομένων. Η ανάλυση των δεδομένων βασίζεται στην απόκριση του αισθητήρα στα φασματικά χαρακτηριστικά των σωματιδίων, με εφαρμογές στη βιομηχανία και στην επιστημονική έρευνα. Το σύστημα που αναπτύσσεται επιδιώκει να προσφέρει μια γρήγορη, και οικονομικά αποδοτική λύση για τη μέτρηση σωματιδίων, μειώνοντας τα σφάλματα και βελτιώνοντας τη διαχείριση των πόρων με χρήση κώδικα.

« Study and Estimation of Particle Quantity in a Tube Using a Color Sensor »

Abstract

This study focuses on the investigation and estimation of the quantity of particles in a tube using a color sensor. The AS7261 sensor is used for non-invasive measurement of particle characteristics, such as color, to estimate their quantity. A specialized C++ code was developed using **i2c** for communication between the computer and the sensor, along with a server for the transmission, storage, and visualization of the data. Data analysis is based on the sensor's response to the spectral properties of the particles, with applications in industry and scientific research. The developed system aims to provide a fast and cost-effective solution for particle measurement, reducing errors and improving resource management through the use of code.

Ευχαριστίες

Θα ήθελα να εκφράσω την ευγνωμοσύνη μου στους γονείς μου για τη στήριξή τους, καθώς και στον επιβλέποντα κ. για την αδιάκοπη καθοδήγηση, τις επιστημονικές συμβουλές του, και τη συμβολή του στην ανάπτυξη του κώδικα της εφαρμογής.

Περιεχόμενα

Περίληψη	iv
Abstract	v
Ευχαριστίες	vi
Περιεχόμενα.....	vii
Κατάλογος Σχημάτων	ix
Κεφάλαιο 1ο: Εισαγωγή.....	10
1.1 Εισαγωγή.....	10
1.2 Συνεισφορά της εργασίας.....	11
1.3 Δομή της εργασίας	11
Κεφάλαιο 2ο: Βιβλιογραφική Ανασκόπηση	12
2.1 Εισαγωγή.....	12
Κεφάλαιο 3ο: Γλώσσες προγραμματισμού και το kit αξιολόγησης	16
3.1 Python	16
3.1.1 Εισαγωγή στην Python για Ανάπτυξη Ιστού.....	18
3.2 C++.....	21
3.3 ASM AS7261	24
3.3.1 Αρχιτεκτονική και Λειτουργίες του AS7261	27
3.4 AS726x-iSPI.Evaluation.Kit	31
3.4.1 Η πειραματική διάταξη	34
Κεφάλαιο 4ο: Το σύστημα μέτρησης.....	35
4.1 Εισαγωγή.....	35
4.1.1 Η εφαρμογή.....	35
4.1.2 Η εργασία	36
4.1.3 Ανάλυση Δεδομένων.....	37
4.1.4 Ανάλυση μέσω κώδικα.....	40
4.2 Μετρήσεις	60
4.2.1 Υλικά και Μέθοδοι	61
4.2.2 Διαδικασία Πειραμάτων.....	61
4.3 Το IoT Portal.....	63

4.4	Η Βάση του συστήματος	75
Κεφάλαιο 5ο:	Συμπεράσματα και προτάσεις βελτίωσης.....	77
ΒΙΒΛΙΟΓΡΑΦΙΑ.....		78
ΠΑΡΑΡΤΗΜΑ Α		79

Κατάλογος Σχημάτων

Εικόνα 3.1: Πειραματική Διάταξη με κυλινδρικό σωληνάριο με καθαρό νερό	34
Εικόνα 3.2: Πειραματική Διάταξη με κυλινδρικό σωληνάριο με πράσινο χρώμα νερό	34
Εικόνα 4.1: Διάγραμμα ροής της λειτουργίας μέτρησης μέσω i2c	40
Εικόνα 4.2: Μέτρηση μαύρου.....	65
Εικόνα 4.3: Μέτρηση άσπρου.....	66
Εικόνα 4.4: Μέτρηση - Μόνο νερό	67
Εικόνα 4.5: Μέτρηση - Πράσινο δείγμα - Χωρίς σωματίδια.....	68
Εικόνα 4.6: Μέτρηση - Πράσινο δείγμα - Με σωματίδια όγκου 0.5cm^3	69
Εικόνα 4.7: Μέτρηση - Πράσινο δείγμα - Με σωματίδια όγκου 0.75cm^3	70
Εικόνα 4.8: Μέτρηση - Πράσινο δείγμα - Με σωματίδια όγκου 1cm^3	71
Εικόνα 4.9: Διάγραμμα μέσης τιμής NIR - Δείγμα με Trend Line	74

Κεφάλαιο 1ο: Εισαγωγή

1.1 Εισαγωγή

Η παρούσα εργασία εξετάζει τη μελέτη και την εκτίμηση της ποσότητας ενός υγρού σε σωλήνα με τη χρήση ενός χρωματικού αισθητήρα, στοχεύοντας στη δημιουργία ενός αποδοτικού συστήματος μέτρησης βασισμένου στην ανάλυση χρωματικών δεδομένων. Η εκτίμηση της ποσότητας ενός υγρού μέσα σε σωλήνα μπορεί να είναι κρίσιμη για πολλές εφαρμογές, τόσο στην επιστημονική έρευνα όσο και στη βιομηχανία. Τέτοιες εφαρμογές μπορεί να περιλαμβάνουν τον έλεγχο της ποιότητας, τη βιομηχανική παραγωγή, την ανάλυση δειγμάτων σε εργαστήρια, καθώς και τη μελέτη βιολογικών ή χημικών παραμέτρων. Η ανάγκη για ακριβείς και αξιόπιστες μετρήσεις καθιστά τη χρήση σύγχρονων αισθητήρων αναγκαία, ειδικά όταν απαιτείται η μη καταστροφική ανάλυση των δειγμάτων.

Η μελέτη αυτή εστιάζει στη χρήση του αισθητήρα AS7261, ο οποίος είναι ένας πολυφασματικός χρωματικός αισθητήρας σχεδιασμένος για ακριβή μέτρηση χρωματικών συντεταγμένων XYZ, σύμφωνα με το πρότυπο CIE 1931. Αυτός ο αισθητήρας προσφέρει τη δυνατότητα για ανάλυση του χρώματος, παρέχοντας λεπτομερείς πληροφορίες σχετικά με τη φασματική απόκριση του υγρού που βρίσκεται στον σωλήνα. Με βάση αυτές τις μετρήσεις, μπορεί να εκτιμηθεί η ποσότητα του υγρού, καθώς και άλλες παράμετροι όπως η διαύγεια και η σύσταση, οι οποίες σχετίζονται άμεσα με το χρώμα.

Η ανάπτυξη αυτής της τεχνολογίας είναι ιδιαίτερα σημαντική για τη βελτίωση της ακρίβειας των μετρήσεων και την εξοικονόμηση πόρων. Παραδοσιακές μέθοδοι εκτίμησης της ποσότητας και ανάλυσης των χαρακτηριστικών ενός υγρού συχνά απαιτούν επαφή με το δείγμα ή καταστροφικές μεθόδους, οι οποίες μπορεί να επηρεάσουν την ποιότητα των δειγμάτων ή να οδηγήσουν σε σπατάλη πόρων. Η παρούσα εργασία προσφέρει μια λύση σε αυτό το πρόβλημα μέσω της μη επεμβατικής ανάλυσης με χρήση ενός οπτικού αισθητήρα.

Η έρευνα αυτή ενσωματώνει τις δυνατότητες του αισθητήρα AS7261 σε υπάρχουσες μεθοδολογίες μέτρησης και ανάλυσης, συνδυάζοντας τις σύγχρονες τεχνολογίες με παραδοσιακές τεχνικές για την επίτευξη αξιόπιστων αποτελεσμάτων. Επίσης, εξετάζεται η δυνατότητα ενσωμάτωσης αυτών των μεθόδων σε αυτοματοποιημένα συστήματα, όπου οι μετρήσεις μπορούν να συλλέγονται και να αναλύονται σε πραγματικό χρόνο, προσφέροντας έτσι ευελιξία και αποτελεσματικότητα στη διαδικασία.

Μέσω της εργασίας αυτής, προτείνεται ένα νέο σύστημα μέτρησης το οποίο αξιοποιεί την τεχνολογία φασματικής ανάλυσης του αισθητήρα AS7261 για την εκτίμηση της ποσότητας και της σύστασης υγρών, με εφαρμογές σε ποικίλους τομείς, από τη βιομηχανική παραγωγή έως την επιστημονική έρευνα. Με την ακριβή ανάλυση των χρωματικών δεδομένων, το σύστημα που αναπτύσσεται μπορεί

να προσφέρει βελτιωμένες δυνατότητες ελέγχου ποιότητας και παρακολούθησης της κατάστασης των υλικών, με ελάχιστο ανθρώπινο παρέμβαση και εξασφαλίζοντας υψηλή ακρίβεια.

1.2 Συνεισφορά της εργασίας

Στην εργασία παρουσιάζεται μια νέα προσέγγιση στη μέτρηση και εκτίμηση της ποσότητας του περιεχομένου σωλήνα, με τη χρήση του αισθητήρα AS7261. Ο αισθητήρας αυτός είναι σχεδιασμένος για την ακριβή μέτρηση των χρωματικών συντεταγμένων XYZ, σύμφωνα με το πρότυπο CIE 1931, και παρέχει δυνατότητες για ανάλυση του περιεχομένου του σωλήνα βάσει των μετρήσεων που λαμβάνονται. Η εργασία συμβάλλει επίσης στη μελέτη των διαφορετικών παραγόντων που επηρεάζουν τις μετρήσεις, καθώς και στην ανάπτυξη μιας μεθοδολογίας επεξεργασίας των δεδομένων για την εκτίμηση της ποσότητας.

1.3 Δομή της εργασίας

Η εργασία δομείται ως εξής:

- Στο πρώτο κεφάλαιο παρουσιάζεται μια εισαγωγή στο θέμα και στους στόχους της μελέτης.
- Στο δεύτερο κεφάλαιο ακολουθεί βιβλιογραφική ανασκόπηση παρόμοιων τεχνολογιών και μεθόδων μέτρησης που έχουν χρησιμοποιηθεί σε αντίστοιχες εφαρμογές.
- Στο τρίτο κεφάλαιο αναλύονται οι γλώσσες προγραμματισμού και το κιτ αξιολόγησης που χρησιμοποιήθηκαν για την υλοποίηση της μεθοδολογίας.
- Στο τέταρτο κεφάλαιο περιγράφεται το σύστημα μέτρησης και παρουσιάζονται οι τεχνικές και τα εργαλεία που χρησιμοποιήθηκαν για τη συλλογή και την ανάλυση των δεδομένων.
- Τέλος, στο πέμπτο κεφάλαιο παρουσιάζονται τα συμπεράσματα και προτάσεις για μελλοντική βελτίωση της προσέγγισης.

Κεφάλαιο 2ο: Βιβλιογραφική Ανασκόπηση

2.1 Εισαγωγή

Το άρθρο "TinyML-Sensor for Shelf Life Estimation of Fresh Date Fruits" επικεντρώνεται στην ανάπτυξη και χρήση ενός TinyML αισθητήρα για την εκτίμηση της διάρκειας ζωής φρέσκων φρούτων, συγκεκριμένα των φρέσκων χουρμάδων. Οι ερευνητές χρησιμοποίησαν πολυφασματικούς αισθητήρες χαμηλού κόστους (AS7265x), οι οποίοι ενσωματώθηκαν σε ένα μοντέλο μηχανικής μάθησης (ML) που προορίζεται για μικροελεγκτές (TinyML). Το βασικό κίνητρο της έρευνας ήταν να επιτευχθεί μια γρήγορη, ακριβής και οικονομικά προσιτή μέθοδος εκτίμησης της διάρκειας ζωής των χουρμάδων χωρίς να καταστρέφονται οι φρούτοι.

Οι φρέσκοι χουρμάδες έχουν περιορισμένη διάρκεια ζωής και είναι επιρρεπείς σε αλλοίωση, γεγονός που μπορεί να οδηγήσει σε οικονομικές απώλειες για τους παραγωγούς και τους προμηθευτές. Για να επιτευχθεί η εκτίμηση της διάρκειας ζωής, οι ερευνητές χρησιμοποίησαν τους φυσικοχημικούς δείκτες των χουρμάδων, όπως η υγρασία, η περιεκτικότητα σε διαλυτά στερεά, το pH, η σκληρότητα, καθώς και την ανάλυση των αλλαγών στο χρώμα μέσω των πολυφασματικών αισθητήρων.

Η μελέτη περιέγραψε τη χρήση διαφορετικών τύπων συσκευασίας, όπως συσκευασία σε κενό αέρος και τροποποιημένη ατμόσφαιρα, για να βελτιώσουν την ποιότητα και να παρατείνουν τη διάρκεια ζωής των φρούτων. Τα αποτελέσματα έδειξαν ότι η συσκευασία σε κενό αέρος και η χρήση τροποποιημένης ατμόσφαιρας με CO₂ βελτίωσαν τη διάρκεια ζωής των φρούτων, ιδιαίτερα σε χαμηλές θερμοκρασίες (5 °C).

Το μοντέλο TinyML που αναπτύχθηκε χρησιμοποιήθηκε για την πρόβλεψη της διάρκειας ζωής των χουρμάδων σε πραγματικό χρόνο, υποστηρίζοντας τη διαχείριση της αλυσίδας εφοδιασμού και μειώνοντας την απώλεια τροφίμων. Οι ερευνητές τόνισαν επίσης τα οφέλη της χρήσης μικροελεγκτών χαμηλού κόστους για την εφαρμογή μοντέλων μηχανικής μάθησης στις άκρες του δικτύου (edge computing), καθιστώντας την τεχνολογία προσιτή και αποδοτική για αγροτικές εφαρμογές.

Γενικά, η έρευνα προτείνει μια ολοκληρωμένη προσέγγιση που συνδυάζει τεχνικές ανάλυσης, μηχανική μάθηση και διαφορετικές μεθόδους συσκευασίας, προσφέροντας νέες πληροφορίες για την εκτίμηση και διατήρηση της διάρκειας ζωής των φρέσκων χουρμάδων. [1]

Η έρευνα του Talon et al. εξετάζει την επίδραση της φωτεινότητας και της συσχετιζόμενης θερμοκρασίας χρώματος (CCT) στην ορατότητα των οδηγών στις εσωτερικές ζώνες των τούνελ. Οι ερευνητές χρησιμοποιούν LED με διαφορετικά CCT (2000K, 3000K, 5000K) για να δουν πώς επηρεάζεται η ορατότητα των οδηγών υπό διαφορετικές συνθήκες φωτισμού. Δοκιμάστηκαν έξι διαφορετικές κατανομές φάσματος (SPDs) σε ένα ψυχοοπτικό πείραμα, χρησιμοποιώντας ένα μοντέλο τούνελ κλίμακας 1:20.

Στην έρευνα συμμετείχαν 48 παρατηρητές, που κλήθηκαν να αξιολογήσουν την ορατότητα ενός στόχου χρησιμοποιώντας κλίμακα πέντε επιπέδων ορατότητας (VT1-VT5), με VT4 να θεωρείται το ελάχιστο για ασφαλή οδήγηση και VT5 να είναι το μέγιστο. Τα αποτελέσματα δείχνουν ότι τα χαμηλότερα CCT συνδέονται με καλύτερη ορατότητα σε χαμηλές φωτεινότητες, κυρίως στα επίπεδα VT1-VT3, κάτι που αποδίδεται στη λειτουργία των μπλε κόνων και στο μεσοπικό μοντέλο της CIE.

Η αύξηση του περιεχομένου μελανοψίνης, που αποτελεί στοιχείο των ipRGCs (ενδογενώς φωτοευαίσθητων γαγγλιακών κυττάρων του αμφιβληστροειδούς), δεν είχε σημαντική επίδραση στην ορατότητα, καθώς οι μεσοπικές συνθήκες φωτισμού στα τούνελ δεν επιτρέπουν την υπέρβαση του ορίου διέγερσης της μελανοψίνης.

Η συγκεκριμένη εργασία, "TinyML-Sensor for Shelf Life Estimation of Fresh Date Fruits", χρησιμοποιεί τον αισθητήρα **AS7265x**, ο οποίος αποτελείται από τρεις διαφορετικούς οπτικούς αισθητήρες (AS72651, AS72652 και AS72653). Ο αισθητήρας αυτός χρησιμοποιήθηκε για την αξιολόγηση των φυσικοχημικών ιδιοτήτων φρέσκων χουρμάδων σε διαφορετικά στάδια ωρίμανσης. Η μελέτη επικεντρώθηκε σε χαρακτηριστικά όπως το pH, η υγρασία (MC), τα ολικά διαλυτά στερεά (TSS), τα επίπεδα σακχάρων και οι τανίνες. Αυτό επιτεύχθηκε μέσω μετρήσεων της φασματικής ανακλαστικότητας σε διαφορετικά μήκη κύματος, από 410 nm έως 940 nm. Οι φασματικές μπάντες χρησιμοποιήθηκαν για την πρόβλεψη της διάρκειας ζωής των χουρμάδων, ενώ τα δεδομένα του αισθητήρα χρησιμοποιήθηκαν σε μοντέλα μηχανικής μάθησης (machine learning) για ανάλυση σε πραγματικό χρόνο χρησιμοποιώντας τεχνικές TinyML σε πλατφόρμα Arduino.

Σε άλλο τμήμα της έρευνας, ο αισθητήρας AS7265x χρησιμοποιήθηκε για τον υπολογισμό του λόγου ανακλαστικότητας, συγκρίνοντας τις τιμές εισερχόμενου και ανακλώμενου φωτός για τον καθορισμό χαρακτηριστικών των φρούτων, όπως τα επίπεδα σακχάρων, οι τανίνες και η υγρασία. Τα χαρακτηριστικά της ανακλαστικότητας χρησιμοποιήθηκαν στη συνέχεια για την πρόβλεψη της διάρκειας ζωής των φρούτων μέσω νευρωνικών δικτύων βασισμένων σε TinyML. [2]

Συμπεράσματα:

1. Οι πηγές φωτός με χαμηλότερη θερμοκρασία χρώματος (2000K) παρέχουν καλύτερη ορατότητα στις χαμηλές φωτεινότητες, καθιστώντας τις πιο κατάλληλες για τη βελτίωση της ορατότητας των οδηγών σε συνθήκες χαμηλού φωτισμού, όπως αυτές που επικρατούν στις εσωτερικές ζώνες των τούνελ.
2. Η αύξηση της διέγερσης των ipRGCs δεν φάνηκε να επηρεάζει την ορατότητα στις μεσοπικές συνθήκες, γεγονός που σημαίνει ότι η χρήση πηγών φωτός με υψηλότερο περιεχόμενο μελανοψίνης δεν βελτιώνει απαραίτητα την ορατότητα των οδηγών.
3. Η μελέτη συνιστά την επανεξέταση των υφιστάμενων προτύπων φωτισμού για τα τούνελ, καθώς οι συστάσεις της CIE και CEN ενδέχεται να μην ανταποκρίνονται πλήρως στις

πραγματικές ανάγκες ορατότητας των οδηγών. Προτείνεται ένα εύρος φωτεινότητας μεταξύ 2.6 και 7.7 cd/m², το οποίο είναι διαφορετικό από τις τρέχουσες συστάσεις.

Η εργασία "Development of a Fibre Optic-Based Patch Sensor for Monitoring Structures", περιγράφει την ανάπτυξη και χρήση ενός αισθητήρα με βάση τις οπτικές ίνες για την παρακολούθηση των δομών σε έργα πολιτικού μηχανικού. Στόχος της μελέτης είναι να βελτιώσει τις μεθόδους παρακολούθησης της υγείας των κατασκευών (Structural Health Monitoring - SHM), χρησιμοποιώντας νέες τεχνολογίες που επιτρέπουν τη συνεχή ή απομακρυσμένη παρακολούθηση.

Ο αισθητήρας που χρησιμοποιείται είναι ένας καινοτόμος οπτικός αισθητήρας τύπου 2D, ο οποίος είναι εξοπλισμένος με υφασμένες οπτικές ίνες και έχει τη δυνατότητα να ενσωματώνεται σε σκυρόδεμα ή να επικολλάται στην επιφάνειά του. Ο ανιχνευτής που χρησιμοποιήθηκε αναπτύχθηκε από την εταιρεία Inouid και χρησιμοποιεί τον αισθητήρα AS7261 της ams OSRAM, ο οποίος είναι ένας χρωματικός αισθητήρας λευκού χρώματος που παρέχει συντεταγμένες χρώματος XYZ καθώς και ανάλυση στο φάσμα του εγγύς υπερύθρου (NIR).

Η τεχνολογία αυτή δοκιμάστηκε σε δείγματα σκυροδέματος στο εργαστήριο, χρησιμοποιώντας μεθόδους όπως η ψηφιακή συσχέτιση εικόνας 2D (DIC) για την ανίχνευση ρωγμών και τη μέτρηση του πλάτους τους. Ο αισθητήρας έδειξε ότι μπορεί να ανιχνεύσει το άνοιγμα ρωγμών της τάξης των 0.04 mm. Επιπλέον, ο αισθητήρας ήταν ευαίσθητος στην παρουσία νερού και στις μεταβολές θερμοκρασίας, καταγράφοντας μεταβολές στην ένταση του φωτός ανάλογα με τη θερμοκρασία ή την παρουσία νερού.

Η μελέτη παρουσίασε επίσης την εφαρμογή του αισθητήρα σε μεγαλύτερη κλίμακα, όπως στις πραγματικές κατασκευές, και τους περιορισμούς που θα μπορούσαν να αντιμετωπιστούν κατά την εφαρμογή της τεχνολογίας σε πραγματικά έργα πολιτικού μηχανικού. Οι προοπτικές της τεχνολογίας αυτής είναι σημαντικές, καθώς προσφέρουν μια αξιόπιστη και οικονομικά αποδοτική μέθοδο για τη διαρκή παρακολούθηση της κατάστασης των δομών, συμβάλλοντας στην ασφάλεια και στη μείωση του κόστους συντήρησης. [3]

Η εργασία "Use of the Chromation Spectrometer for White Point Balancing in Mobile Devices", περιγράφει τη χρήση ενός φασματογράφου της Chromation για την εξισορρόπηση του λευκού σημείου σε κινητές συσκευές. Ο κύριος στόχος της μελέτης είναι η αναπροσαρμογή του λευκού σημείου στις οθόνες των συσκευών ώστε να ταιριάζει με το λευκό σημείο του περιβάλλοντος φωτισμού, κάτι που βελτιώνει την αντιληπτή ποιότητα χρωμάτων και την απόδοση της οθόνης.

Συγκεκριμένα, ο αισθητήρας AS7261 από την AMS αναφέρθηκε ως σχετικός με την τεχνολογία True Tone της Apple, η οποία χρησιμοποιείται για την προσαρμογή των χρωμάτων στις οθόνες των προϊόντων της Apple (π.χ. iPhone, iPad). Ο αισθητήρας αυτός μετρά τέσσερα κανάλια,

συμπεριλαμβανομένων τριών CIE-matched καναλιών και ενός καναλιού υπερύθρου, το οποίο μπορεί επίσης να χρησιμοποιηθεί ως αισθητήρας εγγύτητας.

Η εργασία αναφέρεται σε εφαρμογές όπου ο αισθητήρας χρησιμοποιείται για τη ρύθμιση του λευκού σημείου στις φωτογραφίες, ενώ επίσης επιτρέπει τη χρήση του αισθητήρα για μέτρηση του απόλυτου χρώματος, την εξισορρόπηση των φωτογραφιών, και για πλήρη φασματική ανάλυση, όπως στην οξυμετρία παλμού. Επίσης, αναφέρει πώς ο αισθητήρας μπορεί να χρησιμοποιηθεί για αναλύσεις φωτισμού του περιβάλλοντος, όπως για τον υπολογισμό της ακτινοβολίας UV και την ανάλυση της έκθεσης σε μικρού μήκους κύματος φως, που μπορεί να επηρεάσει τους κερκάρδιους ρυθμούς.

Επιπλέον, η εργασία σημειώνει τη σημαντική εμπορική ευκαιρία για την ενσωμάτωση της τεχνολογίας αυτής στις κινητές συσκευές, με την πιθανότητα να χρησιμοποιούνται δύο φασματογράφοι σε κάθε συσκευή, ένας για το μπροστινό μέρος και ένας για το πίσω μέρος. [4]

Η εργασία " Nondestructive Evaluation of Banana Maturity Using NIR AS7263 Sensor" ασχολείται με την μη καταστροφική εκτίμηση της ωρίμανσης των μπανανών με τη χρήση του αισθητήρα AS7263. Ο αισθητήρας αυτός χρησιμοποιεί τεχνολογία φασματοσκοπίας στο εγγύς υπέρυθρο (NIR) για να αξιολογήσει τη φυσικοχημική ποιότητα των μπανανών, επικεντρώνοντας στο περιεχόμενο σακχάρων (Brix), το οποίο μεταβάλλεται κατά τη διάρκεια της ωρίμανσης των φρούτων.

Στην παρούσα έρευνα, χρησιμοποιήθηκε το φάσμα του αισθητήρα AS7263 (610 nm – 860 nm) για να μετρηθεί η φασματική αντανάκλαση των μπανανών σε διάφορα στάδια ωρίμανσης. Η ανάλυση των δεδομένων έγινε με τη χρήση πολλαπλής γραμμικής παλινδρόμησης (MLR) και τεχνητού νευρωνικού δικτύου (ANN). Το μοντέλο MLR ανέπτυξε μια ισχυρή συσχέτιση μεταξύ των τιμών Brix και του φάσματος NIR, με τιμή $R^2 = 0.9507$, ενώ το ANN κατέδειξε βελτιωμένη απόδοση με τιμή $R^2 = 0.84$ για την πρόβλεψη της ωρίμανσης.

Η ωρίμανση των μπανανών ταξινομήθηκε σε τέσσερις κατηγορίες: άγουρη, λιγότερο ώριμη, ώριμη, και υπερώριμη, με βάση το περιεχόμενο Brix. Τα αποτελέσματα δείχνουν ότι ο αισθητήρας AS7263 είναι κατάλληλος για την μη καταστροφική εκτίμηση της ωρίμανσης, ενώ οι αλλαγές στη χλωροφύλλη και το Brix αποτελούν αξιόπιστους δείκτες για την παρακολούθηση της ποιότητας των μπανανών. [5]

Κεφάλαιο 3ο: Γλώσσες προγραμματισμού και το kit αξιολόγησης

3.1 Python

Η Python είναι μια δημοφιλής γλώσσα προγραμματισμού που σχεδιάστηκε από τον Guido van Rossum το 1991. Από την κυκλοφορία της μέχρι σήμερα, έχει αποκτήσει τεράστια δημοτικότητα λόγω της απλότητας και ευελιξίας της. Η Python χαρακτηρίζεται από την υψηλού επιπέδου σύνταξή της, που επιτρέπει στους προγραμματιστές να αναπτύσσουν εφαρμογές και να επιλύουν σύνθετα προβλήματα με λιγότερο κόπο. [6-10]

Η Python μπορεί να χρησιμοποιηθεί για ένα ευρύ φάσμα εφαρμογών, όπως η ανάλυση δεδομένων, η ανάπτυξη ιστοσελίδων, η τεχνητή νοημοσύνη και η μηχανική μάθηση, η ανάπτυξη παιχνιδιών, και πολλές άλλες χρήσεις. Η ευελιξία αυτή πηγάζει από την ύπαρξη πλούσιων βιβλιοθηκών και την υποστήριξη πολλών παραδειγμάτων προγραμματισμού, όπως διαδικαστικό, αντικειμενοστραφή, και λειτουργικό προγραμματισμό.

Οι μεταβλητές στην Python είναι "δοχεία" που χρησιμοποιούνται για την αποθήκευση δεδομένων. Υπάρχουν διάφοροι τύποι δεδομένων που υποστηρίζονται, όπως ακέραιοι αριθμοί, δεκαδικοί, αλφαριθμητικές αλυσίδες, και λογικές τιμές. Ο τύπος δεδομένων καθορίζεται αυτόματα κατά την εκχώρηση τιμής σε μία μεταβλητή, και αυτό κάνει τη Python μια γλώσσα με δυναμική τυποποίηση.

