

Για την λήψη αυτών των μετρήσεων θα χρησιμοποιήσουμε αισθητήρα BME280 ο οποίος θα συνδεθεί στον κεντρικό επεξεργαστή του CanSat. Την ένδειξη του υψομέτρου θα την έχουμε είτε από το GPS του CanSat, είτε μέσω υπολογισμού από τον Επίγειο Σταθμό όπου θα υπάρχει ανάλογος αισθητήρας BME280 για τις τιμές εδάφους.

Τα δεδομένα της **δευτερεύουσας αποστολής** περιλαμβάνουν:

- (α) την θέση GPS,
- (β) τις εικόνες,
- (γ) τις εντολές από τον Επίγειο Σταθμό προς το CanSat.

Η θέση GPS αποστέλλεται τακτικά στον Επίγειο Σταθμό, ώστε να συμβάλλει στην ανάκτηση του CanSat.

Για την λήψη των εικόνων χρησιμοποιούνται δύο κάμερες ESP-32CAM με αισθητήρα OV2640 ο οποίος καταγράφει στην ορατή RGB περιοχή του φάσματος (380-720nm). Οι εικόνες αποθηκεύονται τοπικά στο CanSat και η επεξεργασία τους θα γίνει μετά την προσγείωση, ώστε να εξαχθούν συμπεράσματα για την περιοχή (βλάστηση, ύπαρξη νερού, κλπ). Έχουμε δημιουργήσει και εκπαιδεύσει ένα Multi-Layer Perceptron νευρωνικό δίκτυο για τον σκοπό αυτό. Πριν οι εικόνες δοθούν στο νευρωνικό δίκτυο θα εφαρμόσουμε γεωμετρική διόρθωση (ορθοαναγωγή), καθώς συμβαίνει συχνά οι δορυφορικές εικόνες να έχουν γεωμετρικές παραμορφώσεις για πολλούς λόγους.

Οι εντολές από τον Επίγειο Σταθμό προς το CanSat προϋποθέτουν την δυνατότητα για αμφίδρομη επικοινωνία. Δεν χρησιμοποιούνται δύο συχνότητες. Η αμφίδρομη επικοινωνία θα επιτευχθεί με έναν πομποδέκτη στο CanSat κι έναν πομποδέκτη στον Επίγειο Σταθμό και με την εφαρμογή κατάλληλων προγραμματιστικών τεχνικών, ώστε να μην χάνονται τα δεδομένα που πρόκειται να σταλούν όσο ο πομποδέκτης λαμβάνει και το αντίστροφο.

1.2.3 Πρωτόκολλο ασύρματης οπτικής επικοινωνίας

Η τεχνική καινοτομία με την οποία πειραματιζόμαστε είναι η ασύρματη οπτική επικοινωνία μεταξύ των δύο επεξεργαστών του CanSat.