Η Python παρέχει διάφορους τύπους λειτουργιών για την πραγματοποίηση υπολογισμών και συγκρίσεων. Αυτές περιλαμβάνουν αριθμητικές λειτουργίες για βασικές αριθμητικές πράξεις, συγκριτικές λειτουργίες για την αξιολόγηση συνθηκών, και λογικές λειτουργίες που επιτρέπουν τη σύνθεση πολλαπλών συνθηκών. Όλα αυτά τα εργαλεία είναι σημαντικά για τη δημιουργία σύνθετων προγραμμάτων και για τη λήψη αποφάσεων στο πλαίσιο του προγραμματισμού.

Βασικές Δομές Δεδομένων

Λίστες

Οι λίστες είναι ευέλικτες συλλογές δεδομένων που επιτρέπουν την αποθήκευση πολλαπλών στοιχείων μέσα σε μια ενιαία μεταβλητή. Οι λίστες μπορούν να περιέχουν αντικείμενα διαφορετικών τύπων και είναι δυναμικά επεκτάσιμες, επιτρέποντας την προσθήκη και αφαίρεση στοιχείων κατά τη διάρκεια της εκτέλεσης.

Πλειάδες

Οι πλειάδες είναι παρόμοιες με τις λίστες, αλλά έχουν την ιδιότητα ότι είναι αμετάβλητες. Αυτό σημαίνει ότι τα στοιχεία που περιέχονται σε μια πλειάδα δεν μπορούν να τροποποιηθούν μετά τη δημιουργία της. Αυτό καθιστά τις πλειάδες χρήσιμες για την αποθήκευση δεδομένων που δεν πρέπει να αλλάξουν κατά τη διάρκεια της εκτέλεσης.

Λεξικά

Τα λεξικά είναι συλλογές δεδομένων που επιτρέπουν την αποθήκευση δεδομένων με τη μορφή ζευγών κλειδιού-τιμής. Το κάθε κλειδί πρέπει να είναι μοναδικό και χρησιμοποιείται για να εντοπιστεί η αντίστοιχη τιμή. Τα λεξικά είναι πολύ χρήσιμα όταν χρειάζεται να αποθηκευτούν δεδομένα με σαφείς σχέσεις, όπως π.χ. τα χαρακτηριστικά ενός αντικειμένου.

Σύνολα

Τα σύνολα είναι συλλογές που περιέχουν μοναδικά αντικείμενα και δεν διατηρούν συγκεκριμένη σειρά. Είναι χρήσιμα για την αποθήκευση δεδομένων όπου δεν χρειάζεται να επαναληφθούν τα στοιχεία, όπως για παράδειγμα ένα σύνολο μοναδικών αριθμών.

Χειρισμός Αρχείων

Η Python παρέχει εργαλεία για την ανάγνωση και εγγραφή αρχείων. Αυτό είναι ιδιαίτερα χρήσιμο για την αποθήκευση δεδομένων σε εξωτερικά αρχεία, για παράδειγμα, τη δημιουργία αναφορών ή την ανάκτηση πληροφοριών από αρχείο κειμένου. Τα αρχεία μπορούν να ανοιχτούν είτε για ανάγνωση είτε για εγγραφή, και οι λειτουργίες αυτές είναι χρήσιμες σε πολλές εφαρμογές.

Εξωτερικές Βιβλιοθήκες και Εργαλεία

Η Python διαθέτει μια μεγάλη ποικιλία από εξωτερικές βιβλιοθήκες που καλύπτουν ευρύ φάσμα εφαρμογών, όπως:

NumPy και Pandas για την ανάλυση δεδομένων.

Matplotlib και Seaborn για την οπτικοποίηση δεδομένων.

Flask και Django για την ανάπτυξη ιστοσελίδων.

TensorFlow και PyTorch για την τεχνητή νοημοσύνη και τη μηχανική μάθηση.

Εφαρμογές της Python

Η Python χρησιμοποιείται ευρέως σε πολλούς τομείς:

Ανάλυση δεδομένων και επιστήμη δεδομένων: Χάρη στην απλότητα και τη δύναμη των βιβλιοθηκών όπως οι NumPy και Pandas, η Python είναι η γλώσσα προτίμησης για την ανάλυση δεδομένων.

Ανάπτυξη ιστοσελίδων: Πλαίσια όπως το Flask και το Django επιτρέπουν την ταχεία ανάπτυξη διαδικτυακών εφαρμογών.

Τεχνητή νοημοσύνη και μηχανική μάθηση: Βιβλιοθήκες όπως οι TensorFlow και PyTorch καθιστούν την Python ιδανική για την ανάπτυξη συστημάτων τεχνητής νοημοσύνης.

Αυτοματοποίηση και scripting: Η Python χρησιμοποιείται για την αυτοματοποίηση καθημερινών εργασιών και για τη δημιουργία σεναρίων που κάνουν πιο εύκολη τη διαχείριση συστημάτων.

3.1.1 Εισαγωγή στην Python για Ανάπτυξη Ιστού

Η Python είναι μια ισχυρή γλώσσα προγραμματισμού που είναι γνωστή για την ευκολία χρήσης και την ευελιξία της. Ειδικά για την ανάπτυξη εφαρμογών ιστού, η Python έχει αποδειχθεί ιδιαίτερα αποτελεσματική, χάρη στην ύπαρξη διαφόρων ισχυρών frameworks, όπως το **Flask** και το **Django**. Το Flask, συγκεκριμένα, είναι ένα μικρό και ελαφρύ framework που επιτρέπει τη γρήγορη ανάπτυξη ιστού, δίνοντας στους προγραμματιστές την ελευθερία να σχεδιάζουν τις εφαρμογές τους με το δικό τους τρόπο. [11]

Flask: Ένα Ελαφρύ Framework για Ιστό

Το Flask είναι ένα micro-framework για την ανάπτυξη εφαρμογών ιστού με τη Python. Το “micro” δεν σημαίνει ότι το Flask είναι περιορισμένο, αλλά ότι παρέχει μόνο τα βασικά εργαλεία και επιτρέπει στον προγραμματιστή να προσθέσει ό,τι άλλο χρειάζεται. Αυτό κάνει το Flask ιδανικό για μικρές και μεσαίου μεγέθους εφαρμογές που χρειάζονται ευελιξία και ταχύτητα.

Χαρακτηριστικά του Flask

- **Ελαφριά Φύση:** Το Flask παρέχει έναν απλό τρόπο για την ανάπτυξη εφαρμογών ιστού, χωρίς περιττά χαρακτηριστικά.

- **Επεκτασιμότητα:** Παρά την ελαφριά του φύση, το Flask είναι πολύ επεκτάσιμο και μπορεί να συνδεθεί με βιβλιοθήκες για βάσεις δεδομένων, διαχείριση χρήστη, και άλλα.
- **Jinja2 Templating:** Χρησιμοποιεί το **Jinja2** ως τη μηχανή δημιουργίας templates, επιτρέποντας την εύκολη δημιουργία HTML σελίδων με δυναμικά περιεχόμενα.
- **Werkzeug:** Το Flask βασίζεται στο **Werkzeug** για τον χειρισμό των αιτημάτων HTTP, προσφέροντας έναν σταθερό μηχανισμό για τη διαχείριση των αιτήσεων και αποκρίσεων.

Η Φιλοσοφία του Flask

Το Flask έχει σχεδιαστεί γύρω από την ιδέα του **WSGI** (Web Server Gateway Interface) και του **routing**. Αυτό σημαίνει ότι κάθε URL της εφαρμογής σας είναι συνδεδεμένο με μία συγκεκριμένη συνάρτηση στην Python. Αυτές οι συναρτήσεις είναι γνωστές ως *view functions* και είναι υπεύθυνες για την επιστροφή των αποκρίσεων στους χρήστες.

3.1.1.1 Θεμελιώδεις Αρχές του Flask

Διαμόρφωση (Routing)

Το Flask χρησιμοποιεί τη διακόσμηση για τον καθορισμό των URL και τον ορισμό του πώς θα απαντήσουν οι εφαρμογές σε αιτήσεις. Το **routing** επιτρέπει τη σύνδεση συγκεκριμένων διευθύνσεων με συγκεκριμένες λειτουργίες, καθορίζοντας έτσι την αρχιτεκτονική της εφαρμογής σας.

Συναρτήσεις Προβολής (View Functions)

Οι συναρτήσεις προβολής είναι αυτές που διαχειρίζονται τα αιτήματα και επιστρέφουν αποκρίσεις. Όταν ένας χρήστης κάνει μια αίτηση, το Flask καλεί την αντίστοιχη συνάρτηση προβολής, η οποία μπορεί να επιστρέψει δεδομένα, HTML σελίδες, ή ακόμα και να κατευθύνει τον χρήστη σε άλλη σελίδα.

Πρότυπα με Jinja2 (Templates)

Το **Jinja2** είναι η μηχανή δημιουργίας προτύπων που χρησιμοποιείται από το Flask για να δημιουργήσει δυναμικές HTML σελίδες. Επιτρέπει τη χρήση δυναμικού περιεχομένου, όπως η εισαγωγή δεδομένων από τη βάση, και προσφέρει χρήσιμες λειτουργίες όπως η κληρονομικότητα προτύπων και η χρήση φίλτρων.

Φόρμες και Δεδομένα Χρήστη

Οι φόρμες είναι κρίσιμες για τη συλλογή δεδομένων από τους χρήστες. Το Flask παρέχει τη δυνατότητα να χειριστεί τα δεδομένα που αποστέλλονται μέσω φόρμας, και να τα επεξεργαστεί στο backend. Χρησιμοποιώντας εργαλεία όπως το **Flask-WTF**, οι προγραμματιστές μπορούν να κάνουν τη διαχείριση των φορμών πιο απλή και να διασφαλίσουν την εγκυρότητα των δεδομένων.

Εργασία με Βάσεις Δεδομένων

Το Flask είναι ευέλικτο και μπορεί να συνδεθεί με διάφορες βάσεις δεδομένων, όπως **SQLite**, **MySQL**, και **PostgreSQL**. Οι προγραμματιστές μπορούν να χρησιμοποιήσουν βιβλιοθήκες όπως το **SQLAlchemy**, το οποίο είναι ένας ισχυρός Object-Relational Mapper (ORM), που διευκολύνει τη διαχείριση των βάσεων δεδομένων με Pythonic τρόπο.

Το SQLAlchemy επιτρέπει στους προγραμματιστές να εργάζονται με τα δεδομένα τους σε επίπεδο αντικειμένου. Αντί να γράφουν απευθείας SQL ερωτήματα, μπορούν να δημιουργούν μοντέλα που αντιπροσωπεύουν τους πίνακες της βάσης δεδομένων και να αλληλεπιδρούν με αυτά μέσω αντικειμένων Python. Αυτό καθιστά τη διαχείριση των δεδομένων πιο φυσική και λιγότερο επιρρεπή σε λάθη.

Επεκτασιμότητα και Διαχείριση Εφαρμογών

Το Flask προσφέρει μεγάλη ευελιξία όσον αφορά τη δομή των έργων. Για μεγαλύτερες εφαρμογές, είναι σύνηθες να χωρίζεται ο κώδικας σε πολλαπλά αρχεία και modules, ώστε να διατηρείται η οργάνωση και η ευκολία στη συντήρηση του κώδικα.

Ασφάλεια και Διαχείριση Χρήστη

Η ασφάλεια είναι κρίσιμη για κάθε εφαρμογή ιστού. Το Flask παρέχει την υποδομή για τη δημιουργία αυθεντικοποίησης και εξουσιοδότησης, ενώ μπορεί να επεκταθεί με βιβλιοθήκες όπως το **Flask-Login** και το **Flask-Security** για την εύκολη διαχείριση χρηστών και ρόλων.

Το Flask υποστηρίζει μηχανισμούς για την προστασία από επιθέσεις CSRF (Cross-Site Request Forgery) και XSS (Cross-Site Scripting), μέσω των ενσωματωμένων εργαλείων και των εξωτερικών επεκτάσεων. Αυτό διασφαλίζει ότι τα δεδομένα των χρηστών παραμένουν ασφαλή και προστατευμένα.

Ανάπτυξη και Εγκατάσταση Εφαρμογών Flask

Κατά την ανάπτυξη εφαρμογών Flask, μπορούν να χρησιμοποιηθούν διάφορα εργαλεία για την απλούστευση της διαδικασίας, όπως οι **Flask Debugger** και **Werkzeug Debugger**, που βοηθούν στην ανίχνευση λαθών κατά την ανάπτυξη.

Για την εγκατάσταση μιας εφαρμογής Flask σε περιβάλλον παραγωγής, είναι κοινή πρακτική να χρησιμοποιούνται εργαλεία όπως το **Gunicorn** ή το **uWSGI** σε συνδυασμό με έναν web server όπως ο **Nginx**. Αυτό επιτρέπει την κλιμάκωση και τη διαχείριση πολλαπλών αιτημάτων με υψηλή απόδοση.

Πρακτικές Εφαρμογές του Flask

Το Flask μπορεί να χρησιμοποιηθεί σε πολλές περιπτώσεις και είδη εφαρμογών, όπως:

- **API Servers:** Το Flask μπορεί να χρησιμοποιηθεί για τη δημιουργία RESTful APIs που εξυπηρετούν δεδομένα σε εξωτερικούς πελάτες.
- **Διαδικτυακές Εφαρμογές:** Χρησιμοποιώντας το Flask, οι προγραμματιστές μπορούν να δημιουργήσουν πλήρως λειτουργικές διαδικτυακές εφαρμογές.
- **Γρήγορα Prototypes:** Λόγω της απλότητας του, το Flask είναι ιδανικό για τη γρήγορη ανάπτυξη πρωτοτύπων και πειραματικών έργων.

3.2 C++

Η C++ είναι μια από τις πιο δημοφιλείς και ισχυρές γλώσσες προγραμματισμού, που χρησιμοποιείται ευρέως σε ποικίλες βιομηχανίες και εφαρμογές. Η ισχύς της έγκειται στη συνδυασμένη υποστήριξη της διαδικασιακής, αντικειμενοστραφούς και γενικής προγραμματιστικής παραδείγματος, κάτι που την καθιστά ιδανική για την ανάπτυξη λογισμικού που απαιτεί υψηλή απόδοση και έλεγχο του υλικού. [12]

Ιστορική Αναδρομή και Χαρακτηριστικά

Η C++ αναπτύχθηκε αρχικά από τον Bjarne Stroustrup το 1983, ως μια επέκταση της C που προσέθεσε την υποστήριξη για αντικειμενοστραφή προγραμματισμό. Η C ήταν ήδη γνωστή για την απόδοσή της και την ικανότητα άμεσης διαχείρισης του υλικού (hardware). Η C++ συνδυάζει την ισχύ της C με νέες έννοιες που προσφέρουν καλύτερη αναπαράσταση και διαχείριση πολύπλοκων προβλημάτων.

Τα κυριότερα χαρακτηριστικά της C++ περιλαμβάνουν:

Υψηλή Απόδοση: Η C++ δημιουργεί αποδοτικό εκτελέσιμο κώδικα, ο οποίος είναι σχεδόν ισάξιος με τον κώδικα που θα γράφονταν απευθείας σε γλώσσα μηχανής (Assembly).

Ευελξία: Υποστηρίζει διαδικασιακή, αντικειμενοστραφή και γενική (template) προγραμματιστική παράδειγμα.

Διαχείριση Μνήμης: Παρέχει άμεσο έλεγχο στη διαχείριση μνήμης, επιτρέποντας χειροκίνητη κατανομή και αποδέσμευση.

Επέκταση: Η C++ είναι δυνατή στη δημιουργία βιβλιοθηκών που μπορούν να επαναχρησιμοποιηθούν σε διάφορα έργα.

Γιατί η C++ είναι Κατάλληλη για Υπολογιστικές Εφαρμογές και Ενσωματωμένα Συστήματα

Η C++ είναι εξαιρετικά χρήσιμη για εφαρμογές που απαιτούν γρήγορη εκτέλεση, αποτελεσματική διαχείριση πόρων, και άμεση επικοινωνία με το υλικό.

Σε εφαρμογές όπως η επεξεργασία δεδομένων από αισθητήρες, όπου η απόδοση και η ταχύτητα είναι ζωτικής σημασίας, η C++ επιτρέπει τη βελτιστοποίηση της εκτέλεσης του κώδικα. Η γλώσσα παρέχει τη δυνατότητα διαχείρισης του υλικού σε χαμηλό επίπεδο, όπως η άμεση πρόσβαση στη μνήμη και τους καταχωρητές, κάτι που την καθιστά ιδανική για προγραμματισμό συστημάτων και ενσωματωμένα (embedded) συστήματα.

Η C++ προσφέρει τη δυνατότητα χειροκίνητης διαχείρισης μνήμης μέσω των δεικτών (pointers) και της χρήσης των εντολών new και delete. Η δυνατότητα άμεσου ελέγχου της κατανομής και της αποδέσμευσης μνήμης είναι σημαντική σε περιβάλλοντα όπου οι πόροι είναι περιορισμένοι, όπως τα ενσωματωμένα συστήματα.

Σε εφαρμογές που εμπλέκονται δεδομένα αισθητήρων, όπως ο μετασχηματισμός δεδομένων από το χρωματικό πρότυπο XYZ στο RGB, η C++ επιτρέπει τη χρήση γρήγορων αλγορίθμων που εκτελούνται άμεσα στο υλικό, χωρίς την καθυστέρηση που θα προκαλούνταν από την ανάγκη για garbage collection ή άλλες διαδικασίες που υπάρχουν σε γλώσσες υψηλότερου επιπέδου.

Για παράδειγμα, μια μετατροπή από το πρότυπο XYZ στο RGB όπως αυτή που αναφέρθηκε μπορεί να απαιτεί αυστηρή διαχείριση των αριθμητικών πράξεων για να εξασφαλιστεί η ακρίβεια και η αποτελεσματικότητα του κώδικα, κάτι που η C++ διευκολύνει μέσω της χρήσης των τύπων float ή double και της άμεσης αριθμητικής πράξης.

Η C++ υποστηρίζει τον αντικειμενοστραφή προγραμματισμό, που είναι απαραίτητος για την καλύτερη οργάνωση του κώδικα, ειδικά σε μεγάλα έργα. Η έννοια των κλάσεων και των αντικειμένων επιτρέπει

στους προγραμματιστές να προσομοιώνουν πραγματικά συστήματα και να εργάζονται με αφηρημένες έννοιες με τρόπο που είναι πιο εύκολο να διαχειριστούν και να αναπτύξουν.

Για παράδειγμα, σε ένα σύστημα αισθητήρων, κάθε αισθητήρας μπορεί να αναπαρασταθεί ως μια κλάση με συγκεκριμένες ιδιότητες (π.χ. τύπος αισθητήρα, τιμή) και μεθόδους (π.χ. καταγραφή δεδομένων, μετατροπή). Αυτό επιτρέπει την επαναχρησιμοποίηση και την καλύτερη δομή του κώδικα.

Η C++ παρέχει τη δυνατότητα γενικού προγραμματισμού μέσω templates, κάτι που επιτρέπει την ανάπτυξη κώδικα που είναι επαναχρησιμοποιήσιμος και ανεξάρτητος από τους τύπους δεδομένων. Αυτό είναι ιδιαίτερα χρήσιμο σε εφαρμογές που απαιτούν την επεξεργασία διαφορετικών τύπων δεδομένων με τον ίδιο τρόπο, όπως στην ανάπτυξη αλγορίθμων για την επεξεργασία εικόνας ή τον υπολογισμό μαθηματικών λειτουργιών σε δεδομένα αισθητήρων.

Τα ενσωματωμένα συστήματα (embedded systems) απαιτούν απόδοση σε επίπεδο υλικού και περιορισμένη χρήση μνήμης. Η C++ επιτρέπει την ανάπτυξη κώδικα που είναι μικρός και αποδοτικός, κάτι που είναι ιδιαίτερα χρήσιμο σε συστήματα όπως microcontrollers και microprocessors. Η υποστήριξη για διαχείριση διακοπών (interrupts), καταχωρητές (registers), και ακριβή χρονοδιαγράμματα είναι απαραίτητη σε τέτοιες εφαρμογές, και η C++ προσφέρει το κατάλληλο εργαλείο για τέτοιες απαιτήσεις.

Παραδείγματα Εφαρμογών της C++

Η C++ χρησιμοποιείται ευρέως σε εφαρμογές επεξεργασίας εικόνας και σήματος λόγω της υψηλής απόδοσης της και της ικανότητας της να χειρίζεται μεγάλες ποσότητες δεδομένων. Βιβλιοθήκες όπως η OpenCV έχουν αναπτυχθεί σε C++ και χρησιμοποιούνται σε πολλά έργα μηχανικής όρασης, όπως η ανίχνευση αντικειμένων, η αναγνώριση προσώπου και η ανάλυση βίντεο.

Η C++ χρησιμοποιείται επίσης για τον προγραμματισμό ενσωματωμένων συστημάτων που περιλαμβάνουν τη χρήση αισθητήρων και επεξεργαστών, όπως τα Arduino και τα Raspberry Pi. Τα συστήματα αυτά χρησιμοποιούν τη C++ για να επικοινωνούν με αισθητήρες, να επεξεργάζονται δεδομένα, και να ελέγχουν εξόδους όπως μοτέρ και φώτα.

Ανάπτυξη Βιβλιοθηκών και Πλατφορμών

Πολλές βασικές βιβλιοθήκες και πλατφόρμες λογισμικού έχουν αναπτυχθεί σε C++, όπως το Qt για ανάπτυξη γραφικών εφαρμογών και η Boost που παρέχει γενικές λειτουργίες για αλγορίθμους και δεδομένα. Αυτές οι βιβλιοθήκες βοηθούν τους προγραμματιστές να επιλύουν σύνθετα προβλήματα χωρίς να χρειάζεται να αναπτύξουν εξ αρχής λειτουργίες.

Η C++ είναι μια από τις πιο σημαντικές γλώσσες προγραμματισμού, ιδιαίτερα για εφαρμογές που απαιτούν υψηλή απόδοση, λεπτομερή διαχείριση του υλικού, και επεξεργασία πολύπλοκων δεδομένων. Η δυνατότητα διαχείρισης μνήμης, η υποστήριξη για διαφορετικά παραδείγματα προγραμματισμού, και η συμβατότητα της με το υλικό καθιστούν την C++ ιδανική για υπολογιστικές εφαρμογές, επεξεργασία εικόνας και σήματος, ενσωματωμένα συστήματα, και ανάπτυξη πλατφορμών λογισμικού. Σε περιπτώσεις όπως η επεξεργασία δεδομένων από αισθητήρες, η C++ δίνει τη δυνατότητα να ελέγχουμε κάθε λεπτομέρεια της διαδικασίας, επιτυγχάνοντας την καλύτερη δυνατή απόδοση. Η ευελιξία της και η αποδοτικότητά της την καθιστούν έναν από τους κύριους "εργάτες" στον χώρο του προγραμματισμού συστημάτων και της ανάπτυξης λογισμικού που απαιτεί ακρίβεια και ταχύτητα.

3.3 ASM AS7261

Ο αισθητήρας **AS7261** είναι ένας ψηφιακός αισθητήρας χρώματος, σχεδιασμένος για την ακριβή μέτρηση των χρωματικών συντεταγμένων XYZ σύμφωνα με το πρότυπο CIE 1931. Επιπλέον, μπορεί να μετατρέπει τις συντεταγμένες XYZ σε x, y του 2-διάστατου χρωματικού χώρου και να τις κλιμακώνει στο σύστημα συντεταγμένων CIE 1976 u'v'. [13-15]

Η συσκευή προσφέρει επίσης ακριβείς μετρήσεις συσχετισμένης θερμοκρασίας χρώματος (CCT) και πληροφορίες για την απόκλιση του σημείου χρώματος από την καμπύλη του μαύρου σώματος στον χώρο των συντεταγμένων delta u'v'. Έχει ενσωματωμένο κανάλι υπερύθρου (NIR) για άλλες εφαρμογές, καθώς και οδηγούς LED με προγραμματιζόμενα ρεύματα για εφαρμογές ηλεκτρονικού κλείστρου.

Ο αισθητήρας AS7261 ενσωματώνει Gaussian φίλτρα στην τυπική CMOS τεχνολογία μέσω τεχνολογίας Nano-optic deposited interference filter, και είναι πακεταρισμένος σε συσκευασία LGA με ενσωματωμένο άνοιγμα για τον έλεγχο του φωτός που εισέρχεται στην περιοχή του αισθητήρα. Τα δεδομένα φασματικής μετατροπής συλλέγονται μέσω δύο ομάδων φωτοδιόδων, με την πρώτη να περιλαμβάνει τα X, Y, Z και NIR, και τη δεύτερη να περιλαμβάνει τα X, Y, D (σκοτεινό) και C (διαυγές).

Λειτουργικά Χαρακτηριστικά

- **Χρωματικές Συντεταγμένες XYZ:** Ο αισθητήρας παρέχει καλιμπραρισμένα δεδομένα για τις συντεταγμένες χρώματος XYZ σύμφωνα με το πρότυπο **CIE 1931**. Αυτές οι συντεταγμένες μπορούν να χρησιμοποιηθούν για την απόδοση χρώματος και για ακριβείς μετρήσεις λευκού φωτός.

- **Συσχετισμένη Θερμοκρασία Χρώματος (CCT):** Ο αισθητήρας παρέχει ακριβείς μετρήσεις CCT, κάτι που είναι πολύ χρήσιμο σε εφαρμογές φωτισμού, όπου απαιτείται έλεγχος της θερμοκρασίας χρώματος.
- **NIR Κανάλι:** Ενσωματώνει ένα κανάλι κοντά στο υπέρυθρο (NIR) φάσμα, επιτρέποντας τη μέτρηση φωτός στην περιοχή αυτή.
- **Διασύνδεση και Χειρισμός:** Η επικοινωνία με τον αισθητήρα γίνεται είτε μέσω διαύλου **PC** είτε μέσω **UART**, επιτρέποντας ευέλικτη ενσωμάτωση με διάφορα συστήματα. Οι εντολές μπορούν να αποσταλούν μέσω ενός απλού κειμενοποιημένου πρωτοκόλλου, γνωστό ως **AT Command Set**, ή άμεσα μέσω των καταχωρητών.

Εφαρμογές

Ο αισθητήρας **AS7261** μπορεί να χρησιμοποιηθεί σε πολλές εφαρμογές, όπως:

- **Μέτρηση και ταυτοποίηση χρώματος:** Η ικανότητα του να μετράει χρώματα ακριβώς τον καθιστά κατάλληλο για εφαρμογές που απαιτούν την ανίχνευση και ταυτοποίηση χρωμάτων.
- **Ρύθμιση και καλιμπράρισμα χρώματος:** Μπορεί να χρησιμοποιηθεί για την ακριβή ρύθμιση χρωμάτων σε φωτιστικά συστήματα.
- **Απορροφητικότητα υλικών:** Ο αισθητήρας μπορεί να χρησιμοποιηθεί για τη μέτρηση της απορροφητικότητας διαφορετικών υλικών, ιδιαίτερα σε φάσματα UV και υπέρυθρου.

και Παραδείγματα Χρήσης

- **Απορροφητικότητα Υλικών και Ανίχνευση Χρώματος**
Ο αισθητήρας **AS7261** μπορεί να χρησιμοποιηθεί για τη μέτρηση και την ανίχνευση χρωμάτων σε διάφορα υλικά. Η ακρίβεια των μετρήσεων στις χρωματικές συντεταγμένες **XYZ** και η δυνατότητα για καλιμπραρισμένες μετρήσεις της θερμοκρασίας χρώματος (**CCT**) επιτρέπει την ανάλυση της απορροφητικότητας των υλικών και τη διαφοροποίηση ανάμεσα σε χρωματικούς τόνους.
- **Ανίχνευση Φασματικής Απόκλισης:** Ο αισθητήρας μπορεί να χρησιμοποιηθεί για τη μέτρηση της φασματικής απόκλισης ενός υλικού, καθιστώντας τον κατάλληλο για εφαρμογές ανάλυσης ποιότητας. Οι πληροφορίες για τις συντεταγμένες **u'v'** στο πρότυπο **CIE 1976** είναι ιδιαίτερα χρήσιμες για τη διαφοροποίηση των χρωμάτων και την ανίχνευση απόκλισης από συγκεκριμένα πρότυπα.
- **Ρύθμιση Φωτισμού σε Εσωτερικούς Χώρους**
Η δυνατότητα του **AS7261** να μετρά ακριβώς τη θερμοκρασία χρώματος (**CCT**) και τη φωτεινότητα (**LUX**) τον καθιστά ιδανικό για τη ρύθμιση φωτισμού σε εσωτερικούς χώρους.

Μπορεί να χρησιμοποιηθεί για να εξασφαλιστεί ότι ο φωτισμός σε ένα δωμάτιο παραμένει σε προκαθορισμένα επίπεδα θερμοκρασίας χρώματος, ανάλογα με τις ανάγκες του χρήστη.

- **Δυναμική Ρύθμιση Φωτισμού:** Με τη χρήση του ηλεκτρονικού κλείστρου και των οδηγών LED, ο αισθητήρας μπορεί να χρησιμοποιηθεί για τον έλεγχο του φωτισμού, ώστε να βελτιωθεί η ποιότητα του φωτός σε πραγματικό χρόνο. Αυτό είναι χρήσιμο σε περιπτώσεις που απαιτείται συγκεκριμένη θερμοκρασία χρώματος, όπως σε γραφεία, σχολικές αίθουσες ή εκθεσιακούς χώρους.
- **Ανάλυση Φυλλώματος**
Ο αισθητήρας **AS7261** μπορεί επίσης να χρησιμοποιηθεί για την ανάλυση της υγείας των φύλλων μέσω της μέτρησης του χρώματος και της απόκρισης στο φως. Το κανάλι **NIR** και οι μετρήσεις στο φάσμα **XYZ** είναι χρήσιμες για την ανάλυση του φυλλώματος και την ανίχνευση αλλαγών που μπορεί να υποδηλώνουν πρόβλημα υγείας στα φυτά.
- **Ανίχνευση Ασθενειών:** Ο αισθητήρας μπορεί να χρησιμοποιηθεί για τη μέτρηση της χρωματικής απόδοσης των φύλλων, ανιχνεύοντας αν υπάρχουν αποχρώσεις καφέ ή απώλεια της πράσινης απόχρωσης, που υποδηλώνουν ασθένειες ή έλλειψη θρεπτικών στοιχείων. Αυτό μπορεί να γίνει με συνεχή παρακολούθηση ή με τη χρήση της λειτουργίας **One-Shot** για περιοδική ανάλυση.

Διαδικασία Συλλογής Δεδομένων

Η συλλογή φασματικών δεδομένων πραγματοποιείται με δύο ομάδες φωτοδιόδων. Κάθε ομάδα συλλέγει δεδομένα από διαφορετικά φάσματα φωτός και τα αποτελέσματα μεταφέρονται στους αντίστοιχους καταχωρητές δεδομένων. Η μετάδοση των αποτελεσμάτων γίνεται με τη χρήση 16-bit ADC με υψηλή ανάλυση, εξασφαλίζοντας την ακρίβεια των μετρήσεων.

Ο αισθητήρας **AS7261** χρησιμοποιεί τράπεζες φωτοδιόδων και υποστηρίζει διαφορετικά Modes για τη συλλογή φασματικών δεδομένων. Αυτά περιλαμβάνουν:

- **Mode 0, Mode 1, Mode 2:** Σε αυτά τα modes, η συλλογή φασματικών δεδομένων γίνεται συνεχώς, και τα δεδομένα αποθηκεύονται στους καταχωρητές **X, Y, Z, NIR, D** και **C**.
- **Mode 3:** Σε αυτό το mode, η λειτουργία είναι **One-Shot**, όπου τα φασματικά δεδομένα συλλέγονται μία φορά, και η διαδικασία επαναλαμβάνεται μόνο με χειροκίνητη εντολή

Ο αισθητήρας προσφέρει επίσης **θερμοκρασία αισθητήρα** μέσω ενός εσωτερικού αισθητήρα θερμοκρασίας, ο οποίος χρησιμοποιείται για τη διαδικασία αντιστάθμισης θερμοκρασίας.

3.3.1 Αρχιτεκτονική και Λειτουργίες του AS7261

Ο αισθητήρας **AS7261** είναι ένα προηγμένο εργαλείο για τη μέτρηση χρωματικών συντεταγμένων και φασματικών δεδομένων. Η αρχιτεκτονική του περιλαμβάνει ένα σύνολο από εσωτερικά μπλοκ που συνεργάζονται για τη μέτρηση και τη μετάδοση δεδομένων.

Ο AS7261 διαθέτει μια σειρά από υποσυστήματα και εξαρτήματα που συνθέτουν την εσωτερική αρχιτεκτονική του:

- **Φωτοδιόδους και Φίλτρα:** Ο αισθητήρας ενσωματώνει έξι φωτοδιόδους, οι οποίες φέρουν φίλτρα παρεμβολής (interference filters) που επιτρέπουν τη μέτρηση των χρωματικών συντεταγμένων X, Y, Z και το κανάλι υπέρυθρου NIR (Near Infrared). Αυτά τα φίλτρα δημιουργούν φασματικές αποκρίσεις που είναι συμβατές με τα πρότυπα CIE 1931 και CIE 1976, διασφαλίζοντας την ακρίβεια των χρωματικών δεδομένων(AS7261_DS000493_1-00).
- **ADC (Analog-to-Digital Converter):** Το ADC του αισθητήρα μετατρέπει τα αναλογικά σήματα από τις φωτοδιόδους σε ψηφιακές τιμές. Ο αισθητήρας διαθέτει 16-bit ADC, το οποίο εξασφαλίζει υψηλή ακρίβεια στις μετρήσεις και ελαχιστοποιεί την ανάγκη για επιπλέον σήματα προσαρμογής(AS7261_DS000493_1-00).
- **Ηλεκτρονικό Κλείστρο (Electronic Shutter):** Ο αισθητήρας ενσωματώνει ένα ηλεκτρονικό κλείστρο μέσω του οδηγού LED (LED_DRV), επιτρέποντας την ακριβή μέτρηση φωτεινότητας από διαφορετικές πηγές φωτός. Οι οδηγοί LED είναι προγραμματιζόμενοι, επιτρέποντας την επιλογή του ρεύματος που διοχετεύεται στο LED σε τιμές 12.5mA, 25mA, 50mA ή 100mA, ανάλογα με την εφαρμογή(AS7261_DS000493_1-00).

Λειτουργία Ολοκληρωμένου Κλειστού Κυκλώματος (Electronic Shutter)

Η χρήση του ηλεκτρονικού κλειστρού μέσω των LED drivers επιτρέπει στον αισθητήρα να ελέγχει τις συνθήκες φωτισμού στις οποίες γίνονται οι μετρήσεις. Αυτό είναι σημαντικό για εφαρμογές που απαιτούν σταθερό φωτισμό, όπως η μέτρηση και η ανάλυση χρώματος σε φωτιστικά ή σε άλλα αντικείμενα. Ο έλεγχος του κλειστρού μπορεί να γίνει μέσω των I²C ή UART διασυνδέσεων με τη χρήση εντολών AT, επιτρέποντας την απομακρυσμένη ρύθμιση των παραμέτρων φωτισμού.

Διασύνδεση με το Σύστημα - UART και I²C

Ο **AS7261** υποστηρίζει δύο διαφορετικές μεθόδους διασύνδεσης για επικοινωνία και διαχείριση των λειτουργιών του: **I²C** και **UART**.

- **I²C Διασύνδεση:** Η διασύνδεση I²C επιτρέπει την επικοινωνία με το αισθητήρα μέσω μιας σειράς εικονικών καταχωρητών. Η επικοινωνία γίνεται με χρήση τυπικού πρωτοκόλλου 7-bit addressing, ενώ υποστηρίζονται λειτουργίες write και read με ρυθμούς έως 400 kHz (Fast Mode). Οι καταχωρητές του αισθητήρα περιλαμβάνουν καταχωρητές για τον έλεγχο του LED, τη ρύθμιση του χρόνου ολοκλήρωσης (integration time) και την αποθήκευση των μετρημένων φασματικών δεδομένων(AS7261_DS000493_1-00).
- **UART Διασύνδεση:** Η διασύνδεση UART υποστηρίζει πλήρη διπλή λειτουργία (Full Duplex), με ανεξάρτητα κανάλια για τη λήψη και τη μετάδοση δεδομένων. Υπάρχει δυνατότητα αποθήκευσης των δεδομένων σε 16 θέση FIFO buffer για κάθε διαδρομή (λήψη και μετάδοση). Το πρωτόκολλο υποστηρίζει ρυθμό 115200 Baud και η επικοινωνία μπορεί να γίνει με εντολές AT για εύκολη διαχείριση των λειτουργιών του αισθητήρα(AS7261_DS000493_1-00).
- **Επιλογή Διασύνδεσης:** Ο χρήστης μπορεί να επιλέξει μεταξύ των δύο διασυνδέσεων μέσω του I2C_ENB pin. Αν το I2C_ENB είναι υψηλό, τότε ενεργοποιείται η διασύνδεση I²C, ενώ όταν είναι χαμηλό ενεργοποιείται η διασύνδεση UART(AS7261_DS000493_1-00).

Χειρισμός Δεδομένων του Αισθητήρα AS7261

Ο αισθητήρας AS7261 συλλέγει φασματικά δεδομένα μέσω των φωτοδιόδων που περιέχει, κάθε μία από τις οποίες καλύπτει διαφορετικό φάσμα φωτός. Οι έξι φωτοδιόδοι του αισθητήρα είναι ευαίσθητες στα φάσματα X, Y, Z (σύμφωνα με το πρότυπο CIE 1931), NIR (κοντά στο υπέρυθρο), Clear (διανυγές) και Dark (σκοτεινό). Αυτή η συλλογή φασματικών δεδομένων είναι απαραίτητη για την απόδοση ακριβών χρωματικών μετρήσεων(AS7261_DS000493_1-00).

Η διαδικασία περιλαμβάνει τα εξής βήματα:

- **Μετατροπή Αναλογικών Σημάτων σε Ψηφιακά (ADC):** Οι αναλογικές μετρήσεις που παράγονται από τις φωτοδιόδους μετατρέπονται σε ψηφιακές τιμές μέσω ενός 16-bit ADC (Analog-to-Digital Converter). Το ADC συλλέγει το ρεύμα από κάθε φωτοδίοδο και το μετατρέπει σε ψηφιακή μορφή για περαιτέρω επεξεργασία(AS7261_DS000493_1-00).
- **Διπλή Αποθήκευση Δεδομένων (Double Buffering):** Οι μετρήσεις που προκύπτουν από τη μετατροπή αποθηκεύονται σε καταχωρητές δεδομένων, με τη χρήση της τεχνικής της διπλής αποθήκευσης (double buffering). Αυτό σημαίνει ότι οι μετατροπές αποθηκεύονται σε δύο βήματα για να εξασφαλιστεί η ακρίβεια των δεδομένων, αποφεύγοντας απώλειες που μπορεί να προκύψουν από την ταυτόχρονη συλλογή νέων δεδομένων(AS7261_DS000493_1-00).

Ρύθμιση Τράπεζας Φωτοδιόδων και Τρόποι Λειτουργίας

Ο αισθητήρας **AS7261** χρησιμοποιεί τέσσερις διαφορετικούς τρόπους λειτουργίας (**BANK Modes**) για τη συλλογή δεδομένων από τις φωτοδιόδους του. Ο καθένας από αυτούς τους τρόπους επιτρέπει διαφορετικό συνδυασμό φασματικών δεδομένων:

- **Mode 0:** Συλλέγει δεδομένα από τα κανάλια **X, Y, Z** και **NIR**. Τα δεδομένα συλλέγονται συνεχώς, και κάθε νέο σετ δεδομένων είναι διαθέσιμο μετά από κάθε ολοκλήρωση μετατροπής.
- **Mode 1:** Συλλέγει δεδομένα από τα κανάλια **X, Y, Dark**, και **Clear**. Όπως και στο **Mode 0**, η συλλογή είναι συνεχής.
- **Mode 2:** Σε αυτό το mode, γίνεται συλλογή δεδομένων από όλα τα κανάλια (**X, Y, Z, NIR, Dark, Clear**), απαιτώντας δύο διαδοχικές ολοκληρώσεις μετατροπής για την απόδοση πλήρων δεδομένων. Η λειτουργία αυτή χρησιμοποιείται όταν χρειάζονται και οι δύο τράπεζες φωτοδιόδων για την εξαγωγή ακριβών δεδομένων.
- **Mode 3 (One-Shot Mode):** Αυτή η λειτουργία επιτρέπει τη συλλογή δεδομένων από όλες τις φωτοδιόδους σε μία και μόνο διαδικασία συλλογής (One-Shot), η οποία εκτελείται όταν απαιτείται ταυτόχρονη λήψη δεδομένων από όλες τις πηγές. Το **DATA_RDY** bit ενεργοποιείται όταν τα δεδομένα είναι διαθέσιμα για ανάγνωση(AS7261_DS000493_1-00).

Θερμοκρασία και Αντιστάθμιση

Ο αισθητήρας **AS7261** διαθέτει ενσωματωμένο αισθητήρα θερμοκρασίας, ο οποίος μετρά συνεχώς τη θερμοκρασία του αισθητήρα και χρησιμοποιείται για την αντιστάθμιση των αποτελεσμάτων. Η θερμοκρασία μπορεί να επηρεάσει τις μετρήσεις του αισθητήρα, ειδικά σε περιπτώσεις που το περιβάλλον λειτουργίας παρουσιάζει μεγάλες διακυμάνσεις θερμοκρασίας.

- **Αντιστάθμιση Θερμοκρασίας:** Τα φασματικά δεδομένα που συλλέγονται διορθώνονται με βάση τη θερμοκρασία που ανιχνεύει ο εσωτερικός αισθητήρας. Αυτό επιτυγχάνεται μέσω ενός συντελεστή αντιστάθμισης που εφαρμόζεται στα δεδομένα κατά τη διάρκεια της επεξεργασίας τους. Αυτός ο συντελεστής προέρχεται από έναν πίνακα συντελεστών αντιστάθμισης θερμοκρασίας (temperature coefficient table), ο οποίος έχει δημιουργηθεί κατά την καλιμπράρισμα του αισθητήρα(AS7261_DS000493_1-00).

Βασικά και Καλιμπραρισμένα Δεδομένα

Τα δεδομένα που συλλέγει ο αισθητήρας χωρίζονται σε τρεις βασικές κατηγορίες:

- **Raw Data (Ακατέργαστα Δεδομένα):** Αυτά είναι τα πρωτογενή δεδομένα που συλλέγονται από το **ADC** και εξαρτώνται από τη ρύθμιση της κλίμακας ενίσχυσης (**Gain**) και του χρόνου ολοκλήρωσης (**Integration Time**).

- **Basic Sensor Values:** Τα βασικά δεδομένα παράγονται διαιρώντας τις τιμές των ακατέργαστων δεδομένων με κώδικες που καθορίζονται από τις παραμέτρους της ρύθμισης. Τα βασικά δεδομένα διορθώνονται επίσης με βάση τον συντελεστή θερμοκρασίας που εξάγεται από τον εσωτερικό αισθητήρα θερμοκρασίας.
- **Calibrated Data (Καλιμπραρισμένα Δεδομένα):** Αυτά τα δεδομένα είναι καλιμπραρισμένα από το εργοστάσιο και προσφέρουν μεγαλύτερη ακρίβεια, δεδομένου ότι λαμβάνουν υπόψη παραμέτρους όπως η ενίσχυση, ο χρόνος ολοκλήρωσης και η θερμοκρασία. Τα καλιμπραρισμένα δεδομένα περιλαμβάνουν μετρήσεις για τις χρωματικές συντεταγμένες **XYZ**, την **Correlated Color Temperature (CCT)**, και τη φωτεινότητα (**LUX**) (AS7261_DS000493_1-00).

Ρύθμιση και Διαχείριση Καταχωρητών μέσω I²C

Ο αισθητήρας **AS7261** χρησιμοποιεί μια σειρά από **εικονικούς καταχωρητές** για τη διαχείριση των λειτουργιών του και την αποθήκευση δεδομένων. Αυτοί οι καταχωρητές επιτρέπουν την άμεση επικοινωνία με το σύστημα και προσφέρουν τη δυνατότητα τόσο για ανάγνωση όσο και για εγγραφή δεδομένων. Η πρόσβαση στους καταχωρητές γίνεται μέσω του διαύλου **I²C**, χρησιμοποιώντας το πρωτόκολλο 7+1-bit addressing.

- **Καταχωρητές Ελέγχου:** Χρησιμοποιούνται για τον έλεγχο των λειτουργιών του αισθητήρα, όπως η ενεργοποίηση του **LED**, ο καθορισμός του χρόνου ολοκλήρωσης, και η ρύθμιση του τρόπου λειτουργίας.
- **Καταχωρητές Δεδομένων:** Αποθηκεύουν τα φασματικά δεδομένα που προκύπτουν από τη συλλογή και την επεξεργασία των αναλογικών σημάτων των φωτοдиодών

Η **I²C** διεπαφή του αισθητήρα **AS7261** επιτρέπει τη διαχείριση των λειτουργιών του και την ανάγνωση των φασματικών δεδομένων μέσω μιας σειράς εικονικών καταχωρητών. Οι καταχωρητές αυτοί παρέχουν πρόσβαση στις διαφορετικές ρυθμίσεις και τις πληροφορίες του αισθητήρα, επιτρέποντας την ενσωμάτωσή του σε ευρύτερα συστήματα.

Διαχείριση Εικονικών Καταχωρητών

Οι **εικονικοί καταχωρητές** του **AS7261** είναι βασικά εικονικές οντότητες που υλοποιούνται μέσω λογισμικού, ενώ η πραγματική υλοποίηση στον αισθητήρα περιλαμβάνει τρεις βασικούς καταχωρητές **I²C**: τον **STATUS Register**, τον **WRITE Register**, και τον **READ Register**.

- **STATUS Register (Διεύθυνση: 0x00):** Παρέχει πληροφορίες για την κατάσταση της διεπαφής **I²C** και την ετοιμότητα των καταχωρητών για ανάγνωση ή εγγραφή. Περιλαμβάνει τις σημαίες

TX_VALID και **RX_VALID**, οι οποίες δείχνουν εάν υπάρχει εισερχόμενη εγγραφή ή αν υπάρχουν δεδομένα διαθέσιμα προς ανάγνωση, αντίστοιχα.

- **WRITE Register (Διεύθυνση: 0x01):** Χρησιμοποιείται για την εγγραφή δεδομένων στο σύστημα. Τα δεδομένα μπορούν να είναι είτε διευθύνσεις εικονικών καταχωρητών είτε δεδομένα προς εγγραφή.
- **READ Register (Διεύθυνση: 0x02):** Χρησιμοποιείται για την ανάγνωση δεδομένων από τους εικονικούς καταχωρητές. Οι τιμές που συλλέγονται από τους αισθητήρες αποθηκεύονται στον **READ Register** και μπορούν να αναγνωστούν μέσω του διαύλου **I²C**.

Ροή Επικοινωνίας με Εικονικούς Καταχωρητές

Για την επικοινωνία με τους εικονικούς καταχωρητές, ο χρήστης πρέπει να ακολουθήσει συγκεκριμένη ροή εντολών. Αυτή η διαδικασία περιλαμβάνει την επανάληψη της ανάγνωσης της κατάστασης (**STATUS Register**) και, εάν η σημαία **TX_VALID** είναι 0, μπορεί να ξεκινήσει η εγγραφή στον καταχωρητή. Η διαδικασία είναι παρόμοια και για την ανάγνωση δεδομένων, με τη χρήση της σημαίας **RX_VALID**.

- **Εγγραφή στον Καταχωρητή:**
 1. Ανάγνωση της κατάστασης από τον **STATUS Register**.
 2. Εάν **TX_VALID** είναι 0, εγγραφή της διεύθυνσης του εικονικού καταχωρητή στο **WRITE Register**.
 3. Επαλήθευση της κατάστασης, και στη συνέχεια εγγραφή των δεδομένων.
- **Ανάγνωση από τον Καταχωρητή:**
 1. Ανάγνωση της κατάστασης από τον **STATUS Register**.
 2. Εάν **RX_VALID** είναι 1, τα δεδομένα είναι έτοιμα για ανάγνωση από τον **READ Register**.

Αυτή η διαδικασία επιτρέπει τη διαχείριση των διαφόρων παραμέτρων του αισθητήρα, όπως τη ρύθμιση της ενίσχυσης (**GAIN**), τον χρόνο ολοκλήρωσης (**Integration Time**), και την ενεργοποίηση των **LED**.

3.4 AS726x-iSPI.Evaluation.Kit

Θα εξετάσουμε το **AS726x-iSPI Evaluation Kit** και το λογισμικό **AS726x Spectral Sensing iSPI Application**, τα οποία χρησιμοποιούνται για την αξιολόγηση και ανάπτυξη λύσεων με τον αισθητήρα

AS7261. Τα εργαλεία αυτά προσφέρουν μια ολοκληρωμένη πλατφόρμα για τη δοκιμή των λειτουργιών του αισθητήρα, την ανάλυση των δεδομένων και την ενσωμάτωσή τους σε διάφορα συστήματα.

Περιεχόμενα του Evaluation Kit

Το **AS726x-iSPI Evaluation Kit** είναι ένα εργαλείο αξιολόγησης για τον αισθητήρα **AS7261** και τις υπόλοιπες συσκευές της σειράς AS726x. Περιλαμβάνει τα εξής:

- **AS726x-iSPI Board:** Πρόκειται για την κύρια πλακέτα αξιολόγησης που περιλαμβάνει τον αισθητήρα **AS7261** και άλλες βασικές υποδομές, όπως τα κυκλώματα μνήμης και τα κυκλώματα επικοινωνίας για τη διασύνδεση με υπολογιστή(AS726x-iSPI.Evaluation....).
- **FTDI USB-MPSSE Cable:** Καλώδιο για τη σύνδεση της πλακέτας με τον υπολογιστή μέσω USB, παρέχοντας ισχύ στην πλακέτα (3.3V) και επιτρέποντας την επικοινωνία μέσω I²C ή UART.
- **USB Memory Stick:** Περιλαμβάνει έγγραφα και λογισμικό που απαιτείται για την εγκατάσταση και τη χρήση του Evaluation Kit, όπως οδηγοί (drivers) και το εκτελέσιμο αρχείο του λογισμικού GUI(AS726x-iSPI.Evaluation....).

Σύνδεση και Εγκατάσταση του Evaluation Kit

Η σύνδεση του **AS726x-iSPI Evaluation Kit** είναι απλή και περιλαμβάνει τα εξής βήματα:

1. **Σύνδεση του FTDI USB-MPSSE Cable:** Συνδέστε το καλώδιο στην υποδοχή **J2** της πλακέτας, φροντίζοντας το κόκκινο καλώδιο να είναι συνδεδεμένο με το **Pin 1**. Το καλώδιο αυτό παρέχει τάση 3.3V και επιτρέπει την επικοινωνία μεταξύ της πλακέτας και του υπολογιστή.
2. **Εγκατάσταση του FTDI Driver:** Αν ο οδηγός δεν είναι ήδη εγκατεστημένος, μπορείτε να τον βρείτε στο USB Memory Stick και να τον εγκαταστήσετε. Ο οδηγός αυτός απαιτείται για τη χρήση του **FTDI USB-MPSSE Cable**(AS726x-iSPI.Evaluation....).
3. **Λειτουργία του GUI:** Δεν απαιτείται εγκατάσταση του λογισμικού **AS726x Spectral Sensing iSPI Application**. Απλά αντιγράψτε τα αρχεία από το USB Memory Stick στον υπολογιστή σας και εκτελέστε το αρχείο .exe για να ξεκινήσετε τη γραφική διεπαφή(AS726x-iSPI.Evaluation....).

Αρχιτεκτονική Hardware του Evaluation Kit

Το **AS726x-iSPI Evaluation Kit** αποτελείται από τα εξής κύρια τμήματα:

- **AS726x Intelligent Spectral Sensor:** Το ολοκληρωμένο αυτό περιλαμβάνει τον αισθητήρα και τους απαραίτητους καταχωρητές για τη συλλογή και επεξεργασία φασματικών δεδομένων.
- **Flash Memory:** Χρησιμοποιείται για την αποθήκευση του λογισμικού και των δεδομένων που συλλέγονται από τον αισθητήρα.
- **Jumper Configuration Header:** Επιτρέπει την επιλογή του τρόπου επικοινωνίας μεταξύ UART και I²C. Η προεπιλεγμένη ρύθμιση είναι για I²C(AS726x-iSPI.Evaluation...).

AS726x Spectral Sensing iSPI Application GUI

Το λογισμικό **AS726x Spectral Sensing iSPI Application** παρέχει μια γραφική διεπαφή χρήστη (GUI) για την εύκολη αξιολόγηση των δυνατοτήτων του αισθητήρα **AS7261**. Μέσω αυτής της εφαρμογής, μπορείτε να:

- **Παρακολουθήσετε τις Τιμές των Καναλιών XYZ και NIR:** Τα δεδομένα εμφανίζονται σε πραγματικό χρόνο, επιτρέποντας την ανάλυση της φασματικής απόκρισης του αισθητήρα.
- **Ελέγξτε τα Ρεύματα των LED:** Μπορείτε να ρυθμίσετε το ρεύμα που διοχετεύεται στους οδηγούς LED για να ελέγξετε τον φωτισμό της περιοχής που μετράται από τον αισθητήρα.
- **Αποθήκευση και Ανάλυση Δεδομένων:** Τα δεδομένα που συλλέγονται μπορούν να αποθηκευτούν για περαιτέρω ανάλυση ή για σύγκριση με άλλα δεδομένα από προηγούμενες μετρήσεις.

Ρυθμίσεις Επικοινωνίας

Η πλακέτα του **Evaluation Kit** υποστηρίζει δύο πρωτόκολλα επικοινωνίας: I²C και UART. Οι διακόπτες (jumpers) **J3** και **J4** επιτρέπουν την επιλογή της επικοινωνίας που προτιμάτε:

- **UART Επικοινωνία:** Αν θέλετε να χρησιμοποιήσετε UART, θα πρέπει να ορίσετε το **J3** σε **OFF** και το **J4** σε **ON**. Στη συνέχεια, μπορείτε να επικοινωνήσετε με την πλακέτα χρησιμοποιώντας οποιοδήποτε λογισμικό τερματικού, ρυθμισμένο σε **115200 Baud, 8-bit data, χωρίς parity, και 1 stop bit**.
- **I²C Επικοινωνία:** Η προεπιλογή της πλακέτας είναι η I²C επικοινωνία, με το **J3** σε **ON** και το **J4** σε **OFF**. Η αλλαγή της επικοινωνίας απαιτεί την αποσύνδεση και την επανασύνδεση του καλωδίου για να αναγνωριστεί σωστά η νέα ρύθμιση(AS726x-iSPI.Evaluation...).

Χρήση του Evaluation Kit για Ανάλυση Δεδομένων

Το AS726x-iSPI Evaluation Kit επιτρέπει στους χρήστες να αναλύσουν τα δεδομένα που συλλέγονται από τον αισθητήρα σε πραγματικό χρόνο, ενώ το GUI βοηθά στην απεικόνιση των μετρήσεων με κατανοητό τρόπο:

- **Φασματική Ανάλυση:** Μπορείτε να δείτε τα δεδομένα από τα κανάλια XYZ και NIR, καθώς και να αναλύσετε τη θερμοκρασία χρώματος (CCT).
- **Δυνατότητα Χρήσης AT Εντολών:** Η εφαρμογή υποστηρίζει τη χρήση **AT commands** για την απευθείας ρύθμιση και διαχείριση των λειτουργιών του αισθητήρα. Μπορείτε να ρυθμίσετε τα ρεύματα των LED και να ελέγξετε την κατάσταση του αισθητήρα μέσω αυτών των εντολών (AS726x-iSPI.Evaluation...).

3.4.1 Η πειραματική διάταξη

Στην Εικόνα 3.1 παρουσιάζεται η διάταξη έτοιμη για μέτρηση του χρώματος και του NIR σε ένα κυλινδρικό σωληνάριο με καθαρό νερό και στην επόμενη το ίδιο με πράσινου χρώμα.



Εικόνα 3.1: Πειραματική Διάταξη με κυλινδρικό σωληνάριο με καθαρό νερό



Εικόνα 3.2: Πειραματική Διάταξη με κυλινδρικό σωληνάριο με πράσινο χρώμα νερό

Κεφάλαιο 4ο: Το σύστημα μέτρησης

4.1 Εισαγωγή

Ο αισθητήρας AS7261 είναι ένας προηγμένος αισθητήρας μέτρησης χρωμάτων, ο οποίος χρησιμοποιεί το φως για να μετρήσει τις χρωματικές συντεταγμένες X, Y, Z σύμφωνα με το πρότυπο CIE 1931. Η λειτουργία του βασίζεται στη διέλευση φωτός μέσα από ένα σωληνάκι που περιέχει δείγμα, το οποίο φωτίζεται από ενσωματωμένα LED.

Ο AS7261 μετρά τις χρωματικές συντεταγμένες X, Y, Z, οι οποίες αντιστοιχούν στις αντιλήψεις του ανθρώπινου ματιού για την αίσθηση των χρωμάτων. Ο αισθητήρας παράγει άμεσα τις τιμές X, Y, Z από την ανίχνευση του φωτός που διαχέεται από το δείγμα. Το φως αυτό εκπέμπεται από ενσωματωμένα προγραμματιζόμενα LED, τα οποία ελέγχονται μέσω του ηλεκτρονικού κλείστρου του αισθητήρα.

Ο αισθητήρας διαθέτει δύο κύριες τράπεζες φωτοдиодων:

- **Τράπεζα 1** που περιέχει τις φωτοдиодους για τις τιμές X, Y, Z και NIR (κοντά στο υπέρυθρο).
- **Τράπεζα 2** που περιλαμβάνει επιπλέον αισθητήρες για καθαρές τιμές φωτεινότητας και σκοτεινές διορθώσεις.

Κατά τη διάρκεια της μέτρησης, το φως εισέρχεται στο σωληνάκι και διαχέεται στο δείγμα, ενώ οι φωτοдиодοι του αισθητήρα ανιχνεύουν το φως και μετατρέπουν τα δεδομένα σε ψηφιακή μορφή.

Η φασματική μετατροπή στον AS7261 επιτυγχάνεται μέσω των δύο τραπεζών φωτοдиодων και η διάρκεια της ολοκλήρωσης εξαρτάται από το χρόνο ολοκλήρωσης της ενσωμάτωσης (Integration Time). Εάν χρησιμοποιηθούν και οι δύο τράπεζες, η συσκευή απαιτεί δύο πλήρεις κύκλους μετατροπής για την παραγωγή των τιμών X, Y, Z.

Ο AS7261 είναι ένας ισχυρός αισθητήρας μέτρησης χρωμάτων που επιτρέπει την ακριβή μέτρηση των τιμών X, Y, Z. Οι τιμές αυτές είναι χρήσιμες σε εφαρμογές όπως η μέτρηση χρωμάτων, η απορρόφηση φωτός και η αναγνώριση χρωμάτων σε βιομηχανικές και επιστημονικές εφαρμογές.

4.1.1 Η εφαρμογή

Η λειτουργία της εφαρμογής αφορά τη μέτρηση και επεξεργασία χρωματικών δεδομένων χρησιμοποιώντας έναν αισθητήρα AS7261, ο οποίος είναι εξειδικευμένος για τη μέτρηση χρωματικών συντεταγμένων XYZ. Η εφαρμογή αυτή έχει ως στόχο να διαβάσει τα δεδομένα από τον αισθητήρα, να τα μετατρέψει σε RGB τιμές και στη συνέχεια να τα αποστείλει σε ένα απομακρυσμένο διακομιστή μέσω ενός HTTP αιτήματος.

Αρχικά, ο αισθητήρας AS7261 χρησιμοποιείται για τη μέτρηση των τιμών φωτεινότητας από ένα δείγμα που φωτίζεται από ένα LED. Ο αισθητήρας διαθέτει δύο τράπεζες φωτοдиодων που ανιχνεύουν το φως και μετατρέπουν τις αναλογικές μετρήσεις σε ψηφιακές τιμές. Αυτές οι τιμές αντιστοιχούν στις χρωματικές συντεταγμένες X, Y και Z σύμφωνα με το πρότυπο CIE 1931, το οποίο χρησιμοποιείται για την τυποποιημένη αναπαράσταση των χρωμάτων.

Μετά τη λήψη των χρωματικών δεδομένων, τα δεδομένα μετατρέπονται από τις χρωματικές συντεταγμένες XYZ σε τιμές RGB. Αυτή η διαδικασία απαιτεί τον υπολογισμό των τιμών Red, Green και Blue (RGB), οι οποίες αντιπροσωπεύουν το χρώμα σε μια μορφή που μπορεί να απεικονιστεί σε οθόνες και άλλες συσκευές απεικόνισης. Η μετατροπή αυτή πραγματοποιείται μέσω μαθηματικών αλγορίθμων που λαμβάνουν υπόψη τα χαρακτηριστικά του φωτός και τη χρωματική κλίμακα (sRGB) που χρησιμοποιείται στις περισσότερες οθόνες.

Αφού ολοκληρωθεί η μετατροπή, οι τιμές RGB αποστέλλονται σε έναν απομακρυσμένο διακομιστή μέσω ενός HTTP αιτήματος. Η αποστολή των δεδομένων πραγματοποιείται σε συγκεκριμένα χρονικά διαστήματα, διασφαλίζοντας έτσι ότι οι πληροφορίες είναι διαθέσιμες στον διακομιστή σε πραγματικό χρόνο. Το αίτημα περιλαμβάνει τις τιμές Red, Green και Blue και NIR καθώς και έναν μοναδικό αριθμό που δημιουργείται βάσει της χρονικής στιγμής, προκειμένου να διαχωρίζονται τα αιτήματα.

Η εφαρμογή λειτουργεί σε έναν αέριο βρόχο, κατά τη διάρκεια του οποίου ενεργοποιεί το LED του αισθητήρα, διαβάζει τα δεδομένα, τα μετατρέπει και τα αποστέλλει στον διακομιστή κάθε δέκα δευτερόλεπτα. Μετά την ανάγνωση και αποστολή των δεδομένων, το LED απενεργοποιείται μέχρι τον επόμενο κύκλο μέτρησης.

Η συνολική λειτουργία της εφαρμογής στοχεύει στην ακριβή και συνεχή καταγραφή χρωματικών δεδομένων και την αποστολή τους για απομακρυσμένη επεξεργασία ή απεικόνιση, διασφαλίζοντας την ακεραιότητα και ακρίβεια των μετρήσεων μέσω του ενσωματωμένου συστήματος διαχείρισης φωτισμού και αισθητήρων του AS7261.

4.1.2 Η εργασία

Η εργασία μας αφορά τη μέτρηση της ποσότητας άλης σε ένα σωληνάκι, χρησιμοποιώντας τεχνολογίες χρωματικής ανάλυσης και μέτρησης του κοντά στο υπέρυθρο (NIR) φάσματος. Ο αισθητήρας AS7261 είναι το βασικό εργαλείο για την καταγραφή των δεδομένων που προέρχονται από τις μετρήσεις αυτές. Η διαδικασία περιλαμβάνει δύο στάδια:

1. **Αρχική Μέτρηση με Άδειο Σωληνάκι:** Πριν εισάγουμε το περιεχόμενο, γίνεται αρχική μέτρηση του άδειου σωληνακίου. Αυτή η μέτρηση χρησιμοποιείται ως αναφορά, ώστε να ληφθούν υπόψη οι επιδράσεις του ίδιου του σωλήνα στο φως που ανιχνεύει ο αισθητήρας. Οι τιμές που λαμβάνονται περιλαμβάνουν τις χρωματικές συντεταγμένες XYZ, καθώς και

μετρήσεις από το φάσμα του κοντά στο υπέρυθρο (NIR), το οποίο είναι χρήσιμο για την ανίχνευση της συγκέντρωσης άλγης.

2. **Μέτρηση με Περιεχόμενο:** Αφού τοποθετηθεί το περιεχόμενο (άλγη) μέσα στο σωληνάκι, πραγματοποιούμε νέες μετρήσεις, χρησιμοποιώντας τον ίδιο αισθητήρα AS7261. Αυτές οι μετρήσεις περιλαμβάνουν τόσο τις τιμές χρωματικών συντεταγμένων όσο και τις τιμές NIR. Ο σκοπός αυτής της δεύτερης μέτρησης είναι να συγκριθούν οι τιμές με τις αρχικές για την εκτίμηση της ποσότητας άλγης.

Αποθήκευση και Οργάνωση Δεδομένων

Κάθε μέτρηση αποθηκεύεται στη βάση δεδομένων και κατατάσσεται σε συγκεκριμένα groups, που αντιστοιχούν σε κάθε σειρά μετρήσεων. Η οργάνωση των δεδομένων ανά group διευκολύνει τη σύγκριση και την ανάλυση μεταξύ των διαφόρων μετρήσεων, καθώς και την παρακολούθηση της εξέλιξης της άλγης σε επαναλαμβανόμενα πειράματα.

Προβολή Δεδομένων

Τα αποθηκευμένα δεδομένα είναι προσβάσιμα μέσω μιας ιστοσελίδας, η οποία εμφανίζει τις μετρήσεις ανά group. Η προβολή αυτή επιτρέπει την εύκολη σύγκριση των χρωματικών και NIR δεδομένων ανάμεσα στις διάφορες καταστάσεις του σωληνακίου (άδειο ή γεμάτο), ενώ παρέχει επίσης οπτικοποίηση των μετρήσεων με τρόπο κατανοητό για τον χρήστη.

4.1.3 Ανάλυση Δεδομένων

Μετά την αποθήκευση και την προβολή των δεδομένων, πραγματοποιείται ανάλυση με σκοπό να εξαχθούν συμπεράσματα για την ποσότητα της άλγης στο δείγμα. Η χρωματική ανάλυση και η μέτρηση NIR προσφέρουν μια ποσοτική εκτίμηση της συγκέντρωσης της άλγης, καθώς οι διαφορές στις χρωματικές συντεταγμένες XYZ και στις μετρήσεις NIR υποδηλώνουν αλλαγές στην ποσότητα βιομάζας που βρίσκεται στο σωληνάκι.

Η ανάλυση με βάση τις τιμές RGB και NIR μπορεί να βοηθήσει στην ποσοτική εκτίμηση της ποσότητας άλγης μέσα στο σωληνάκι. Οι τιμές Red, Green και Blue (RGB) που προκύπτουν από τις μετρήσεις του αισθητήρα AS7261 μπορούν να χρησιμοποιηθούν για την εκτίμηση της πυκνότητας της άλγης μέσω της απόχρωσης και της φωτεινότητας του δείγματος.

4.1.3.1 Ανάλυση Βασισμένη στη Σύγκριση RGB Τιμών

- **Μετατροπή των RGB τιμών σε Hue-Saturation-Value (HSV):** Ο χρωματικός χώρος HSV είναι πιο εύκολος για την ανάλυση χρώματος, καθώς διαχωρίζει την απόχρωση (hue) από την ένταση (saturation) και τη φωτεινότητα (value). Αλλαγές στην απόχρωση και τη φωτεινότητα μπορεί να υποδεικνύουν την αύξηση της βιομάζας της άλγης.
- **Σύγκριση μετρήσεων με άδειο και γεμάτο σωληνάκι:** Με τη σύγκριση των RGB τιμών του άδειου σωληνακίου με τις τιμές όταν το σωληνάκι περιέχει άλγη, μπορεί να παρατηρηθεί η διαφορά στην απορρόφηση και αντανάκλαση του φωτός. Όσο πιο πράσινη είναι η απόχρωση, τόσο μεγαλύτερη είναι η συγκέντρωση της άλγης.
- **Δείκτης Χρώματος (Color Index):** Μέσω του δείκτη χρώματος και την αναλογία μεταξύ των καναλιών Red, Green και Blue, το πράσινο (Green) συνήθως κυριαρχεί στις μετρήσεις άλγης. Ένας δείκτης όπως το **Green-to-Red Ratio (G/R)** μπορεί να χρησιμοποιηθεί για να εκτιμήσουμε την ποσότητα της άλγης, με υψηλότερες τιμές να υποδηλώνουν αυξημένη βιομάζα.

4.1.3.2 Ανάλυση Βασισμένη στις Τιμές NIR (Κοντά στο Υπέρυθρο)

Η NIR ανάλυση χρησιμοποιεί το κοντά στο υπέρυθρο φάσμα (Near-Infrared) για να εντοπίσει τη συγκέντρωση βιομάζας, καθώς η άλγη απορροφά διαφορετικά τα υπέρυθρα μήκη κύματος.

- **Δείκτης Φυτοπλαγκτού (Normalized Difference Vegetation Index - NDVI):** Μια καλή προσέγγιση για την ανάλυση της άλγης είναι να χρησιμοποιηθεί ο NDVI, ένας δείκτης που χρησιμοποιείται στη γεωργία για την εκτίμηση της βλάστησης. Ο NDVI υπολογίζεται ως:

$$NDVI = \frac{NIR - Red}{NIR + Red} \quad (Εξ.1)$$

Υψηλότερες τιμές του NDVI υποδεικνύουν μεγαλύτερη συγκέντρωση άλγης, καθώς το NIR φως αντανακλάται περισσότερο από την υγιή βιομάζα και λιγότερο από το νερό ή το άδειο σωληνάκι.

- **Αναλογία NIR προς RGB:** Η αναλογία της τιμής NIR προς την τιμή Green ή τη συνολική φωτεινότητα μπορεί να προσφέρει μια καλή ένδειξη της ποσότητας άλγης. Με την αύξηση της άλγης, αναμένεται η τιμή του NIR να αυξάνεται λόγω της υψηλότερης αντανάκλασης στο φάσμα αυτό.

4.1.3.3 Υπολογισμός Δείκτη Απορρόφησης Φωτός

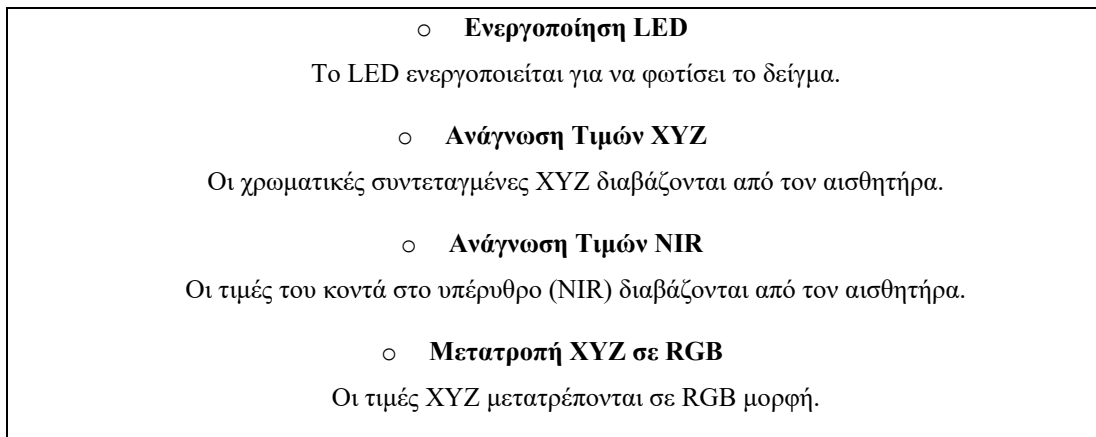
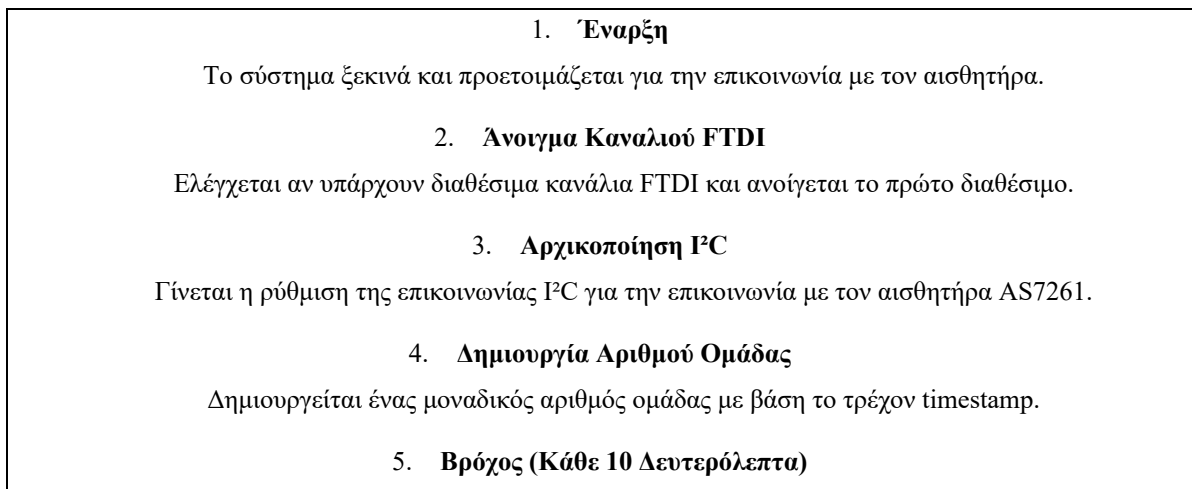
Η άλγη έχει χαρακτηριστικά απορρόφησης φωτός σε συγκεκριμένα μήκη κύματος, και η ανάλυση των δεδομένων RGB και NIR μπορεί να γίνει με στόχο την εκτίμηση της απορρόφησης. Ο δείκτης απορρόφησης μπορεί να υπολογιστεί με βάση τη διαφορά των τιμών πριν και μετά την εισαγωγή της άλγης:

$$\text{Δείκτης Απορρόφησης} = \frac{(RGB \text{ αρχικό} - RGB \text{ τελικό}) + (NIR \text{ αρχικό} - NIR \text{ τελικό})}{RGB \text{ αρχικό} + NIR \text{ αρχικό}} \quad (\text{Εξ.2})$$

Αυτός ο δείκτης μπορεί να χρησιμοποιηθεί για την εκτίμηση της ποσότητας βιομάζας που απορροφά το φως και ενδεχομένως να συσχετιστεί με την αύξηση της άλγης.

Το πρόγραμμα για την υλοποίηση της διαδικασίας

Ακολουθεί μια περιγραφή του διαγράμματος ροής βάσει του κώδικα:



○ **Αποστολή Δεδομένων στο API**

Οι τιμές RGB και NIR αποστέλλονται στο API για περαιτέρω επεξεργασία.

○ **Απενεργοποίηση LED**

Το LED απενεργοποιείται μετά τη συλλογή των δεδομένων.

6. Κλείσιμο Καναλιών FTDI/I²C

Μετά τη λήξη του βρόχου, οι συνδέσεις I²C και FTDI κλείνουν.

7. Τέλος

Το σύστημα τερματίζει τη λειτουργία του.

Εικόνα 4.1: Διάγραμμα ροής της λειτουργίας μέτρησης μέσω i2c

Το διάγραμμα ροής περιγράφει τη διαδικασία μέτρησης και αποστολής δεδομένων που πραγματοποιείται από το σύστημα με χρήση του αισθητήρα AS7261. Αρχικά, το σύστημα ξεκινά (Εναρξη) και προετοιμάζεται για την επικοινωνία με τον αισθητήρα. Το πρώτο βήμα είναι το άνοιγμα του καναλιού FTDI, όπου ελέγχεται αν υπάρχουν διαθέσιμα κανάλια και αν είναι διαθέσιμο το πρώτο κανάλι, το οποίο χρησιμοποιείται για την επικοινωνία. Στη συνέχεια, γίνεται η αρχικοποίηση της επικοινωνίας μέσω του πρωτοκόλλου I²C, το οποίο συνδέει τον αισθητήρα με το σύστημα.

Μόλις η επικοινωνία έχει εγκατασταθεί, δημιουργείται ένας μοναδικός αριθμός ομάδας βάσει της τρέχουσας χρονικής σφραγίδας (timestamp), ο οποίος θα χρησιμοποιηθεί για την κατηγοριοποίηση των δεδομένων που θα συλλεχθούν κατά τις μετρήσεις. Το σύστημα εισέρχεται σε έναν βρόχο που εκτελείται κάθε 10 δευτερόλεπτα. Σε κάθε κύκλο του βρόχου, το LED του αισθητήρα ενεργοποιείται για να φωτίσει το δείγμα, επιτρέποντας την ανίχνευση των χρωματικών συντεταγμένων XYZ και των τιμών NIR (κοντά στο υπέρυθρο φάσμα). Αυτά τα δεδομένα στη συνέχεια μετατρέπονται από τον αισθητήρα στις τιμές RGB, οι οποίες είναι πιο κατανοητές για περαιτέρω ανάλυση.

Αφού συλλεχθούν τα δεδομένα, αποστέλλονται σε έναν απομακρυσμένο διακομιστή (API) για αποθήκευση και επεξεργασία. Μόλις ολοκληρωθεί η αποστολή, το LED απενεργοποιείται μέχρι τον επόμενο κύκλο μέτρησης. Μετά τη λήξη του βρόχου, το σύστημα κλείνει τις συνδέσεις του με το κανάλι FTDI και το I²C, ολοκληρώνοντας την διαδικασία.

4.1.4 Ανάλυση μέσω κώδικα

Βασικές Περιλήψεις και Εισαγωγές Βιβλιοθηκών

```
#include <windows.h> // Βιβλιοθήκη για χειρισμό Windows API
```

```

#include <iostream>           // Βιβλιοθήκη για είσοδο/έξοδο στο terminal (cin/cout)

#undef max                    // Απενεργοποίηση μακροεντολής max για να χρησιμοποιηθεί το std::max
#undef min                    // Απενεργοποίηση μακροεντολής min για να χρησιμοποιηθεί το std::min

#include <algorithm>          // Βιβλιοθήκη για αλγορίθμους όπως std::min και std::max
#include <cmath>              // Βιβλιοθήκη για μαθηματικές συναρτήσεις όπως pow
#include "ftd2xx.h"          // Βιβλιοθήκη για επικοινωνία μέσω FTDI
#include "libmpsse_i2c.h"    // Βιβλιοθήκη για επικοινωνία I2C μέσω MPSSE
#include <stdint.h>           // Βιβλιοθήκη για καθορισμό τύπων δεδομένων όπως uint8_t, uint32_t
#include <string>             // Βιβλιοθήκη για διαχείριση συμβολοσειρών (std::string)
#include <winhttp.h>          // Βιβλιοθήκη για αποστολή HTTP αιτημάτων

#pragma comment(lib, "FTD2XX.LIB") // Συνδέει τη βιβλιοθήκη FTD2XX για FTDI επικοινωνία
#pragma comment(lib, "libmpsse.lib") // Συνδέει τη βιβλιοθήκη MPSSE για I2C επικοινωνία
#pragma comment(lib, "winhttp.lib") // Συνδέει τη βιβλιοθήκη WinHTTP για HTTP αιτήσεις

using namespace std;        // Χρήση του χώρου ονομάτων std για συντόμευση

```

Αυτό το αρχικό τμήμα του κώδικα εισάγει τις απαραίτητες βιβλιοθήκες που χρησιμοποιούνται για την επικοινωνία με τον αισθητήρα AS7261 και την αποστολή δεδομένων μέσω HTTP. Εισάγεται η βιβλιοθήκη Windows API (windows.h), η βιβλιοθήκη εισόδου/εξόδου για το terminal (iostream), και βιβλιοθήκες για μαθηματικές συναρτήσεις (cmath) και αλγορίθμους όπως std::max και std::min. Επίσης, περιλαμβάνονται οι βιβλιοθήκες που αφορούν την επικοινωνία FTDI (ftd2xx.h) και I²C (libmpsse_i2c.h), καθώς και η βιβλιοθήκη winhttp.h για την αποστολή αιτημάτων HTTP. Οι εντολές #pragma comment συνδέουν τις κατάλληλες βιβλιοθήκες ώστε να χρησιμοποιούνται στην επικοινωνία FTDI, I²C και HTTP. Με την εντολή using namespace std; επιτρέπει τη χρήση αντικειμένων από το χώρο ονομάτων std χωρίς να γράφεται επαναλαμβανόμενα.

Ορισμός Σταθερών και Καταχωρητών

```

// Διεύθυνση I2C της συσκευής AS7261
#define AS7261_I2C_ADDRESS 0x49 // Ορίζουμε τη διεύθυνση I2C του αισθητήρα AS7261

// Καθορισμός των καταχωρητών και των bits για τους Virtual Registers

```

```

#define I2C_AS72XX_SLAVE_STATUS_REG 0x00 // Καταχωρητής Status για την παρακολούθηση της κατάστασης της
συσκευής
#define I2C_AS72XX_SLAVE_WRITE_REG 0x01 // Καταχωρητής εγγραφής για αποστολή δεδομένων στον αισθητήρα
#define I2C_AS72XX_SLAVE_READ_REG 0x02 // Καταχωρητής ανάγνωσης για λήψη δεδομένων από τον αισθητήρα
#define I2C_AS72XX_SLAVE_TX_VALID 0x02 // Bit που δείχνει εάν η συσκευή είναι έτοιμη για αποστολή δεδομένων
(TX ready)
#define I2C_AS72XX_SLAVE_RX_VALID 0x01 // Bit που δείχνει εάν η συσκευή έχει δεδομένα έτοιμα για ανάγνωση
(RX ready)

```

Αυτό το τμήμα του κώδικα καθορίζει τη διεύθυνση I²C του αισθητήρα AS7261 και ορίζει διάφορους καταχωρητές που χρησιμοποιούνται για την επικοινωνία με τον αισθητήρα μέσω I²C. Η διεύθυνση I²C της συσκευής ορίζεται ως 0x49. Οι καταχωρητές I2C_AS72XX_SLAVE_STATUS_REG, I2C_AS72XX_SLAVE_WRITE_REG, και I2C_AS72XX_SLAVE_READ_REG χρησιμοποιούνται για τον έλεγχο της κατάστασης της συσκευής, την εγγραφή δεδομένων προς τον αισθητήρα και την ανάγνωση δεδομένων από τον αισθητήρα αντίστοιχα. Τα bits I2C_AS72XX_SLAVE_TX_VALID και I2C_AS72XX_SLAVE_RX_VALID δείχνουν εάν η συσκευή είναι έτοιμη για αποστολή ή αν υπάρχουν διαθέσιμα δεδομένα για ανάγνωση. Αυτές οι σταθερές χρησιμοποιούνται στις συναρτήσεις I²C για την αποστολή και λήψη δεδομένων, εξασφαλίζοντας ότι η επικοινωνία γίνεται σωστά.

Συνάρτηση Μετατροπής XYZ σε RGB

```

// Συνάρτηση μετατροπής XYZ σε RGB χρησιμοποιώντας το sRGB χρωματικό πρότυπο και το D65 φωτιστικό
void XYZtoRGB(float X, float Y, float Z, float& R, float& G, float& B) {

    // Κανονικοποίηση των τιμών XYZ βάσει του D65 φωτιστικού
    X /= 95.047f; // Τιμή αναφοράς για το X στο D65 φωτιστικό
    Y /= 100.000f; // Τιμή αναφοράς για το Y στο D65 φωτιστικό
    Z /= 108.883f; // Τιμή αναφοράς για το Z στο D65 φωτιστικό

    // Μετατροπή μήτρας από XYZ σε γραμμικό RGB (D65 φωτιστικό)
    float R_linear = 3.2406f * X - 1.5372f * Y - 0.4986f * Z; // Υπολογισμός γραμμικής τιμής R
    float G_linear = -0.9689f * X + 1.8758f * Y + 0.0415f * Z; // Υπολογισμός γραμμικής τιμής G
    float B_linear = 0.0557f * X - 0.2040f * Y + 1.0570f * Z; // Υπολογισμός γραμμικής τιμής B

```

```

// Εφαρμογή διόρθωσης gamma
auto gammaCorrection = [](float C) -> float {
    return (C <= 0.0031308f) ? (12.92f * C) : (1.055f * pow(C, 1.0f / 2.4f) - 0.055f);
};

R = gammaCorrection(R_linear); // Διόρθωση gamma για το R
G = gammaCorrection(G_linear); // Διόρθωση gamma για το G
B = gammaCorrection(B_linear); // Διόρθωση gamma για το B

// Περιορισμός των τιμών στο εύρος [0, 1]
R = std::max(0.0f, std::min(1.0f, R)); // Έλεγχος ότι η τιμή του R είναι μεταξύ 0 και 1
G = std::max(0.0f, std::min(1.0f, G)); // Έλεγχος ότι η τιμή του G είναι μεταξύ 0 και 1
B = std::max(0.0f, std::min(1.0f, B)); // Έλεγχος ότι η τιμή του B είναι μεταξύ 0 και 1
}

```

Αυτή η συνάρτηση μετατρέπει τις τιμές χρώματος XYZ σε τιμές RGB χρησιμοποιώντας το sRGB χρωματικό πρότυπο και το D65 φωτιστικό. Η κανονικοποίηση των τιμών XYZ γίνεται πρώτα με βάση τις τιμές αναφοράς D65. Έπειτα, εφαρμόζεται μια μήτρα μετατροπής για την απόδοση γραμμικών τιμών RGB. Στη συνέχεια, εφαρμόζεται διόρθωση gamma για τη ρύθμιση των χρωματικών τιμών έτσι ώστε να είναι κατάλληλες για απεικόνιση σε οθόνες. Τέλος, οι τιμές περιορίζονται στο εύρος [0, 1] ώστε να διασφαλιστεί ότι είναι έγκυρες για το χρωματικό πρότυπο RGB. Αυτή η διαδικασία είναι απαραίτητη για τη μετατροπή των δεδομένων που λαμβάνονται από τον αισθητήρα (σε μορφή XYZ) σε μορφή κατάλληλη για χρήση σε συστήματα που λειτουργούν με RGB.

Αποστολή Δεδομένων στο API

```

void sendRGBtoAPI(int R, int G, int B, uint8_t NIR_LOW, uint8_t NIR_HIGH, unsigned long num) {
    // Δημιουργία του URL αιτήματος με παραμέτρους τα R, G, B, NIR και τον αριθμό ομάδας
    std::wstring query = L"/rgb?R=" + std::to_wstring(R) + L"&G=" + std::to_wstring(G) + L"&B=" + std::to_wstring(B) +

```

```

        L"&group=G" + std::to_wstring(num) + L"&nirl=" + std::to_wstring(NIR_LOW) + L"&nirh=" +
std::to_wstring(NIR_HIGH);

// Ανοίγουμε μια συνεδρία HTTP για την αποστολή του αιτήματος
HINTERNET hSession = WinHttpOpen(L"A WinHTTP Example Program/1.0",
    WINHTTP_ACCESS_TYPE_DEFAULT_PROXY,
    WINHTTP_NO_PROXY_NAME,
    WINHTTP_NO_PROXY_BYPASS, 0);

if (hSession) {
    // Καθορισμός του διακομιστή στόχου (localhost στη θύρα 2323)
    HINTERNET hConnect = WinHttpConnect(hSession, L"localhost", 2323, 0);

    if (hConnect) {
        // Δημιουργία HTTP αιτήματος GET
        HINTERNET hRequest = WinHttpOpenRequest(hConnect, L"GET", query.c_str(),
            NULL, WINHTTP_NO_REFERER, WINHTTP_DEFAULT_ACCEPT_TYPES, 0);

        if (hRequest) {
            // Αποστολή του αιτήματος GET στον διακομιστή
            BOOL result = WinHttpSendRequest(hRequest, WINHTTP_NO_ADDITIONAL_HEADERS, 0,
WINHTTP_NO_REQUEST_DATA, 0, 0, 0);

            if (result) {
                // Λήψη της απόκρισης από τον διακομιστή
                WinHttpReceiveResponse(hRequest, NULL);
            }

            // Κλείσιμο του αιτήματος HTTP
            WinHttpCloseHandle(hRequest);
        }

        // Κλείσιμο της σύνδεσης HTTP
    }
}

```

```

        WinHttpCloseHandle(hConnect);
    }

    // Κλείσιμο της συνεδρίας HTTP
    WinHttpCloseHandle(hSession);
}

// Εκτύπωση μηνύματος ότι ολοκληρώθηκε το αίτημα HTTP
cout << "HTTP request end" << endl;
}

```

Αυτή η συνάρτηση δημιουργεί ένα HTTP GET αίτημα και αποστέλλει τις τιμές RGB και NIR που λαμβάνονται από τον αισθητήρα σε έναν τοπικό διακομιστή (localhost) που λειτουργεί στη θύρα 2323. Οι τιμές RGB και NIR, μαζί με τον αριθμό της ομάδας, ενσωματώνονται στη διεύθυνση URL ως παράμετροι. Αρχικά, ανοίγεται μια HTTP συνεδρία με τη χρήση της βιβλιοθήκης WinHTTP. Έπειτα, δημιουργείται μια σύνδεση με τον τοπικό διακομιστή και κατασκευάζεται ένα αίτημα GET που περιέχει τις παραμέτρους των δεδομένων. Το αίτημα αποστέλλεται στον διακομιστή, και αν η αποστολή είναι επιτυχής, λαμβάνεται η απόκριση από τον διακομιστή. Τέλος, η συνεδρία και οι συνδέσεις HTTP κλείνουν. Αυτή η συνάρτηση επιτρέπει στο σύστημα να αποστέλλει τα δεδομένα του αισθητήρα σε απομακρυσμένους διακομιστές για περαιτέρω ανάλυση ή αποθήκευση.

Αρχικοποίηση του I²C Καναλιού

```

// Συνάρτηση για αρχικοποίηση της MPSSE σε I2C mode
FT_STATUS i2c_init(FT_HANDLE* handle) {
    ChannelConfig config; // Δημιουργία δομής ρυθμίσεων καναλιού
    config.ClockRate = I2C_CLOCK_STANDARD_MODE; // Ορισμός ταχύτητας ρολογιού στα 100 kHz (Standard Mode)
    config.LatencyTimer = 2; // Ρύθμιση του latency timer
    config.Options = I2C_DISABLE_3PHASE_CLOCKING; // Ρύθμιση επιλογών για την απενεργοποίηση του 3-phase
    clocking
    config.currentPinState = 0; // Καθορισμός της κατάστασης των pins

    // Αρχικοποίηση του καναλιού I2C με τις παραπάνω ρυθμίσεις
    FT_STATUS status = I2C_InitChannel(*handle, &config);
    if (status != FT_OK) { // Αν η αρχικοποίηση αποτύχει, εμφάνιση μηνύματος λάθους

```

```

    cerr << "Failed to initialize I2C channel." << endl;

    return status;
}

return FT_OK; // Επιστρέφει επιτυχία αν η αρχικοποίηση είναι επιτυχής
}

```

Αυτή η συνάρτηση πραγματοποιεί την αρχικοποίηση της επικοινωνίας μέσω I²C χρησιμοποιώντας το MPSSE πρωτόκολλο (Multi-Protocol Synchronous Serial Engine). Ορίζονται οι ρυθμίσεις του καναλιού, όπως η ταχύτητα του ρολογιού στα 100 kHz (Standard Mode), η ρύθμιση του latency timer, και οι επιλογές για το 3-phase clocking (το οποίο απενεργοποιείται). Αφού καθοριστούν οι παράμετροι, γίνεται η αρχικοποίηση του καναλιού I²C με τις ρυθμίσεις αυτές. Αν η διαδικασία αποτύχει, εμφανίζεται ένα μήνυμα λάθους. Εάν η αρχικοποίηση είναι επιτυχής, επιστρέφεται μια τιμή επιτυχίας FT_OK. Αυτή η συνάρτηση είναι απαραίτητη για να ξεκινήσει η επικοινωνία με τον αισθητήρα μέσω του I²C πρωτοκόλλου.

Εγγραφή στον Virtual Register του Αισθητήρα

```

// Συνάρτηση για εγγραφή στον virtual register του AS7261
FT_STATUS i2c_write_virtual_register(FT_HANDLE handle, uint8_t regAddress, uint8_t data) {
    UCHAR statusBuffer[1]; // Buffer για την ανάγνωση της κατάστασης
    UCHAR writeBuffer[2]; // Buffer για την εγγραφή της διεύθυνσης καταχωρητή και των δεδομένων
    DWORD bytesTransferred = 0; // Αριθμός bytes που έχουν μεταφερθεί

    // Πρώτα ελέγχουμε το TX bit για να βεβαιωθούμε ότι είναι έτοιμο για εγγραφή
    do {
        FT_STATUS status = I2C_DeviceRead(handle, AS7261_I2C_ADDRESS, 1, statusBuffer, &bytesTransferred,
        I2C_TRANSFER_OPTIONS_START_BIT | I2C_TRANSFER_OPTIONS_STOP_BIT);

        if (status != FT_OK || bytesTransferred != 1) { // Αν υπάρχει σφάλμα, επιστρέφεται 0xFF
            cerr << "Error reading status register." << endl;

            return 0xFF;
        }
    } while (statusBuffer[0] & I2C_AS72XX_SLAVE_TX_VALID); // Αναμονή μέχρι να είναι έτοιμο για εγγραφή

    // Εγγραφή της διεύθυνσης του καταχωρητή στο Write Register

```

```

writeBuffer[0] = I2C_AS72XX_SLAVE_WRITE_REG; // Καθορίζεται ότι γράφουμε στον Write Register

writeBuffer[1] = regAddress | 0x80; // Η διεύθυνση του virtual register

// Στέλνουμε την εντολή εγγραφής στον I2C bus

FT_STATUS status = I2C_DeviceWrite(handle, AS7261_I2C_ADDRESS, 2, writeBuffer, &bytesTransferred,
I2C_TRANSFER_OPTIONS_START_BIT | I2C_TRANSFER_OPTIONS_STOP_BIT);

if (status != FT_OK || bytesTransferred != 2) { // Έλεγχος αν η εγγραφή ήταν επιτυχής

    cerr << "Error writing to the Write Register." << endl;

    return status;

}

// Αναμονή για μικρό χρονικό διάστημα

Sleep(100);

// Εγγραφή των δεδομένων στον καταχωρητή

writeBuffer[0] = I2C_AS72XX_SLAVE_WRITE_REG; // Ξανά καθορισμός για εγγραφή στον Write Register

writeBuffer[1] = data; // Τα δεδομένα που θα γραφτούν στον καταχωρητή

status = I2C_DeviceWrite(handle, AS7261_I2C_ADDRESS, 2, writeBuffer, &bytesTransferred,
I2C_TRANSFER_OPTIONS_START_BIT | I2C_TRANSFER_OPTIONS_STOP_BIT);

if (status != FT_OK || bytesTransferred != 2) { // Έλεγχος για επιτυχία εγγραφής

    cerr << "Error writing to the Write Register." << endl;

    return status;

}

return FT_OK; // Επιστροφή επιτυχίας αν όλα πήγαν καλά
}

```

Αυτή η συνάρτηση επιτρέπει την εγγραφή δεδομένων στον virtual register του αισθητήρα AS7261 μέσω του I2C πρωτοκόλλου. Η διαδικασία περιλαμβάνει τον έλεγχο του TX bit για να διασφαλιστεί ότι η συσκευή είναι έτοιμη για αποστολή δεδομένων. Αφού βεβαιωθεί ότι το TX bit είναι έτοιμο, γίνεται εγγραφή της διεύθυνσης του καταχωρητή στον Write Register. Μετά από μικρή καθυστέρηση, αποστέλλονται τα δεδομένα που πρέπει να γραφτούν στον καταχωρητή. Αν υπάρξει οποιοδήποτε

σφάλμα κατά την εγγραφή, επιστρέφεται κωδικός σφάλματος. Σε περίπτωση επιτυχίας, επιστρέφεται η τιμή FT_OK. Αυτή η συνάρτηση είναι απαραίτητη για να στέλνουμε εντολές και δεδομένα στον αισθητήρα.

Ανάγνωση από τον Virtual Register του Αισθητήρα

```
// Συνάρτηση για ανάγνωση από τον virtual register του AS7261
uint8_t i2c_read_virtual_register(FT_HANDLE handle, uint8_t regAddress) {
    UCHAR statusBuffer[1]; // Buffer για την ανάγνωση της κατάστασης
    UCHAR readBuffer[2]; // Buffer για την ανάγνωση των δεδομένων
    UCHAR writeBuffer[2]; // Buffer για την αποστολή της εντολής εγγραφής
    DWORD bytesTransferred = 0; // Αριθμός bytes που έχουν μεταφερθεί

    // Πρώτα ελέγχουμε το TX bit για να βεβαιωθούμε ότι είναι έτοιμο για ανάγνωση
    do {
        FT_STATUS status = I2C_DeviceRead(handle, AS7261_I2C_ADDRESS, 1, statusBuffer, &bytesTransferred,
        I2C_TRANSFER_OPTIONS_START_BIT | I2C_TRANSFER_OPTIONS_STOP_BIT);

        if (status != FT_OK || bytesTransferred != 1) { //Έλεγχος για σφάλμα κατά την ανάγνωση του status
            cerr << "Error reading status register." << endl;
            return 0xFF; // Αν υπάρχει σφάλμα, επιστρέφεται 0xFF
        }
    } while (statusBuffer[0] & I2C_AS72XX_SLAVE_TX_VALID); // Αναμονή μέχρι να είναι έτοιμο για ανάγνωση

    // Καταχωρούμε τη διεύθυνση του virtual register στο Write Register
    writeBuffer[0] = I2C_AS72XX_SLAVE_WRITE_REG; // Ορισμός ότι γράφουμε στον Write Register
    writeBuffer[1] = regAddress; // Η διεύθυνση του virtual register που θέλουμε να διαβάσουμε

    // Στέλνουμε την εντολή στον I2C bus
    FT_STATUS status = I2C_DeviceWrite(handle, AS7261_I2C_ADDRESS, 2, writeBuffer, &bytesTransferred,
    I2C_TRANSFER_OPTIONS_START_BIT | I2C_TRANSFER_OPTIONS_STOP_BIT);

    if (status != FT_OK || bytesTransferred != 2) { //Έλεγχος αν η εντολή στάλθηκε σωστά
        cerr << "Error writing to the Write Register." << endl;
        return status;
    }
}
```

```

Sleep(100); // Αναμονή για την ολοκλήρωση της διαδικασίας

// Περιμένουμε να οριστεί το RX bit που υποδεικνύει ότι υπάρχουν δεδομένα για ανάγνωση
readBuffer[0] = I2C_AS72XX_SLAVE_READ_REG; // Ορισμός ότι διαβάζουμε από τον Read Register
do {
    status = I2C_DeviceRead(handle, AS7261_I2C_ADDRESS, 2, readBuffer, &bytesTransferred,
I2C_TRANSFER_OPTIONS_START_BIT | I2C_TRANSFER_OPTIONS_STOP_BIT);

    if (status != FT_OK || bytesTransferred != 2) { // Έλεγχος για σφάλμα κατά την ανάγνωση των δεδομένων

        cerr << "Error reading status register." << endl;

        return 0xFF;

    }

    Sleep(10); // Αναμονή 10 ms πριν τον επόμενο

```

Αυτή η συνάρτηση αναλαμβάνει την ανάγνωση δεδομένων από τον virtual register του αισθητήρα AS7261 μέσω της I²C επικοινωνίας. Πρώτα, ελέγχεται αν το TX bit είναι έτοιμο για την εκκίνηση της διαδικασίας ανάγνωσης. Εάν η συσκευή είναι έτοιμη, στέλνεται η διεύθυνση του register που θέλουμε να διαβάσουμε στον Write Register του αισθητήρα. Μετά από μικρή αναμονή για την ολοκλήρωση της διαδικασίας, ξεκινά η διαδικασία ανάγνωσης. Ο αισθητήρας απαντά με τα δεδομένα από τον συγκεκριμένο register. Η διαδικασία ολοκληρώνεται όταν το RX bit υποδείξει ότι τα δεδομένα είναι έτοιμα για ανάγνωση. Ανάλογα με το αν η διαδικασία ήταν επιτυχής, επιστρέφονται τα δεδομένα ή ένας κωδικός σφάλματος.

Μετατροπή Bytes σε Float (IEEE 754)

```

// Συνάρτηση για μετατροπή 4 bytes σε float (IEEE 754)
float convert_bytes_to_float(uint8_t byte1, uint8_t byte2, uint8_t byte3, uint8_t byte4) {

    union {

        uint8_t bytes[4]; // 4 bytes που αντιπροσωπεύουν τα δεδομένα

        float value; // Η αντίστοιχη τιμή σε floating point

    } u;

    u.bytes[0] = byte4; // Πιο σημαντικό byte τελευταίο (Big Endian)

    u.bytes[1] = byte3; // Δεύτερο πιο σημαντικό byte

```

```

u.bytes[2] = byte2; // Τρίτο πιο σημαντικό byte

u.bytes[3] = byte1; // Λιγότερο σημαντικό byte πρώτο

return u.value; // Επιστρέφουμε την τιμή σε float
}

```

Αυτή η συνάρτηση μετατρέπει 4 bytes σε μια τιμή τύπου float σύμφωνα με το πρότυπο IEEE 754. Χρησιμοποιεί μια ένωση (union) για να αντιπροσωπεύσει τόσο τα 4 bytes όσο και την τιμή σε μορφή floating point. Η ένωση επιτρέπει την αποθήκευση των bytes στην κατάλληλη σειρά (Big Endian), ώστε να υπολογιστεί σωστά η τιμή. Τα bytes αποθηκεύονται στο αντίστοιχο πεδίο bytes της ένωσης, και στη συνέχεια η συνάρτηση επιστρέφει την αντίστοιχη floating point τιμή που προκύπτει από τη μετατροπή αυτών των bytes. Αυτή η λειτουργία είναι σημαντική για τη μετατροπή των τιμών που λαμβάνονται από τον αισθητήρα AS7261 σε floating point δεδομένα.

Ανάγνωση και Μετατροπή Καλιμπραρισμένων Δεδομένων

```

void read_calibrated_data(FT_HANDLE ftHandle, unsigned long num) {

    // Ανάγνωση των καλιμπραρισμένων τιμών X (0x14 έως 0x17)

    uint8_t Cal_X_BYTE1 = i2c_read_virtual_register(ftHandle, 0x14);
    uint8_t Cal_X_BYTE2 = i2c_read_virtual_register(ftHandle, 0x15);
    uint8_t Cal_X_BYTE3 = i2c_read_virtual_register(ftHandle, 0x16);
    uint8_t Cal_X_BYTE4 = i2c_read_virtual_register(ftHandle, 0x17);

    float Cal_X = convert_bytes_to_float(Cal_X_BYTE1, Cal_X_BYTE2, Cal_X_BYTE3, Cal_X_BYTE4);

    cout << "Cal_X = " << Cal_X << endl;

    // Ανάγνωση των καλιμπραρισμένων τιμών Y (0x18 έως 0x1B)

    uint8_t Cal_Y_BYTE1 = i2c_read_virtual_register(ftHandle, 0x18);
    uint8_t Cal_Y_BYTE2 = i2c_read_virtual_register(ftHandle, 0x19);
    uint8_t Cal_Y_BYTE3 = i2c_read_virtual_register(ftHandle, 0x1A);
    uint8_t Cal_Y_BYTE4 = i2c_read_virtual_register(ftHandle, 0x1B);

    float Cal_Y = convert_bytes_to_float(Cal_Y_BYTE1, Cal_Y_BYTE2, Cal_Y_BYTE3, Cal_Y_BYTE4);

    cout << "Cal_Y = " << Cal_Y << endl;
}

```

```

// Ανάγνωση των καλιμπραρισμένων τιμών Z (0x1C έως 0x1F)
uint8_t Cal_Z_BYTE1 = i2c_read_virtual_register(ftHandle, 0x1C);
uint8_t Cal_Z_BYTE2 = i2c_read_virtual_register(ftHandle, 0x1D);
uint8_t Cal_Z_BYTE3 = i2c_read_virtual_register(ftHandle, 0x1E);
uint8_t Cal_Z_BYTE4 = i2c_read_virtual_register(ftHandle, 0x1F);

float Cal_Z = convert_bytes_to_float(Cal_Z_BYTE1, Cal_Z_BYTE2, Cal_Z_BYTE3, Cal_Z_BYTE4);

cout << "Cal_Z = " << Cal_Z << endl;

// Ανάγνωση των τιμών NIR (0x0E έως 0x0F)
uint8_t NIR_HIGH = i2c_read_virtual_register(ftHandle, 0x0E);

uint8_t NIR_LOW = i2c_read_virtual_register(ftHandle, 0x0F);

cout << "NIR_HIGH ===== : 0x" << hex << (int)NIR_HIGH << endl;
cout << "NIR_LOW ===== : 0x" << hex << (int)NIR_LOW << endl;

// Μετατροπή των καλιμπραρισμένων τιμών XYZ σε RGB
float R, G, B;
XYZtoRGB(Cal_X, Cal_Y, Cal_Z, R, G, B);

// Μετατροπή των τιμών RGB σε κλίμακα [0, 255] για αναπαράσταση 8-bit
int R_8bit = static_cast<int>(R * 255.0f);
int G_8bit = static_cast<int>(G * 255.0f);
int B_8bit = static_cast<int>(B * 255.0f);

// Εκτύπωση των τιμών RGB
std::cout << "RGB values: R = " << R_8bit << ", G = " << G_8bit << ", B = " << B_8bit << std::endl;

// Αποστολή των τιμών RGB και NIR στο API
sendRGBtoAPI(R_8bit, G_8bit, B_8bit, (int)NIR_LOW, (int)NIR_HIGH, num);
}

```

Αυτή η συνάρτηση είναι υπεύθυνη για την ανάγνωση και μετατροπή των καλιμπραρισμένων δεδομένων από τον αισθητήρα AS7261. Πρώτα, διαβάζονται οι καλιμπραρισμένες τιμές για το X, Y

και Z από τους κατάλληλους virtual registers μέσω I²C επικοινωνίας. Αυτές οι τιμές στη συνέχεια μετατρέπονται από τα raw δεδομένα (bytes) σε floating-point τιμές τύπου float με τη βοήθεια της συνάρτησης `convert_bytes_to_float`.

Μετά την ανάγνωση των τιμών XYZ, διαβάζονται επίσης οι τιμές NIR (κοντά στο υπέρυθρο φάσμα) από τους virtual registers του αισθητήρα.

Στη συνέχεια, η συνάρτηση καλεί τη `XYZtoRGB`, η οποία μετατρέπει τις καλιμπραρισμένες τιμές XYZ σε τιμές RGB. Οι τιμές RGB στη συνέχεια προσαρμόζονται ώστε να βρίσκονται στο εύρος 8-bit (0-255), το οποίο είναι κατάλληλο για απεικόνιση χρώματος σε συστήματα RGB. Τέλος, οι τιμές RGB και NIR αποστέλλονται σε έναν απομακρυσμένο διακομιστή μέσω της συνάρτησης `sendRGBtoAPI`.

Εκτέλεση Λειτουργίας LED και Ανάγνωση Δεδομένων

```
// Συνάρτηση που ενεργοποιεί/απενεργοποιεί το LED και διαβάζει τα δεδομένα
void performLEDOperation(FT_HANDLE ftHandle, unsigned long num) {
    // Ενεργοποίηση του LED
    i2c_write_virtual_register(ftHandle, 0x07, 0x08); // Εντολή για ενεργοποίηση του LED
    std::cout << "LED ON" << std::endl;

    // Ανάγνωση καλιμπραρισμένων δεδομένων
    read_calibrated_data(ftHandle, num);

    // Απενεργοποίηση του LED
    i2c_write_virtual_register(ftHandle, 0x07, 0x00); // Εντολή για απενεργοποίηση του LED
    std::cout << "LED OFF" << std::endl;
}
```

Η συνάρτηση `performLEDOperation` ελέγχει την ενεργοποίηση και απενεργοποίηση του LED στον αισθητήρα AS7261, καθώς και την ανάγνωση των δεδομένων κατά τη διάρκεια που το LED είναι ενεργό. Αρχικά, στέλνεται μια εντολή στον αισθητήρα μέσω I²C για την ενεργοποίηση του LED (μέσω του register 0x07). Όταν το LED ενεργοποιηθεί, η συνάρτηση καλεί τη `read_calibrated_data` για να διαβάσει τις τιμές XYZ και NIR από τον αισθητήρα. Αυτές οι τιμές μετατρέπονται και αποστέλλονται στο API. Μόλις ολοκληρωθεί η ανάγνωση, η συνάρτηση απενεργοποιεί το LED αποστέλλοντας την αντίστοιχη εντολή στον αισθητήρα. Με αυτόν τον τρόπο, εξασφαλίζεται ότι το LED παραμένει ενεργό μόνο για τη διάρκεια των μετρήσεων.

Δημιουργία Αριθμού Ομάδας από Timestamp

```
// Συνάρτηση που επιστρέφει έναν αριθμό βασισμένο στο timestamp
unsigned long generateNumberFromTimestamp() {
    // Παίρνουμε το τρέχον timestamp σε δευτερόλεπτα από το 1970 (Epoch time)
    std::time_t currentTime = std::time(nullptr);

    // Μετατρέπουμε το timestamp σε έναν unsigned long αριθμό
    unsigned long generatedNumber = static_cast<unsigned long>(currentTime);

    return generatedNumber; // Επιστρέφουμε τον αριθμό που δημιουργήθηκε από το timestamp
}
```

Αυτή η συνάρτηση δημιουργεί έναν μοναδικό αριθμό ομάδας με βάση την τρέχουσα χρονική στιγμή (timestamp) από την εποχή Unix (Epoch time). Χρησιμοποιεί τη συνάρτηση `std::time(nullptr)` για να πάρει το τρέχον timestamp σε δευτερόλεπτα από το 1970. Στη συνέχεια, το timestamp μετατρέπεται σε έναν unsigned long αριθμό, ο οποίος επιστρέφεται από τη συνάρτηση. Αυτός ο αριθμός μπορεί να χρησιμοποιηθεί για την ταυτοποίηση διαφορετικών μετρήσεων ή sessions.

Κύρια Συνάρτηση (Main)

```
int main() {
    FT_HANDLE ftHandle; // Δημιουργία ενός handle για το κανάλι FTDI
    FT_STATUS ftStatus; // Μεταβλητή για αποθήκευση του status των λειτουργιών FTDI
    DWORD channels = 0; // Μεταβλητή για την αποθήκευση του αριθμού των διαθέσιμων καναλιών

    cout << "....." << endl;

    // Άνοιγμα του FTDI καναλιού
    ftStatus = I2C_GetNumChannels(&channels); // Έλεγχος για τον αριθμό των διαθέσιμων FTDI καναλιών
    if (ftStatus != FT_OK || channels == 0) { // Έλεγχος αν δεν υπάρχουν διαθέσιμα κανάλια
        cerr << "No FTDI channels found." << endl;
    }
}
```

```

return 1; // Αν δεν βρεθούν κανάλια, το πρόγραμμα τερματίζει
}

// Άνοιγμα του πρώτου διαθέσιμου καναλιού
ftStatus = I2C_OpenChannel(0, &ftHandle); // Άνοιγμα του καναλιού 0
if (ftStatus != FT_OK) { // Έλεγχος για σφάλμα κατά το άνοιγμα του καναλιού
    cerr << "Error opening I2C channel." << endl;
    return 1; // Αν υπάρχει σφάλμα, το πρόγραμμα τερματίζει
}

// Αρχικοποίηση του I2C καναλιού
if (i2c_init(&ftHandle) != FT_OK) { // Αν η αρχικοποίηση αποτύχει
    cerr << "Error initializing I2C." << endl;
    return 1; // Αν υπάρχει σφάλμα, το πρόγραμμα τερματίζει
}

// Παράγουμε και εμφανίζουμε τον αριθμό που βασίζεται στο timestamp
unsigned long number = generateNumberFromTimestamp();
std::cout << "Group: " << number << std::endl;

// Επαναληπτικός βρόχος κάθε 10 δευτερόλεπτα
while (true) {
    // Εκτέλεση των λειτουργιών LED και ανάγνωσης
    performLEDOperation(ftHandle, number);

    // Αναμονή για 10 δευτερόλεπτα
    Sleep(10000); // 10000 χιλιοστά του δευτερολέπτου = 10 δευτερόλεπτα
}

// Κλείσιμο της FTDI συσκευής
I2C_CloseChannel(ftHandle); // Κλείσιμο του καναλιού FTDI
return 0; // Επιστροφή επιτυχούς εκτέλεσης

```

```
}
```

Η κύρια συνάρτηση (main) ελέγχει τη συνολική ροή του προγράμματος. Αρχικά, ελέγχει αν υπάρχουν διαθέσιμα FTDI κανάλια μέσω της συνάρτησης I2C_GetNumChannels. Αν δεν βρεθεί κανένα κανάλι διαθέσιμο, το πρόγραμμα τερματίζεται με μήνυμα σφάλματος. Αν υπάρχει διαθέσιμο κανάλι, το πρόγραμμα προχωράει στο άνοιγμα του πρώτου διαθέσιμου καναλιού μέσω της συνάρτησης I2C_OpenChannel. Αν το άνοιγμα του καναλιού είναι επιτυχές, αρχικοποιείται το I2C κανάλι με την i2c_init.

Αφού ολοκληρωθεί η αρχικοποίηση, το πρόγραμμα δημιουργεί έναν μοναδικό αριθμό ομάδας (group number) με βάση το timestamp, ο οποίος χρησιμοποιείται για να ταυτοποιεί τις μετρήσεις.

Το κύριο πρόγραμμα εισέρχεται σε έναν επαναληπτικό βρόχο που εκτελείται κάθε 10 δευτερόλεπτα. Μέσα σε αυτόν τον βρόχο καλείται η συνάρτηση performLEDOperation, η οποία ενεργοποιεί το LED του αισθητήρα, διαβάζει τα καλιμπραρισμένα δεδομένα, και τα στέλνει στο API. Μετά από κάθε ανάγνωση δεδομένων, το πρόγραμμα περιμένει 10 δευτερόλεπτα πριν επαναλάβει τη διαδικασία.

Όταν ο βρόχος σταματήσει (είτε μέσω εξωτερικής παρέμβασης είτε μέσω τερματισμού του προγράμματος), το FTDI κανάλι κλείνει χρησιμοποιώντας τη συνάρτηση I2C_CloseChannel. Τέλος, το πρόγραμμα επιστρέφει επιτυχώς την τιμή 0, υποδεικνύοντας ότι ολοκληρώθηκε χωρίς σφάλματα.

Από την μεριά του Server

Εισαγωγές Βιβλιοθηκών και Αρχικοποίηση Flask

```
from flask import Flask, request, jsonify # Εισαγωγή Flask για δημιουργία web εφαρμογών και επεξεργασία αιτημάτων
import mysql.connector # Εισαγωγή MySQL connector για σύνδεση με βάση δεδομένων MySQL
from datetime import datetime # Εισαγωγή για να χρησιμοποιούμε timestamps

app = Flask(__name__) # Δημιουργία της Flask εφαρμογής
```

Αρχικοποιούμε τη Flask εφαρμογή, η οποία είναι ένα ελαφρύ framework για τη δημιουργία web εφαρμογών. Εισάγουμε επίσης τη βιβλιοθήκη mysql.connector για να μπορούμε να συνδεόμαστε και να εκτελούμε εντολές στη βάση δεδομένων MySQL, καθώς και τη datetime για να καταγράφουμε την ημερομηνία και ώρα σε κάθε εγγραφή.

Ρυθμίσεις Σύνδεσης MySQL

```

# MySQL configuration

db_config = {

    'user': 'root', # Όνομα χρήστη της βάσης δεδομένων

    'password': '', # Κωδικός πρόσβασης για τον χρήστη

    'host': 'localhost', # Τοπικός server για MySQL

    'database': 'amsdb' # Όνομα της βάσης δεδομένων που θα χρησιμοποιηθεί

}

```

Σε αυτό το τμήμα καθορίζονται οι βασικές παράμετροι σύνδεσης με τη βάση δεδομένων MySQL. Οι παράμετροι περιλαμβάνουν το όνομα χρήστη, τον κωδικό πρόσβασης, τον τοπικό server (localhost), και το όνομα της βάσης δεδομένων που θα χρησιμοποιηθεί (amsdb). Αυτές οι παράμετροι θα χρησιμοποιηθούν αργότερα όταν χρειαστεί να συνδεθούμε στη βάση για την εκτέλεση ερωτημάτων.

Συνάρτηση Εισαγωγής Δεδομένων στη Βάση Δεδομένων

```

# Function to insert data into MySQL

def insert_into_db(R, G, B, nirl, nirh, group):

    try:

        # Create a MySQL connection

        conn = mysql.connector.connect(**db_config) # Δημιουργία σύνδεσης με τη βάση δεδομένων MySQL

        cursor = conn.cursor() # Δημιουργία cursor για εκτέλεση εντολών SQL

        # Insert query

        query = ("INSERT INTO amsvalues (R, G, B, nirl, nirh, `group`, created_at) "

            "VALUES (%s, %s, %s, %s, %s, %s, %s)") # Εντολή SQL για εισαγωγή δεδομένων στην amsvalues

        data = (R, G, B, nirl, nirh, group, datetime.now()) # Προετοιμασία των δεδομένων για εισαγωγή

        # Execute the query and commit

        cursor.execute(query, data) # Εκτέλεση της εντολής SQL με τα δεδομένα

        conn.commit() # Επικύρωση των αλλαγών στη βάση δεδομένων

        cursor.close() # Κλείσιμο του cursor

        conn.close() # Κλείσιμο της σύνδεσης με τη βάση δεδομένων

        return True # Επιστροφή επιτυχίας

```

```
except mysql.connector.Error as err: # Διαχείριση σφαλμάτων κατά την εκτέλεση της εντολής SQL

    print(f"Error: {err}") # Εμφάνιση του σφάλματος

    return False # Επιστροφή αποτυχίας
```

Αυτή η συνάρτηση συνδέεται με τη βάση δεδομένων MySQL και εκτελεί την εισαγωγή των δεδομένων RGB και NIR, καθώς και του αριθμού ομάδας. Αρχικά, γίνεται σύνδεση με τη βάση δεδομένων μέσω της `mysql.connector.connect`, και έπειτα εκτελείται η SQL εντολή `INSERT INTO` για να εισάγει τα δεδομένα. Εάν η εισαγωγή ολοκληρωθεί με επιτυχία, τα δεδομένα επικυρώνονται με την `conn.commit`. Εάν παρουσιαστεί σφάλμα, το πρόγραμμα εκτυπώνει το σφάλμα και επιστρέφει `False`.

Διαδρομή Flask για Λήψη RGB και NIR Δεδομένων

```
# Route to handle GET request for RGB values

#http://localhost:2323/rgb?R=23&G=56&B=80&group=a1

@app.route('/rgb', methods=['GET'])

def receive_rgb():

    # Get the R, G, B values from the request

    R = request.args.get('R') # Ανάγνωση της παραμέτρου R από το URL

    G = request.args.get('G') # Ανάγνωση της παραμέτρου G από το URL

    B = request.args.get('B') # Ανάγνωση της παραμέτρου B από το URL

    group = request.args.get('group', 'default_group') # Ανάγνωση της παραμέτρου group από το URL (προεπιλεγμένο 'default_group')

    nir_l = request.args.get('nir_l') # Ανάγνωση της παραμέτρου nir_l από το URL

    nir_h = request.args.get('nir_h') # Ανάγνωση της παραμέτρου nir_h από το URL

    # Έλεγχος για απουσία τιμών R, G ή B

    if R is None or G is None or B is None:

        return jsonify({"error": "Missing R, G, or B value"}), 400

    # Έλεγχος για απουσία τιμών NIR

    if nir_l is None or nir_h is None:

        return jsonify({"error": "Missing nir value"}), 400
```

```

try:
    # Convert the values to integers

    R = int(R)

    G = int(G)

    B = int(B)

    nirl = int(nirl)

    nirh = int(nirh)

    # Insert into MySQL

    if insert_into_db(R, G, B, nirl, nirh, group):

        return jsonify({"status": "success", "R": R, "G": G, "B": B, "nirl": nirl, "nirh": nirh, "group": group}), 200

    else:

        return jsonify({"status": "error", "message": "Failed to insert into database"}), 500

except ValueError:

    return jsonify({"error": "R, G, and B - nir must be integers"}), 400

```

Αυτή η διαδρομή της Flask εφαρμογής χειρίζεται τα HTTP GET αιτήματα που περιέχουν τιμές RGB και NIR. Οι τιμές διαβάζονται από το URL (παράμετροι R, G, B, nirl, nirh, και group). Αν κάποια από τις τιμές λείπει, επιστρέφεται ένα μήνυμα σφάλματος στον χρήστη.

Οι τιμές μετατρέπονται σε ακέραιους με την εντολή int, και εάν επιτυχώς μετατραπούν, εισάγονται στη βάση δεδομένων μέσω της συνάρτησης insert_into_db. Αν η εισαγωγή ολοκληρωθεί με επιτυχία, επιστρέφεται μια απάντηση επιτυχίας με τις τιμές που εισήχθησαν στη βάση. Αν παρουσιαστεί κάποιο πρόβλημα (π.χ., αν οι τιμές δεν είναι ακέραιοι), επιστρέφεται μήνυμα λάθους.

Εκκίνηση της Flask Εφαρμογής

```

# Main entry point

if __name__ == '__main__':

    # Run the Flask app

    app.run(host='0.0.0.0', port=2323, debug=True)

```

Το τελευταίο μέρος του κώδικα ξεκινάει την Flask εφαρμογή. Χρησιμοποιώντας την εντολή `app.run`, η εφαρμογή τρέχει στον τοπικό `server (localhost)` στην πόρτα `2323`. Το `debug=True` επιτρέπει την εμφάνιση λεπτομερειών για τυχόν σφάλματα κατά την εκτέλεση του κώδικα, κάτι που διευκολύνει την ανάπτυξη και τον εντοπισμό σφαλμάτων. Α

υτός ο κώδικας αποτελεί ένα πλήρες σύστημα που λαμβάνει δεδομένα RGB και NIR μέσω HTTP GET αιτημάτων, τα αποθηκεύει στη βάση δεδομένων MySQL, και επιστρέφει την επιτυχία ή τα σφάλματα στον χρήστη.

Προτεινόμενη ασφάλεια του API

Η ασφάλεια του API είναι πολύ σημαντική για την αποφυγή κακόβουλων επιθέσεων και μη εξουσιοδοτημένης πρόσβασης.

Μια από τις πιο κοινές τεχνικές είναι η χρήση ενός **API Key**. Κάθε πελάτης του API θα πρέπει να χρησιμοποιεί ένα μοναδικό API Key που θα επισυνάπτει στα αιτήματα για να πιστοποιείται η αυθεντικότητά του.

```
API_KEY = "my_secret_key"

@app.route('/rgb', methods=['GET'])
def receive_rgb():
    # Έλεγχος για το API Key στο αίτημα
    api_key = request.args.get('api_key')

    if api_key != API_KEY:
        return jsonify({"error": "Unauthorized"}), 401

# Υπόλοιπος κώδικας...
```

Αν το API Key δεν ταιριάζει, επιστρέφεται 401 Unauthorized.

Calibrated data και White Reference και Black Reference

Στη συσκευή AS7261, τα calibrated data αναφέρονται σε δεδομένα που έχουν υποστεί επεξεργασία ή προσαρμογή έτσι ώστε να αντιστοιχούν σε πραγματικές φυσικές μετρήσεις με βάση γνωστά πρότυπα. Η διαδικασία αυτή εξασφαλίζει ότι οι μετρήσεις που προέρχονται από τον αισθητήρα είναι όσο το δυνατόν πιο ακριβείς και αξιόπιστες.

Πιο συγκεκριμένα, ο AS7261, ο οποίος είναι ένας αισθητήρας φάσματος (spectrometer), μετράει το φως σε διαφορετικά μήκη κύματος. Τα "raw" δεδομένα που παράγει είναι οι ακατέργαστες τιμές που συλλέγει ο αισθητήρας, οι οποίες μπορεί να επηρεαστούν από διάφορους παράγοντες, όπως η θερμοκρασία ή το περιβάλλον μέτρησης. Τα calibrated data, αντίθετα, είναι τα διορθωμένα δεδομένα, στα οποία έχουν ληφθεί υπόψη τέτοιες επιδράσεις, με στόχο να αντιπροσωπεύουν με μεγαλύτερη ακρίβεια το φάσμα του φωτός που μετρήθηκε.

Συνοπτικά, τα calibrated data εξασφαλίζουν ότι οι μετρήσεις του AS7261 είναι έτοιμες για αξιόπιστη ανάλυση και δεν χρειάζονται περαιτέρω διόρθωση ή ρύθμιση από τον χρήστη.

Στον αισθητήρα **AS7261** στο έτοιμο πρόγραμμα που παρέχει η εταιρεία έχει τις λειτουργίες **Save White Reference** και **Save Black Reference** που αφορούν την καλιμπράρισή του για ακριβείς μετρήσεις φάσματος.

Save White Reference (Αναφορά Λευκού)

Η **λευκή αναφορά** χρησιμοποιείται για να καλιμπραριστεί ο αισθητήρας έτσι ώστε να μπορεί να μετρά με ακρίβεια την ανακλαστικότητα ή τη φωτεινότητα ενός δείγματος. Για να γίνει αυτό, ο αισθητήρας εκτελεί μια μέτρηση σε ένα αντικείμενο με γνωστές ιδιότητες, συνήθως ένα λευκό υλικό με πολύ υψηλή ανακλαστικότητα (που αντανακλά σχεδόν όλο το φως). Αυτή η μέτρηση χρησιμοποιείται ως **αναφορά** για να καθορίσει το ανώτατο επίπεδο φωτός που μπορεί να αναγνωρίσει, και με βάση αυτό, ο αισθητήρας μπορεί να υπολογίσει σωστά τη φωτεινότητα άλλων αντικειμένων.

Save Black Reference (Αναφορά Μαύρου)

Η **μαύρη αναφορά** λειτουργεί ως η βάση για τη μέτρηση του χαμηλότερου επιπέδου φωτός ή του σήματος του αισθητήρα όταν δεν υπάρχει καθόλου φως. Συνήθως γίνεται με την κάλυψη του αισθητήρα ή με την έκθεση σε ένα πολύ σκούρο υλικό που απορροφά όλο το φως. Αυτό επιτρέπει στον αισθητήρα να καθορίσει ποιο είναι το ελάχιστο σήμα που θα πρέπει να θεωρείται "μηδέν", κάτι που είναι απαραίτητο για την ακρίβεια των μετρήσεων φάσματος.

- Η **λευκή αναφορά** χρησιμοποιείται για να οριστεί το ανώτερο όριο φωτεινότητας.
- Η **μαύρη αναφορά** καθορίζει το κατώτερο όριο (ή "μηδέν").

Αυτές οι διαδικασίες είναι κρίσιμες για την εξασφάλιση ότι ο αισθητήρας μετρά το φάσμα του φωτός με τη μέγιστη δυνατή ακρίβεια.

4.2 Μετρήσεις

Τα πειράματα που πραγματοποιήσαμε είχαν στόχο να διερευνήσουμε πώς οι μετρήσεις του αισθητήρα AS7261, ο οποίος μετράει τις τιμές RGB και NIR (Κοντινό Υπέρυθρο), μπορούν να χρησιμοποιηθούν για την ποσοτική εκτίμηση της παρουσίας άλγης σε υδατικά δείγματα. Τα πειράματα σχεδιάστηκαν ώστε να παρατηρήσουμε τη μεταβολή των τιμών NIR καθώς μεταβάλλεται η συγκέντρωση της άλγης ή άλλων σωματιδίων στο νερό.

Επειδή εμείς δεν είχαμε στη διάθεση μας Άλγη τοποθετήσαμε σωματίδια από Ρίγανη.

4.2.1 Υλικά και Μέθοδοι

- **Αισθητήρας AS7261:** Ο αισθητήρας αυτός επιλέχθηκε λόγω της ικανότητάς του να μετράει τις τιμές του RGB φάσματος, καθώς και την ένταση του NIR φωτός που περνά μέσα από τα δείγματα.
- **Δείγματα Νερού:** Χρησιμοποιήθηκαν σωληνίσκοι γεμάτοι με καθαρό νερό, καθώς και νερό με προσθήκη πράσινων φυτικών υλικών (όπως ρίγανη) για τη δημιουργία "πράσινου νερού", το οποίο προσομοιώνει την παρουσία άλγης.
- **Σωληνίσκοι:** Το κάθε δείγμα τοποθετήθηκε σε διάφανο σωληνάκι για εύκολη μέτρηση του φωτός που περνά μέσα από αυτό.

4.2.2 Διαδικασία Πειραμάτων

Τα πειράματα χωρίστηκαν σε διαφορετικές φάσεις, με σκοπό τη σύγκριση των μετρήσεων του αισθητήρα ανάμεσα στα δείγματα με διαφορετική συγκέντρωση σωματιδίων:

1. **Καθαρό Νερό (Μόνο Νερό)** Στην πρώτη φάση των πειραμάτων, ο αισθητήρας τοποθετήθηκε ώστε να μετρήσει τις τιμές RGB και NIR ενός σωληνίσκου γεμάτου με καθαρό νερό. Οι τιμές NIR που προέκυψαν ήταν αρκετά υψηλές, καθώς το καθαρό νερό έχει χαμηλή απορρόφηση του κοντινού υπέρυθρου φωτός, με αποτέλεσμα μεγάλο μέρος του φωτός να φτάσει στον δέκτη του αισθητήρα.
2. **Πράσινο Νερό Χωρίς Σωματίδια** Στη δεύτερη φάση, προσθέσαμε πράσινο χρώμα ζαχαροπλαστικής χωρίς σωματίδια. Ο σκοπός ήταν να εξετάσουμε την επίδραση του στις τιμές NIR. Οι μετρήσεις έδειξαν ότι οι τιμές του NIR ήταν μειωμένες σε σχέση με το καθαρό νερό, κάτι που υποδηλώνει ότι το χρώμα απορροφά λίγο το υπέρυθρο φως.
3. **Πράσινο Νερό με Σωματίδια - Αυξανόμενος Όγκος** Στις επόμενες φάσεις, προσθέσαμε στο δείγμα πράσινου νερού και σωματίδια διαφόρων όγκων, ώστε να προσομοιώσουμε την άλγη και την επίδρασή της στις μετρήσεις. Τα σωματίδια προστέθηκαν σε όγκους 0.5 cm^3 , 0.75 cm^3 ,

και 1 cm³ αντίστοιχα. Κάθε φάση μέτρησης κατέγραψε χαμηλότερες τιμές του NIR από την προηγούμενη, κάτι που δείχνει ότι τα σωματίδια απορροφούν και διαχέουν το φως, μειώνοντας την ένταση του υπέρυθρου που φτάνει στον αισθητήρα.

Ο κύλινδρος έχει ακτίνα 0.4 cm και ύψος 3 cm. Ο όγκος του κυλίνδρου υπολογίζεται ως:

$$V_{\text{κύλινδρος}} = \pi r^2 h = \pi \times (0.4)^2 \times 3 \approx 1.507 \text{ cm}^3$$

Τα δείγματα που προσθέσαμε ήταν με σωματίδια όγκου 0.5 cm³, 0.75 cm³ και 1 cm³. Για κάθε δείγμα θα υπολογίσουμε το ποσοστό του όγκου του δείγματος σε σχέση με τον συνολικό όγκο του κυλίνδρου.

Ποσοστό όγκου κάθε δείγματος

- Πράσινο δείγμα με σωματίδια όγκου 0.5 cm³: Ο όγκος του δείγματος είναι $V_{\text{δείγμα}} = 0.5 \text{ cm}^3$. Το ποσοστό του όγκου σε σχέση με τον όγκο του κυλίνδρου είναι:

$$\text{Ποσοστό} = (V_{\text{δείγμα}} / V_{\text{κύλινδρος}}) * 100 = (0.5 / 1.507) * 100 \approx 33.18\%$$

- Πράσινο δείγμα με σωματίδια όγκου 0.75 cm³: Ο όγκος του δείγματος είναι $V_{\text{δείγμα}} = 0.75 \text{ cm}^3$. Το ποσοστό του όγκου σε σχέση με τον όγκο του κυλίνδρου είναι:

$$\text{Ποσοστό} = (V_{\text{δείγμα}} / V_{\text{κύλινδρος}}) * 100 \approx (0.75 / 1.507) * 100 \approx 49.77\%$$

- Πράσινο δείγμα με σωματίδια όγκου 1 cm³: Ο όγκος του δείγματος είναι $V_{\text{δείγμα}} = 1 \text{ cm}^3$. Το ποσοστό του όγκου σε σχέση με τον όγκο του κυλίνδρου είναι:

$$\text{Ποσοστό} = (V_{\text{δείγμα}} / V_{\text{κύλινδρος}}) * 100 \approx (1 / 1.507) * 100 \approx 66.35\%$$

Σύγκριση των ποσοστών όγκου

Από τα παραπάνω, παρατηρούμε ότι τα δείγματα που προσθέσαμε καταλαμβάνουν ένα σημαντικό μέρος του συνολικού όγκου του κυλίνδρου. Το δείγμα με σωματίδια όγκου 0.5 cm³ καταλαμβάνει περίπου το 33% του κυλίνδρου, το δείγμα με 0.75 cm³ περίπου το 50%, ενώ το δείγμα με 1 cm³ καταλαμβάνει περίπου το 66%. Αυτά τα ποσοστά δείχνουν την αύξηση της ποσότητας των σωματιδίων και πώς αυτή επηρεάζει τον συνολικό όγκο του κυλίνδρου.

Η κατανόηση αυτών των ποσοστών είναι σημαντική για την ανάλυση της συμπεριφοράς των δειγμάτων και την επίδραση που έχει ο όγκος των σωματιδίων στην απορρόφηση και αντανάκλαση της ακτινοβολίας NIR. Με την αύξηση του όγκου των σωματιδίων, αναμένουμε ότι οι τιμές NIR θα επηρεαστούν αντίστοιχα, γεγονός που έχει αποτυπωθεί στα δεδομένα μας και στα διαγράμματα.

4.3 Το IoT Portal

Για την οργάνωση των πειραματικών δεδομένων και τη διευκόλυνση της ανάλυσής τους, υλοποιήθηκε ένας server με Python, χρησιμοποιώντας το πλαίσιο Flask. Το σύστημα αυτό υποστηρίζει την αποθήκευση των δεδομένων σε μια βάση δεδομένων, καθώς και την απεικόνιση των μετρήσεων και των πειραμάτων μέσω διαδραστικών γραφημάτων. Η δυνατότητα προβολής των δεδομένων με βάση το group-δείγμα δίνει μια ξεκάθαρη εικόνα της μεταβολής στις τιμές RGB και NIR, επιτρέποντας την αποτελεσματικότερη ανάλυση και σύγκριση.

Περιγραφή Συστήματος

Το σύστημα αποτελείται από τα εξής βασικά τμήματα:

1. **Server Flask:** Ο server βασίζεται σε Flask και λειτουργεί ως το κύριο API για την αποθήκευση των πειραματικών δεδομένων στη βάση και την παροχή αυτών μέσω endpoints.
2. **Βάση Δεδομένων (MySQL):** Όλα τα δεδομένα που σχετίζονται με τα πειράματα αποθηκεύονται σε μια βάση MySQL. Η βάση περιλαμβάνει πίνακες για τα δείγματα και τις μετρήσεις, επιτρέποντας εύκολη πρόσβαση στα δεδομένα ανά group.
3. **Προβολή Γραφημάτων:** Οι μετρήσεις αναλύονται και προβάλλονται σε μορφή γραφημάτων, τα οποία μπορούν να οπτικοποιηθούν από τον χρήστη ανά group-δείγμα μέσω ενός απλού web interface.

Δομή Βάσης Δεδομένων

Η βάση δεδομένων περιέχει τους παρακάτω βασικούς πίνακες:

- **Πίνακας Δειγμάτων (Samples):** Περιλαμβάνει τα χαρακτηριστικά κάθε δείγματος, όπως το όνομα του δείγματος, την ημερομηνία δημιουργίας του και τις πληροφορίες της ομάδας.
- **Πίνακας Μετρήσεων (Measurements):** Αποθηκεύει τις τιμές RGB και NIR για κάθε δείγμα, καθώς και το ID του δείγματος με το οποίο συσχετίζεται η μέτρηση.

Λειτουργίες Flask Server

Ο server έχει σχεδιαστεί ώστε να παρέχει συγκεκριμένες λειτουργίες για τη διαχείριση και την ανάλυση των δεδομένων:

1. **API για Εισαγωγή Δεδομένων** Το API παρέχει endpoints για την εισαγωγή νέων μετρήσεων στη βάση δεδομένων. Τα δεδομένα των πειραμάτων συλλέγονται αυτόματα από τον αισθητήρα AS7261 και αποστέλλονται στον server μέσω αυτών των endpoints. Ο server αναλαμβάνει την αποθήκευση των δεδομένων και την επικύρωση της συνέπειάς τους.
2. **Προβολή Δεδομένων με Γραφήματα** Οι χρήστες μπορούν να προβάλλουν τα δεδομένα μέσω της ιστοσελίδας, η οποία είναι ενσωματωμένη στον Flask server. Τα δεδομένα μπορούν να απεικονιστούν ανά group-δείγμα, επιτρέποντας συγκρίσεις μεταξύ διαφορετικών συνθηκών των πειραμάτων. Για την υλοποίηση αυτής της λειτουργίας, χρησιμοποιείται η βιβλιοθήκη Matplotlib για τη δημιουργία γραφημάτων.
3. **Ζωντανή Απεικόνιση Τελευταίων Μετρήσεων** Η ζωντανή απεικόνιση επιτρέπει την εμφάνιση των πιο πρόσφατων μετρήσεων, δίνοντας στους χρήστες τη δυνατότητα να παρακολουθούν σε πραγματικό χρόνο την εξέλιξη των πειραμάτων. Αυτή η λειτουργία επιτυγχάνεται με χρήση Ajax κλήσεων από το front-end της εφαρμογής, που επικοινωνούν με τον server κάθε λίγα δευτερόλεπτα για να ανακτήσουν νέες μετρήσεις.

Η υλοποίηση του Flask server προσφέρει αρκετά πλεονεκτήματα για την οργάνωση και ανάλυση των πειραματικών δεδομένων:

- **Αποτελεσματική Αποθήκευση Δεδομένων:** Τα δεδομένα αποθηκεύονται οργανωμένα στη βάση, διατηρώντας πληροφορίες για κάθε μέτρηση και δείγμα.
- **Διαδραστικά Γραφήματα:** Η δυνατότητα οπτικοποίησης μέσω διαγραμμάτων επιτρέπει την εύκολη παρατήρηση των διαφορών ανάμεσα στα δείγματα και τις ομάδες.
- **Επεκτασιμότητα:** Η δομή του Flask server επιτρέπει την εύκολη επέκταση για μελλοντικές ανάγκες, όπως η προσθήκη νέων λειτουργιών ή η ενσωμάτωση επιπλέον αισθητήρων.

Η κατασκευή ενός Flask server για την αποθήκευση και προβολή των πειραματικών δεδομένων προσέφερε μεγάλη ευελιξία στην ανάλυση των μετρήσεων. Με την ενσωμάτωση του αισθητήρα AS7261 και την καταγραφή των τιμών RGB και NIR, καταφέραμε να δημιουργήσουμε μια πλατφόρμα η οποία διευκολύνει την καταγραφή, ανάλυση και οπτικοποίηση των δεδομένων σε πραγματικό χρόνο.

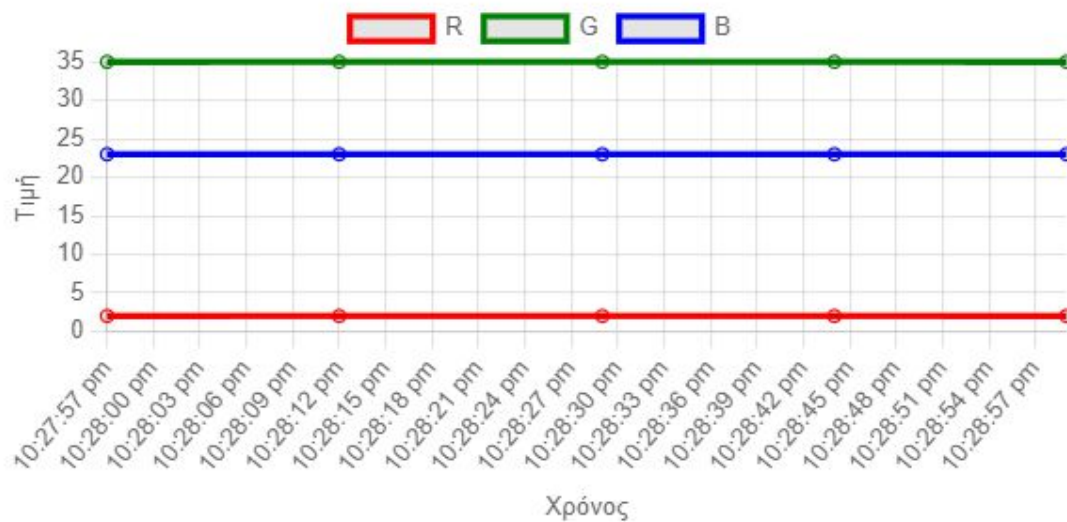
Αυτό το εργαλείο μπορεί να χρησιμοποιηθεί τόσο για εκπαιδευτικούς όσο και για ερευνητικούς σκοπούς, προσφέροντας μια ολοκληρωμένη λύση για την παρακολούθηση της ποιότητας του νερού και της άλγης, ενώ είναι ιδανικό για τη διεξαγωγή πειραμάτων που απαιτούν συνεχή παρακολούθηση και σύγκριση δεδομένων.

Παρακάτω παρατίθενται τα γραφήματα από κάθε δείγμα:

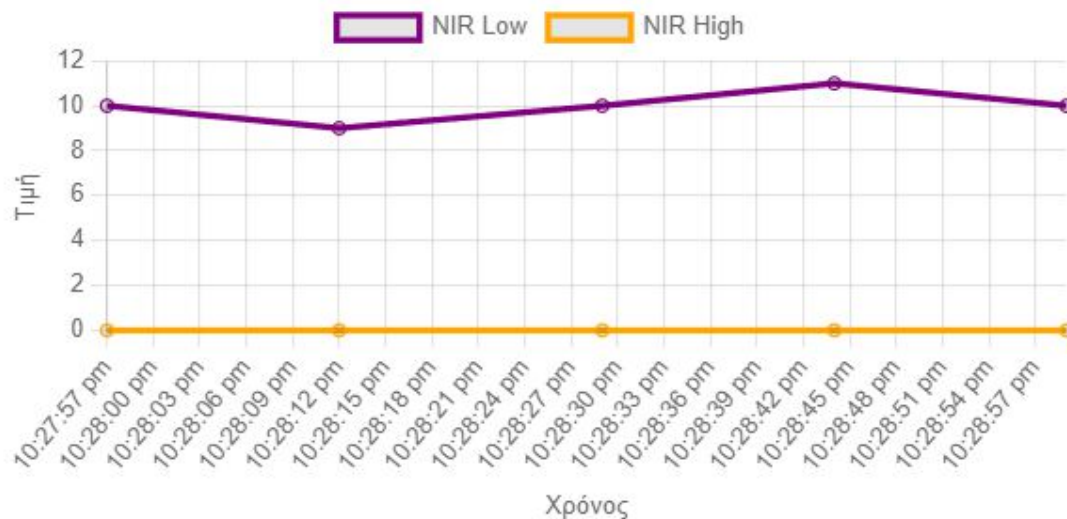
Επιλογή Group και Προβολή Δεδομένων

Επιλέξτε Group:

Δεδομένα Group (RGB)



Δεδομένα Group (NIR)

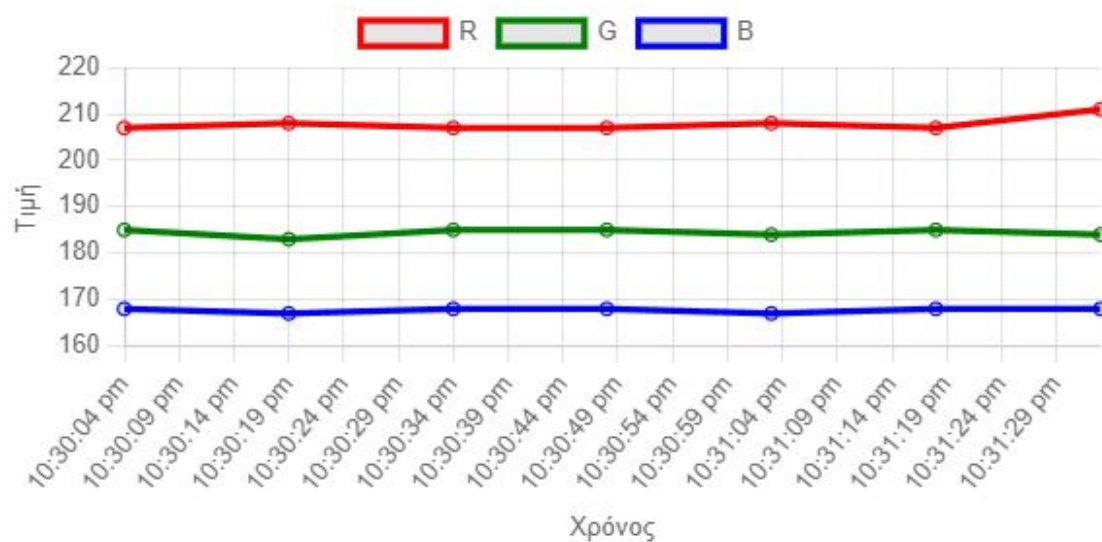


Εικόνα 4.2: Μέτρηση μαύρου

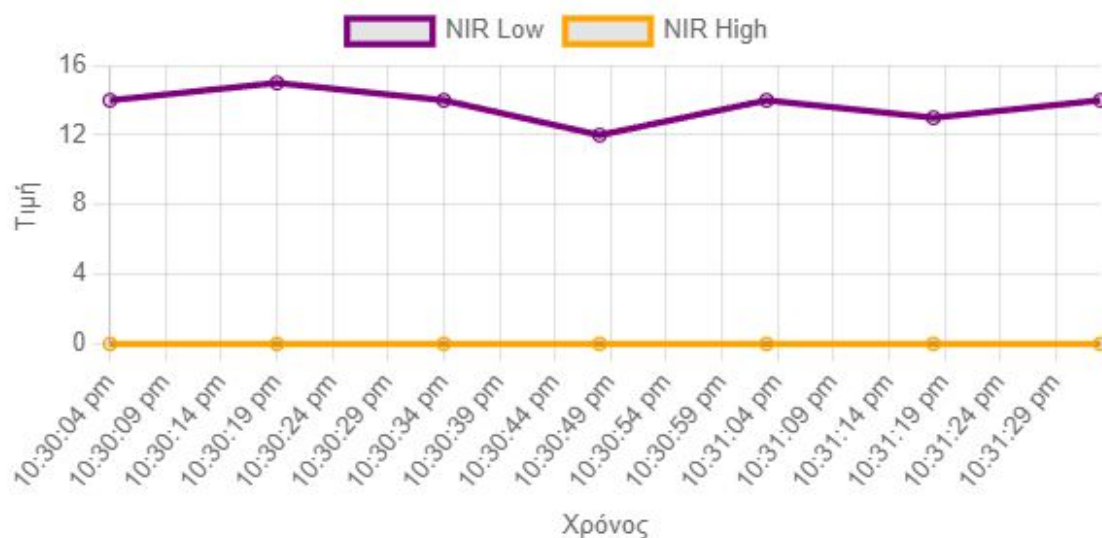
Επιλογή Group και Προβολή Δεδομένων

Επιλέξτε Group:

Δεδομένα Group (RGB)



Δεδομένα Group (NIR)

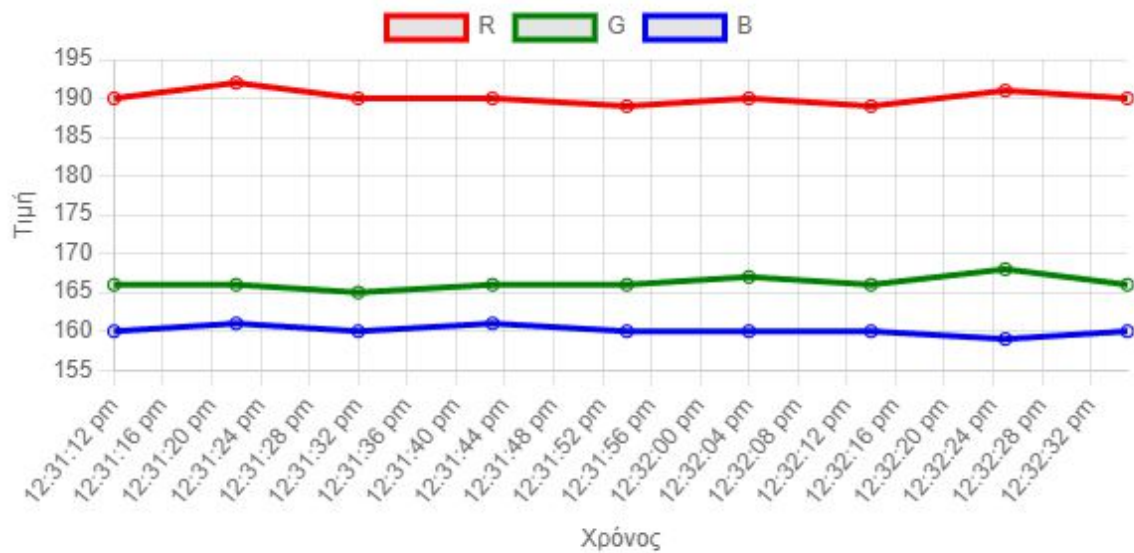


Εικόνα 4.3: Μέτρηση άσπρου

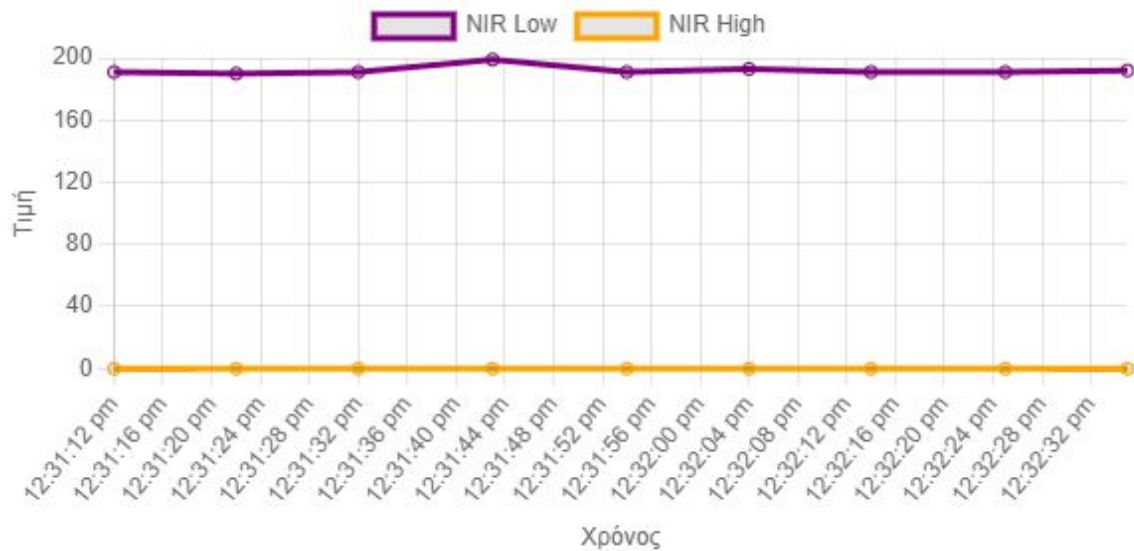
Επιλογή Group και Προβολή Δεδομένων

Επιλέξτε Group:

Δεδομένα Group (RGB)



Δεδομένα Group (NIR)

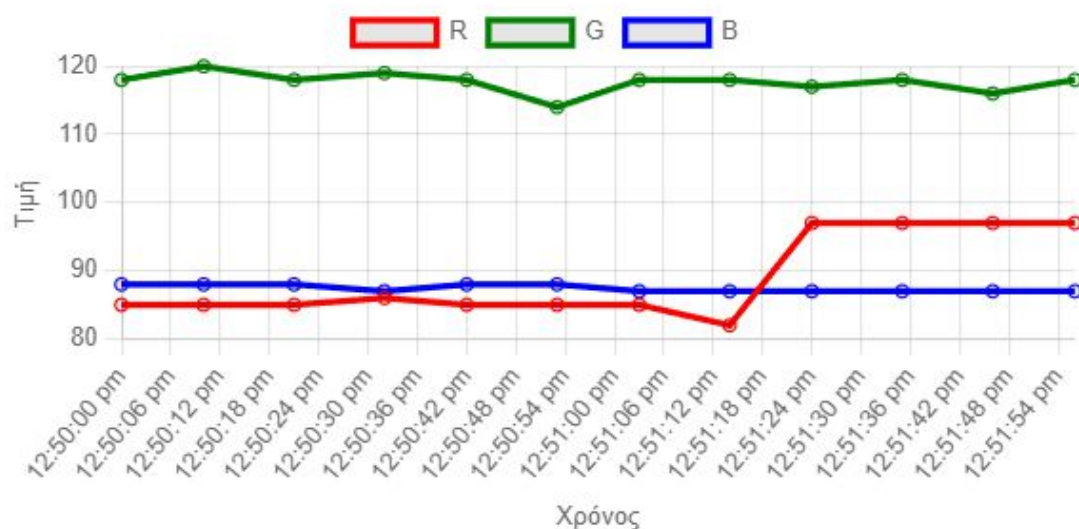


Εικόνα 4.4: Μέτρηση - Μόνο νερό

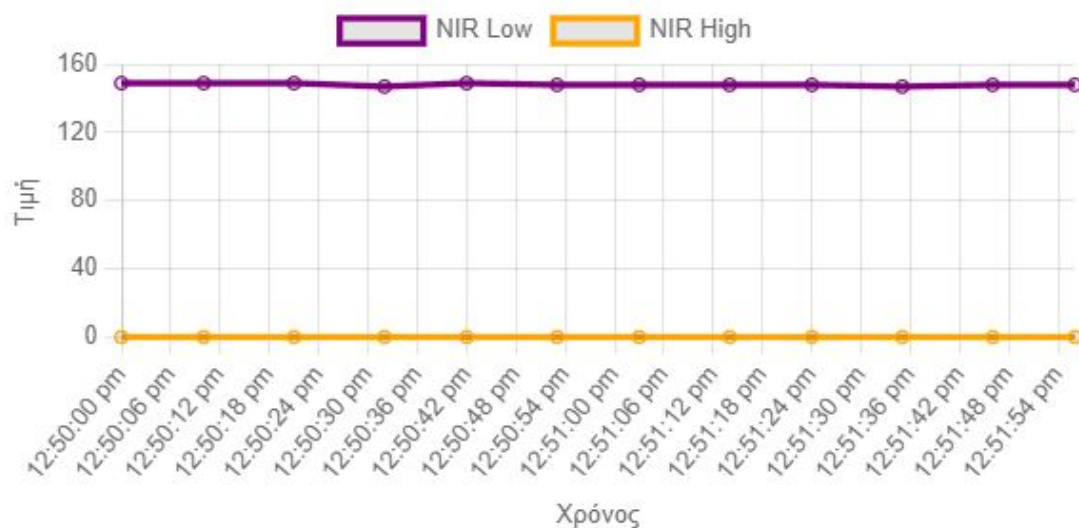
Επιλογή Group και Προβολή Δεδομένων

Επιλέξτε Group:

Δεδομένα Group (RGB)



Δεδομένα Group (NIR)

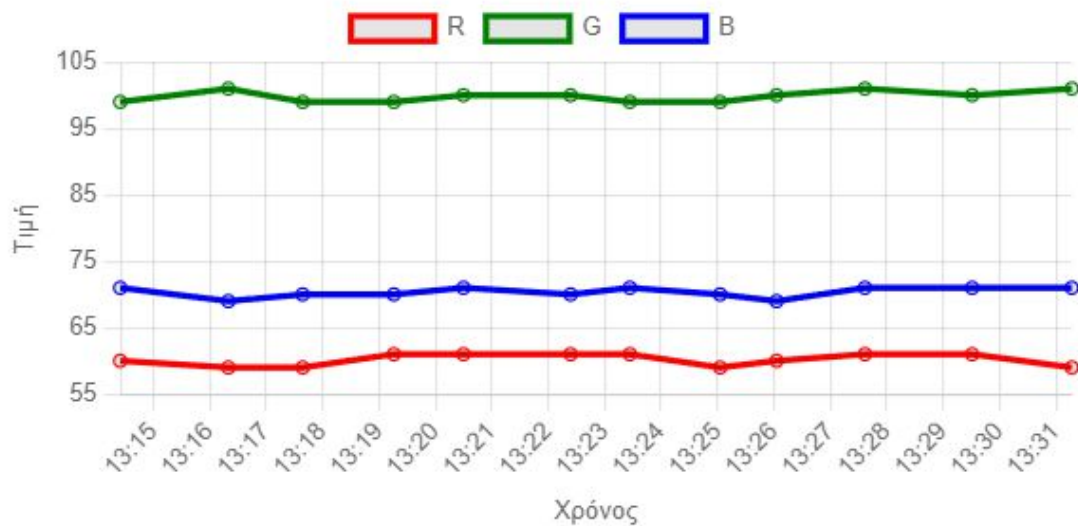


Εικόνα 4.5: Μέτρηση - Πράσινο δείγμα - Χωρίς σωματίδια

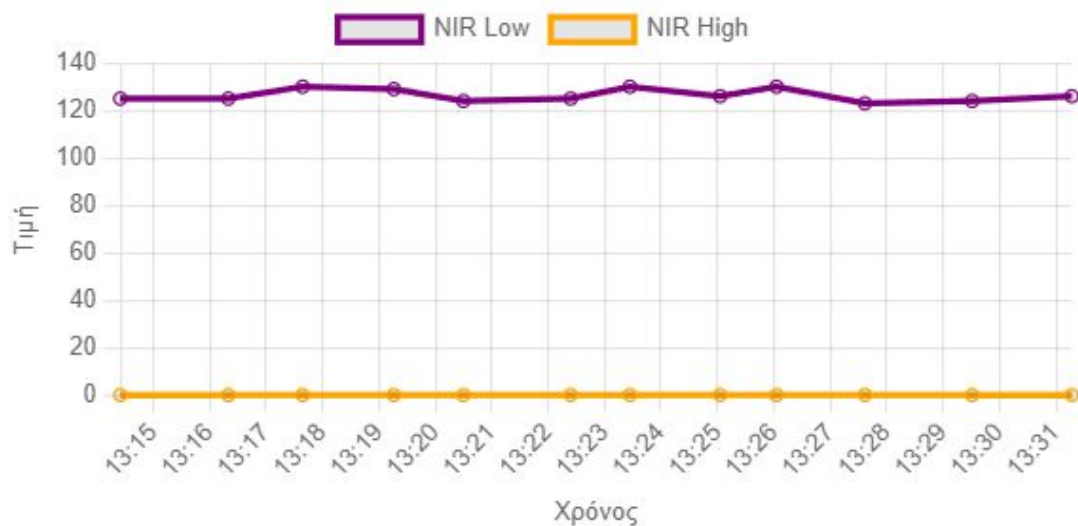
Επιλογή Group και Προβολή Δεδομένων

Επιλέξτε Group:

Δεδομένα Group (RGB)



Δεδομένα Group (NIR)

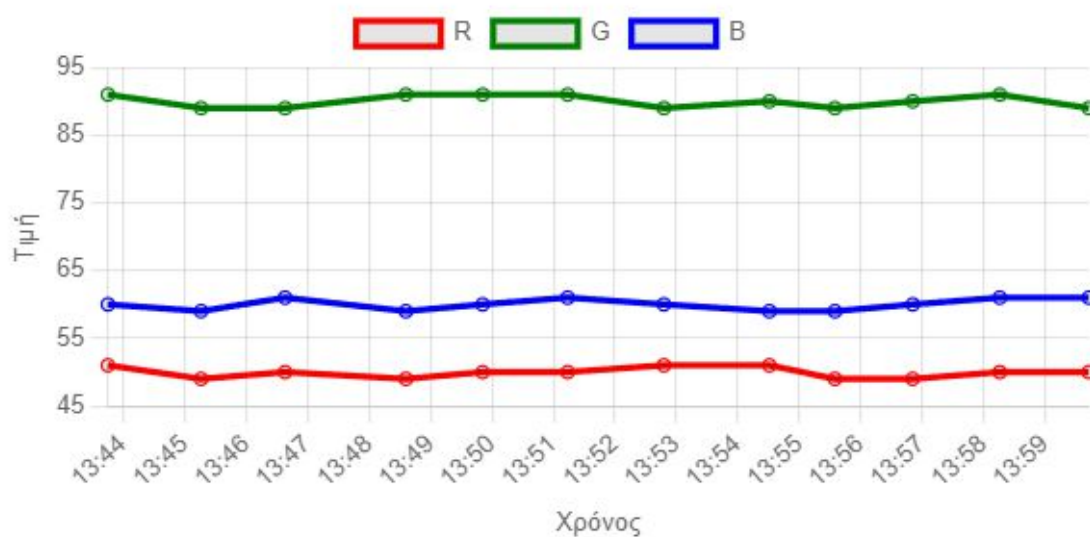


Εικόνα 4.6: Μέτρηση - Πράσινο δείγμα - Με σωματίδια όγκου 0.5cm^3

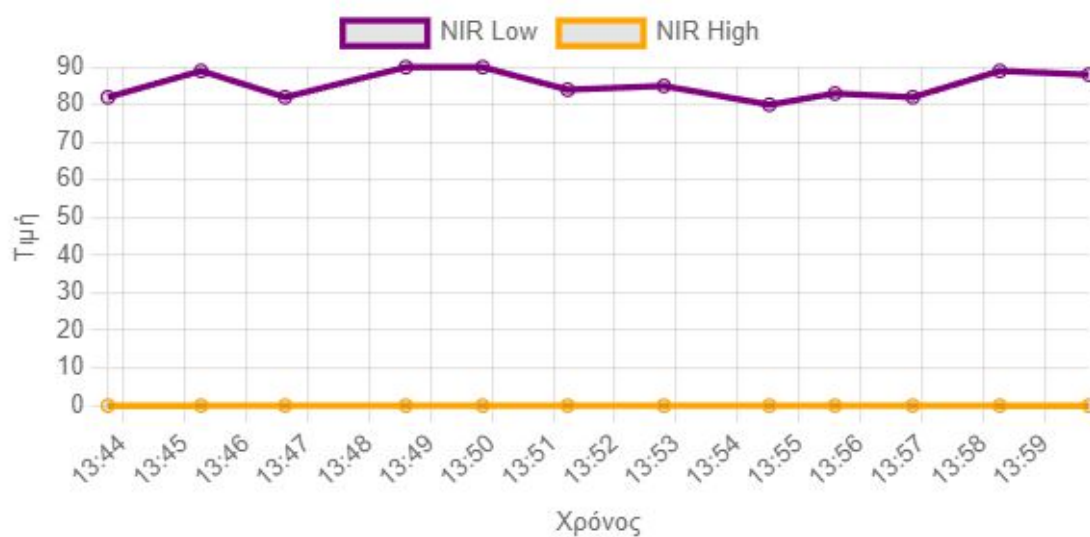
Επιλογή Group και Προβολή Δεδομένων

Επιλέξτε Group:

Δεδομένα Group (RGB)



Δεδομένα Group (NIR)

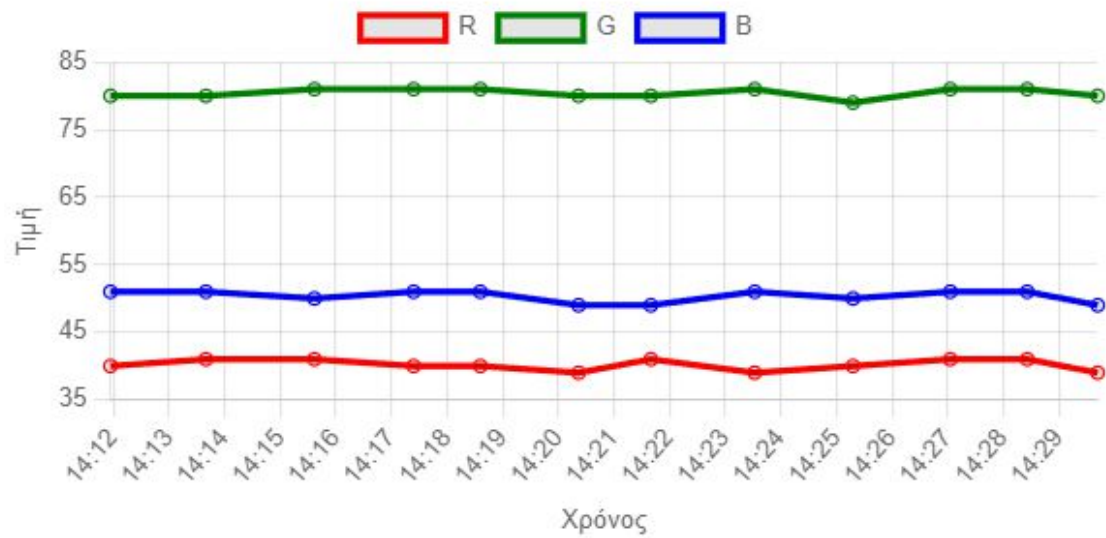


Εικόνα 4.7: Μέτρηση - Πράσινο δείγμα - Με σωματίδια όγκου 0.75cm^3

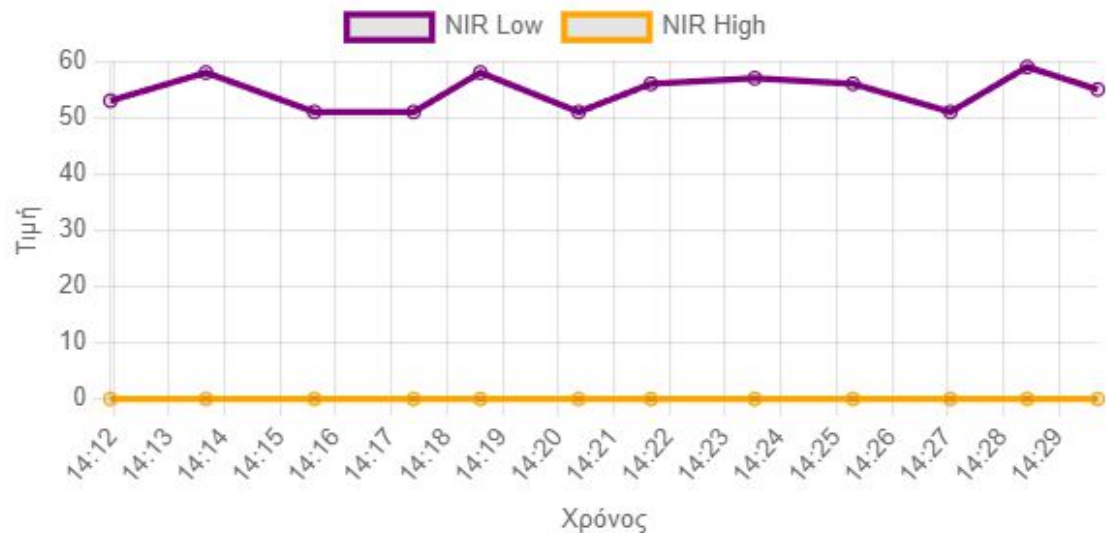
Επιλογή Group και Προβολή Δεδομένων

Επιλέξτε Group:

Δεδομένα Group (RGB)



Δεδομένα Group (NIR)



Εικόνα 4.8: Μέτρηση - Πράσινο δείγμα - Με σωματίδια όγκου 1cm³

Τα αποτελέσματα των πειραμάτων παρουσιάζουν μια σαφή τάση μείωσης των τιμών NIR καθώς αυξάνεται η παρουσία των φυτικών υλικών ή σωματιδίων μέσα στο δείγμα. Η αρχική μέτρηση του καθαρού νερού έδωσε την υψηλότερη τιμή NIR, ενώ οι μετρήσεις με αυξανόμενη ποσότητα σωματιδίων έδειξαν προοδευτική μείωση στις τιμές.

Συγκεκριμένα, παρατηρήθηκε ότι:

- Το καθαρό νερό επέτρεψε τη διέλευση σχεδόν όλου του NIR φωτός, δίνοντας υψηλή ένδειξη.
- Το πράσινο διάλυμα χωρίς σωματίδια έδειξε χαμηλότερη τιμή, λόγω της απορρόφησης από τη χλωροφύλλη.
- Τα δείγματα με σωματίδια έδειξαν ακόμα χαμηλότερες τιμές NIR, υποδεικνύοντας ότι τα σωματίδια παρεμπόδιζαν τη διέλευση του φωτός.

Ειδικός κώδικας σε Python για εξαγωγή

Χρησιμοποιώντας Python και συγκεκριμένα τις βιβλιοθήκες pandas, matplotlib, και numpy, δημιουργούμε γραφήματα και υπολογίζουμε τη γραμμική τάση, η οποία μας βοηθά να κατανοήσουμε τις τάσεις στα δεδομένα μας.

- **Δημιουργία Δεδομένων και Οπτικοποίηση**

Τα δεδομένα περιλαμβάνουν μετρήσεις της μέσης τιμής του NIR για πέντε διαφορετικά δείγματα:

1. **Μόνο νερό:** Δεν περιέχει σωματίδια.
2. **Πράσινο δείγμα - Χωρίς σωματίδια:** Περιέχει μόνο πράσινο χρώμα, χωρίς σωματίδια.
3. **Πράσινο δείγμα - Με σωματίδια συνολικού όγκου 0.5 cm³.**
4. **Πράσινο δείγμα - Με σωματίδια συνολικού όγκου 0.75 cm³.**
5. **Πράσινο δείγμα - Με σωματίδια συνολικού όγκου 1 cm³.**

Τα δεδομένα αυτά περιλαμβάνουν επίσης τις τιμές των καναλιών RGB, καθώς και τη μέση τιμή του NIR για κάθε δείγμα. Τα δεδομένα εισάγονται σε ένα DataFrame με τη βοήθεια της βιβλιοθήκης pandas, η οποία μας επιτρέπει να διαχειριζόμαστε και να αναλύουμε με ευκολία τις πληροφορίες.

Ανάλυση των Τιμών NIR

Η μέση τιμή NIR για κάθε δείγμα παρουσιάζεται σε διάγραμμα, ώστε να μπορούμε να δούμε την επίδραση του όγκου των σωματιδίων στη μετρηθείσα τιμή NIR. Η ανάλυση αυτή είναι σημαντική

καθώς μας δείχνει πώς το μέγεθος των σωματιδίων επηρεάζει την αντανάκλαση ή την απορρόφηση της υπέρυθρης ακτινοβολίας (NIR).

Στο διάγραμμα, η μέση τιμή NIR για κάθε δείγμα απεικονίζεται με μπλε χρώμα, και παρατηρούμε ότι η τιμή NIR μειώνεται όσο αυξάνεται ο όγκος των σωματιδίων. Η τάση αυτή αποτυπώνεται με μεγαλύτερη σαφήνεια με τη χρήση γραμμής τάσης (trend line).

Υπολογισμός Γραμμής Τάσης (Trend Line)

Για να κατανοήσουμε καλύτερα τη σχέση μεταξύ των δειγμάτων και της μέσης τιμής NIR, υπολογίζουμε μια γραμμή τάσης. Η γραμμή τάσης δείχνει τη γενική κατεύθυνση της σχέσης μεταξύ της μεταβλητής της ομάδας δειγμάτων και της τιμής NIR.

Χρησιμοποιούμε την `numpy` για να υπολογίσουμε την καλύτερη γραμμική προσαρμογή (linear fit) μέσω της συνάρτησης `np.polyfit()`. Αυτή η συνάρτηση χρησιμοποιεί τις τιμές των δειγμάτων και τις αντίστοιχες τιμές NIR για να υπολογίσει την κλίση και την τομή της ευθείας.

Κλίση (Slope): Η κλίση της γραμμής είναι ένα μέτρο του ρυθμού μεταβολής της τιμής NIR καθώς αυξάνεται ο όγκος των σωματιδίων. Στην περίπτωση μας, μια αρνητική κλίση δείχνει ότι όσο αυξάνεται ο όγκος των σωματιδίων, τόσο μειώνεται η τιμή NIR.

Τομή (Intercept): Η τομή της γραμμής με τον άξονα y δείχνει τη θεωρητική τιμή NIR όταν δεν υπάρχουν σωματίδια. Αυτό μας βοηθά να κατανοήσουμε την αρχική κατάσταση του δείγματος.

Συντελεστής Προσδιορισμού R^2 : Για να αξιολογήσουμε την ποιότητα της προσαρμογής της γραμμής τάσης στα δεδομένα, υπολογίζουμε τον συντελεστή προσδιορισμού R^2 χρησιμοποιώντας τη συνάρτηση `r2_score()` της `sklearn`. Ο συντελεστής R^2 είναι ένα μέτρο της ακρίβειας της γραμμής τάσης σε σχέση με τα πραγματικά δεδομένα:

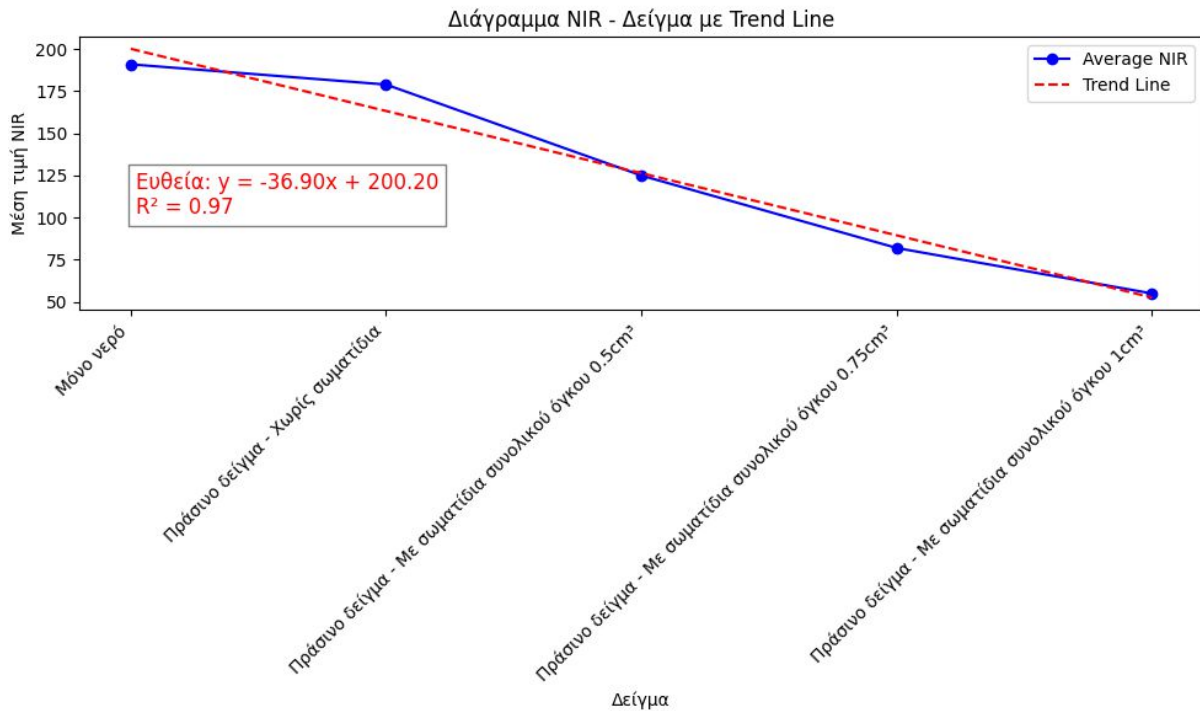
Ο συντελεστής R^2 κυμαίνεται από 0 έως 1. Μια τιμή κοντά στο 1 υποδηλώνει ότι η γραμμή τάσης περιγράφει καλά την διακύμανση στα δεδομένα. Στην περίπτωση μας, η τιμή R^2 δίνει μια εικόνα για το πόσο καλά η γραμμή τάσης αντιπροσωπεύει την τάση της μέσης τιμής NIR σε σχέση με τα δείγματα.

Οπτικοποίηση του Διαγράμματος

Το διάγραμμα που δημιουργείται παρουσιάζει:

- Τις μέσες τιμές NIR για κάθε δείγμα ως μπλε σημεία που συνδέονται με μια γραμμή, ώστε να φαίνεται η τάση με την προσθήκη σωματιδίων.
- Τη γραμμή τάσης, η οποία προστίθεται με κόκκινο διακεκομμένο χρώμα, για να δείξει τη γραμμική σχέση μεταξύ των δειγμάτων και της τιμής NIR.

- Μια επεξήγηση στο γράφημα που περιλαμβάνει την εξίσωση της γραμμής τάσης καθώς και την τιμή R².



Εικόνα 4.9: Διάγραμμα μέσης τιμής NIR - Δείγμα με Trend Line

Ανάλυση Αποτελεσμάτων

Από το διάγραμμα και τη γραμμή τάσης, μπορούμε να δούμε ότι υπάρχει μια σαφής αρνητική τάση στις τιμές NIR όσο αυξάνεται ο όγκος των σωματιδίων. Αυτό σημαίνει ότι η προσθήκη σωματιδίων στο δείγμα οδηγεί σε μείωση της αντανάκλασης της υπέρυθρης ακτινοβολίας, πιθανώς λόγω της αυξημένης απορρόφησης από τα σωματίδια.

Αυτό το αποτέλεσμα είναι συνεπές με την αναμενόμενη φυσική συμπεριφορά των σωματιδίων που προστίθενται στο δείγμα. Όσο περισσότερα σωματίδια υπάρχουν, τόσο μεγαλύτερη είναι η πιθανότητα να απορροφηθεί η ακτινοβολία, αντί να ανακλαστεί. Το γεγονός αυτό αντικατοπτρίζεται άμεσα στις μετρήσεις του NIR, οι οποίες μειώνονται με την αύξηση του όγκου των σωματιδίων.

Τα πειράματα έδειξαν ότι το σύστημα μπορεί να χρησιμοποιηθεί αποτελεσματικά για την ανίχνευση και την εκτίμηση της ποσότητας σωματιδίων ή άλγης σε νερό. Οι τιμές NIR μειώνονται όσο αυξάνεται η συγκέντρωση των φυτικών υλικών ή σωματιδίων, παρέχοντας έναν άμεσο τρόπο για την αξιολόγηση της καθαρότητας ή της παρουσίας άλγης σε υδατικά δείγματα.

Η μεθοδολογία αυτή θα μπορούσε να επεκταθεί για την παρακολούθηση των υδάτινων οικοσυστημάτων και τον έλεγχο της ποιότητας του νερού, παρέχοντας μια απλή αλλά αξιόπιστη μέθοδο για την ανάλυση της παρουσίας οργανικών ή ανόργανων σωματιδίων. Η τεχνολογία του αισθητήρα

AS7261 σε συνδυασμό με τις μετρήσεις NIR αποτελεί ένα ισχυρό εργαλείο για την ανίχνευση και παρακολούθηση περιβαλλοντικών αλλαγών.

4.4 Η Βάση του συστήματος

Ο πίνακας `amsvalues` περιέχει δεδομένα που φαίνεται να σχετίζονται με αναλύσεις χρώματος και μετρήσεις φωτεινότητας σε διαφορετικές ομάδες δεδομένων.

1. **id (int(11), NOT NULL):**

- Είναι η κύρια στήλη αναγνώρισης για κάθε γραμμή. Η χρήση του `int(11)` υποδεικνύει ότι μπορεί να αποθηκεύσει ακέραιες τιμές μέχρι $\pm 2,147,483,647$.
- Ορίζεται σαν `AUTO_INCREMENT`

2. **R (int(11), NOT NULL):**

- Αντιπροσωπεύει την κόκκινη (Red) συνιστώσα ενός χρώματος.
- Οι τιμές είναι ακέραιοι αριθμοί και δεν επιτρέπεται να είναι `NULL`.

3. **G (int(11), NOT NULL):**

- Αντιπροσωπεύει την πράσινη (Green) συνιστώσα ενός χρώματος.
- Οι τιμές είναι επίσης ακέραιοι αριθμοί και δεν επιτρέπεται να είναι `NULL`.

4. **B (int(11), NOT NULL):**

- Αντιπροσωπεύει την μπλε (Blue) συνιστώσα ενός χρώματος.
- Οι τιμές είναι ακέραιοι αριθμοί και δεν επιτρέπεται να είναι `NULL`.

5. **nirl (int(11), NOT NULL):**

- Αναφέρεται στη χαμηλή τιμή μέτρησης NIR (Near-Infrared).
- Η τιμή είναι ακέραιος αριθμός και πρέπει να υπάρχει.

6. **nirh (int(11), NOT NULL):**

- Αναφέρεται στη υψηλή τιμή μέτρησης NIR.
- Η τιμή είναι ακέραιος αριθμός και πρέπει να υπάρχει.

7. **groupname (varchar(255), DEFAULT 'default_group'):**

- Αυτή η στήλη κρατάει το όνομα της ομάδας στην οποία ανήκουν τα δεδομένα.

- Έχει τύπο varchar με μήκος 255 χαρακτήρες και προεπιλεγμένη τιμή 'default_group'.
- Αυτό το πεδίο είναι προαιρετικό, αν και αν δεν παρέχεται τιμή, θα χρησιμοποιηθεί η προεπιλεγμένη τιμή.

8. created_at (datetime, DEFAULT current_timestamp()):

- Αυτή η στήλη αποθηκεύει την ημερομηνία και ώρα δημιουργίας της γραμμής.
- Η προεπιλεγμένη τιμή είναι το τρέχον χρονικό στίγμα κατά την εισαγωγή της γραμμής.

Διασφάλιση Ακεραιότητας:

- Όλα τα πεδία (εκτός από το groupname) είναι NOT NULL, κάτι που σημαίνει ότι πρέπει να έχουν πάντα τιμή, εξασφαλίζοντας ότι όλες οι γραμμές είναι πλήρεις.
- Ο πίνακας υποστηρίζει διαφορετικές ομάδες δεδομένων μέσω της στήλης groupname, διευκολύνοντας την οργάνωση και την κατηγοριοποίηση των δεδομένων.
- Η στήλη R, G, B φαίνεται να αποθηκεύει δεδομένα που σχετίζονται με το χρώμα σε RGB μορφή. Οι τιμές αυτές μαζί με τα nir1 και nir2 είναι χρήσιμες για την ανάλυση χρωμάτων και μετρήσεων φωτεινότητας, πιθανώς σε σχέση με την άλγη ή άλλες μετρήσεις.

Κεφάλαιο 5ο: Συμπεράσματα και προτάσεις βελτίωσης

Η παρούσα εργασία επικεντρώθηκε στην ανάπτυξη και εφαρμογή ενός συστήματος για τη μελέτη και εκτίμηση της ποσότητας σωματιδίων σε σωλήνα με χρήση ενός χρωματικού αισθητήρα, του AS7261. Το σύστημα αξιοποίησε τις δυνατότητες του αισθητήρα για μη επεμβατική μέτρηση των χαρακτηριστικών των σωματιδίων, όπως το χρώμα, και συνέβαλε στην ακριβή εκτίμηση της ποσότητάς τους. Η χρήση της φασματικής απόκρισης του αισθητήρα έδωσε τη δυνατότητα να κατανοηθούν καλύτερα οι αλλαγές στη σύσταση και την ποσότητα των σωματιδίων, με εφαρμογές σε ποικίλους τομείς.

Αναπτύχθηκε ειδικός κώδικας σε C++ με χρήση της I²C επικοινωνίας, ώστε να επιτευχθεί η σύνδεση του αισθητήρα με τον υπολογιστή. Παράλληλα, αναπτύχθηκε server για την αποστολή, αποθήκευση και προβολή των δεδομένων, επιτρέποντας τη συνεχή παρακολούθηση και ανάλυση των μετρήσεων σε πραγματικό χρόνο. Η προσέγγιση αυτή απέδειξε ότι η χρήση τεχνολογιών σε συνδυασμό με αισθητήρες χαμηλού κόστους, μπορεί να συμβάλλει στην επίτευξη ακριβών αποτελεσμάτων χωρίς την ανάγκη πολύπλοκων ή δαπανηρών μεθόδων.

Η μη καταστροφική προσέγγιση που ακολουθήθηκε είχε πλεονεκτήματα όσον αφορά την ευκολία της διαδικασίας, την ταχύτητα των μετρήσεων και την ακρίβεια. Ο χρωματικός αισθητήρας AS7261 αποδείχθηκε εξαιρετικά χρήσιμος για την εκτίμηση των χαρακτηριστικών των σωματιδίων, ενώ η χρήση των χρωματικών καναλιών XYZ αποδείχθηκε κατάλληλη για την ανάλυση του περιεχομένου σε σωλήνα.

Τα αποτελέσματα της εργασίας δείχνουν ότι το αναπτυχθέν σύστημα μπορεί να χρησιμοποιηθεί σε εφαρμογές, όπου η παρακολούθηση της ποιότητας και η αξιολόγηση της σύστασης των υλικών είναι κρίσιμη. Επιπλέον, η εφαρμογή του συστήματος σε συνθήκες πραγματικού χρόνου βελτιώνει τη διαχείριση των πόρων και επιτρέπει καλύτερο έλεγχο της ποιότητας των προϊόντων, με ελάχιστο κόστος και ανθρώπινη παρέμβαση.

Συνοψίζοντας, η παρούσα εργασία εισήγαγε μια αποδοτική και οικονομικά προσιτή λύση για την εκτίμηση της ποσότητας και της σύστασης σωματιδίων σε σωλήνα, με σημαντικές προοπτικές εφαρμογής τόσο στη βιομηχανία όσο και στην επιστημονική έρευνα. Μελλοντική βελτίωση της προσέγγισης αυτής θα μπορούσε να περιλαμβάνει τη χρήση αισθητήρων υψηλότερης ακρίβειας και την ενσωμάτωση πιο εξελιγμένων μεθόδων μηχανικής μάθησης για την ανάλυση των δεδομένων.

ΒΙΒΛΙΟΓΡΑΦΙΑ

- [1] Srinivasagan R, Mohammed M, Alzahrani A. TinyML-Sensor for Shelf Life Estimation of Fresh Date Fruits. *Sensors*. 2023; 23(16):7081. <https://doi.org/10.3390/s23167081>
- [2] Talon, D., Dominique, D., Sophie, J., Séverine, B., & Matthieu, I. Impact of Luminance and Correlated Colour Temperature on Drivers' Visibility in Road Tunnels' Interior Zones. Available at SSRN 4518289.
- [3] Gomes, Y. D. S., SAIDI, M., Lushnikova, A., & Plé, O. (2023). Development of a fibre optic-based patch sensor for monitoring structures. *Academic Journal of Civil Engineering*, 41(1), 407-416.
- [4] Kymissis, I. J., & Pervez, N. Use of the Chromation Spectrometer for White Point Balancing in Mobile Devices.
- [5] Kapse, S., Kedia, P., Kausley, S. et al. Nondestructive Evaluation of Banana Maturity Using NIR AS7263 Sensor. *J Nondestruct Eval* 42, 30 (2023). <https://doi.org/10.1007/s10921-023-00943-z>
- [6] Guido van Rossum και άλλοι, "The Python Language Reference", Python Software Foundation, <https://docs.python.org/3/reference/>.
- [7] Mark Lutz, "Learning Python", 5th Edition, O'Reilly Media, 2013.
- [8] David Beazley και Brian K. Jones, "Python Cookbook", 3rd Edition, O'Reilly Media, 2013.
- [9] Eric Matthes, "Python Crash Course: A Hands-On, Project-Based Introduction to Programming", No Starch Press, 2019.
- [10] Python Software Foundation, "The Python Standard Library", <https://docs.python.org/3/library/>.
- [11] Morchid, A., Jebabra, R., Ismail, A., Khalid, H. M., El Alami, R., Qjidaa, H., & Jamil, M. O. (2024). IoT-enabled fire detection for sustainable agriculture: A real-time system using flask and embedded technologies. *Results in Engineering*, 23, 102705.
- [12] https://alex-robenko.gitbook.io/bare_metal_cpp/peripherals/i2c
- [13] <https://ams-osram.com/products/sensors/ambient-light-color-spectral-proximity-sensors/ams-as7261-spectral-sensing-engine>
- [14] <https://docs.rs-online.com/bdb3/A700000006921377.pdf>
- [15] <https://eu.mouser.com/datasheet/2/588/AS726x-iSPI.Evlauation.Kit.User.Guide.2V0-1140337.pdf>

ΠΑΡΑΡΤΗΜΑ Α

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

Python Server

```
from flask import Flask, request, jsonify, render_template
import mysql.connector
from datetime import datetime

app = Flask(__name__)

# MySQL configuration
db_config = {
    'user': 'root',
    'password': '',
    'host': 'localhost',
    'database': 'amsdb'
}

# Function to insert data into MySQL
def insert_into_db(R, G, B, nirl, nirh, group):
    try:
        conn = mysql.connector.connect(**db_config)
        cursor = conn.cursor()

        query = ("INSERT INTO amsvalues (R, G, B, nirl, nirh, `groupname`, created_at) "
                "VALUES (%s, %s, %s, %s, %s, %s, %s)")
        data = (R, G, B, nirl, nirh, groupname, datetime.now())

        cursor.execute(query, data)
        conn.commit()
        cursor.close()
        conn.close()
        return True
    except mysql.connector.Error as err:
        print(f"Error: {err}")
        return False

# Route to handle GET request for RGB values
@app.route('/rgb', methods=['GET'])
def receive_rgb():
    R = request.args.get('R')
    G = request.args.get('G')
    B = request.args.get('B')
    group = request.args.get('group', 'default_group')
    nirl = request.args.get('nirl')
    nirh = request.args.get('nirh')
```

```

if R is None or G is None or B is None:
    return jsonify({"error": "Missing R, G, or B value"}), 400
if nir1 is None or nirh is None:
    return jsonify({"error": "Missing NIR value"}), 400

try:
    R = int(R)
    G = int(G)
    B = int(B)
    nir1 = int(nir1)
    nirh = int(nirh)

    if insert_into_db(R, G, B, nir1, nirh, groupname):
        return jsonify({"status": "success", "R": R, "G": G, "B": B, "nir1": nir1, "nirh": nirh,
"group": group}), 200
    else:
        return jsonify({"status": "error", "message": "Failed to insert into database"}), 500

except ValueError:
    return jsonify({"error": "R, G, B, nir1, and nirh must be integers"}), 400

# Function to get the latest value from MySQL
def get_latest_value():
    try:
        conn = mysql.connector.connect(**db_config)
        cursor = conn.cursor(dictionary=True)
        cursor.execute("SELECT * FROM amsvalues ORDER BY created_at DESC LIMIT 1")
        result = cursor.fetchone()
        cursor.close()
        conn.close()
        return result
    except mysql.connector.Error as err:
        print(f"Error: {err}")
        return None

# Function to get values for a specific group
def get_group_values(group):
    try:
        conn = mysql.connector.connect(**db_config)
        cursor = conn.cursor(dictionary=True)
        query = "SELECT * FROM amsvalues WHERE `groupname` = %s ORDER BY created_at"
        cursor.execute(query, (group,))
        result = cursor.fetchall()
        cursor.close()
        conn.close()
        return result
    except mysql.connector.Error as err:
        print(f"Error: {err}")

```

```

        return []

# Function to get available groups from MySQL
def get_available_groups():
    try:
        conn = mysql.connector.connect(**db_config)
        cursor = conn.cursor(dictionary=True)
        cursor.execute("SELECT DISTINCT `groupname` FROM amsvalues")
        result = cursor.fetchall()
        cursor.close()
        conn.close()
        return result
    except mysql.connector.Error as err:
        print(f"Error: {err}")
        return []

# Route to get all available groups
@app.route('/groups', methods=['GET'])
def get_groups():
    groups = get_available_groups()
    return jsonify(groups)

# Route to render the group selection page with charts
@app.route('/group-selection')
def group_selection():
    return render_template('group_selection.html')

# Route to get the latest value (for live update)
@app.route('/latest', methods=['GET'])
def latest():
    latest_value = get_latest_value()
    if latest_value:
        return jsonify(latest_value), 200
    else:
        return jsonify({"error": "No data available"}), 500

# Route to get group data for charts
@app.route('/group/<group>', methods=['GET'])
def group_data(group):
    group_values = get_group_values(group)
    if group_values:
        return jsonify(group_values), 200
    else:
        return jsonify({"error": "No data found for the group"}), 404

# Route to render the main page with charts (if needed)
@app.route('/')
def index():

```

```
    return render_template('index.html')

# Main entry point
if __name__ == '__main__':
    app.run(host='0.0.0.0', port=2323, debug=True)
```

group_selection.html

```
<!DOCTYPE html>
<html lang="el">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>AMS Επιλογή Group</title>
    <!-- Θόρτωση jQuery και Chart.js -->
    <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
    <script src="https://cdn.jsdelivr.net/npm/chart.js"></script>
    <script src="https://cdn.jsdelivr.net/npm/moment@2.29.1/moment.min.js"></script>
    <script src="https://cdn.jsdelivr.net/npm/chartjs-adapter-moment@1.0.0"></script>
</head>
<body>
    <h1>Επιλογή Group και Προβολή Δεδομένων</h1>

    <!-- Dropdown menu για την επιλογή group -->
    <label for="groupSelect">Επιλέξτε Group:</label>
    <select id="groupSelect">
        <option value="">Θόρτωση groups...</option>
    </select>

    <!-- Χώρος για το chart RGB -->
    <div style="width:30%; margin-top: 30px;">
        <h2>Δεδομένα Group (RGB)</h2>
        <canvas id="groupChart"></canvas>
    </div>

    <!-- Χώρος για το chart NIR -->
    <div style="width:30%; margin-top: 30px;">
        <h2>Δεδομένα Group (NIR)</h2>
        <canvas id="nirChart"></canvas>
    </div>

    <script>
```

```

// Συνάρτηση για να φορτώσει τα διαθέσιμα groups στο dropdown menu
function loadAvailableGroups() {
    $.get("/groups", function(groups) {
        const groupSelect = $('#groupSelect');
        groupSelect.empty(); // Καθαρισμός των προηγούμενων επιλογών
        groupSelect.append(new Option("Επιλέξτε Group", "")); // Προσθήκη default επιλογής

        // Προσθήκη των διαθέσιμων groups στο dropdown
        groups.forEach(function(group) {
            groupSelect.append(new Option(group.groupname, group.groupname));
        });
    });
}

// Συνάρτηση για να φορτώσει τα δεδομένα του group και να ενημερώσει τα charts
function loadGroupData(group) {
    $.get(`/group/${group}`, function(data) {
        // Ετοιμασία των δεδομένων για το RGB chart
        let labels = data.map(item => item.created_at);
        let rValues = data.map(item => item.R);
        let gValues = data.map(item => item.G);
        let bValues = data.map(item => item.B);

        // Δημιουργία ή ενημέρωση του RGB chart
        const ctxRGB = document.getElementById('groupChart').getContext('2d');
        const rgbChart = new Chart(ctxRGB, {
            type: 'line',
            data: {
                labels: labels,
                datasets: [{
                    label: 'R',
                    data: rValues,
                    borderColor: 'red',
                    fill: false
                }, {
                    label: 'G',
                    data: gValues,
                    borderColor: 'green',
                    fill: false
                }, {

```

```

        label: 'B',
        data: bValues,
        borderColor: 'blue',
        fill: false
    }]
    },
options: {
    scales: {
        x: {
            type: 'time', // Καθορίζει ότι ο άξονας είναι χρονικός
            time: {
                displayFormats: {
                    minute: 'HH:mm', // Δείχνει μόνο την ώρα και τα λεπτά
                    hour: 'HH:mm', // Προαιρετικά για τις μεγαλύτερες μονάδες
                },
                tooltipFormat: 'YYYY-MM-DD HH:mm:ss', // Πλήρης εμφάνιση στο tooltip
            },
            title: {
                display: true,
                text: 'Χρόνος'
            }
        },
        y: {
            title: {
                display: true,
                text: 'Τιμή'
            }
        }
    }
}

});

// Ετοιμασία των δεδομένων για το NIR chart
let nirlValues = data.map(item => item.nirl);
let nirhValues = data.map(item => item.nirh);

// Δημιουργία ή ενημέρωση του NIR chart
const ctxNIR = document.getElementById('nirChart').getContext('2d');
const nirChart = new Chart(ctxNIR, {
    type: 'line',

```

```

        data: {
            labels: labels,
            datasets: [{
                label: 'NIR Low',
                data: nirlValues,
                borderColor: 'purple',
                fill: false
            }, {
                label: 'NIR High',
                data: nirhValues,
                borderColor: 'orange',
                fill: false
            }]
        },
options: {
    scales: {
        x: {
            type: 'time', // Καθορίζει ότι ο άξονας είναι χρονικός
            time: {
                displayFormats: {
                    minute: 'HH:mm', // Δείχνει μόνο την ώρα και τα λεπτά
                    hour: 'HH:mm', // Προαιρετικά για τις μεγαλύτερες μονάδες
                },
                tooltipFormat: 'YYYY-MM-DD HH:mm:ss', // Πλήρης εμφάνιση στο tooltip
            },
            title: {
                display: true,
                text: 'Χρόνος'
            }
        },
        y: {
            title: {
                display: true,
                text: 'Τιμή'
            }
        }
    }
}
});
});

```

```
}

// Όταν φορτωθεί η σελίδα, φορτώνουμε τα διαθέσιμα groups
$(document).ready(function() {
    loadAvailableGroups();

    // Όταν αλλάζει το group από το dropdown, φορτώνουμε τα δεδομένα
    $('#groupSelect').on('change', function() {
        const selectedGroup = $(this).val();
        if (selectedGroup) {
            loadGroupData(selectedGroup); // Φορτώνουμε τα δεδομένα του επιλεγμένου group
        }
    });
});
</script>
</body>
</html>
```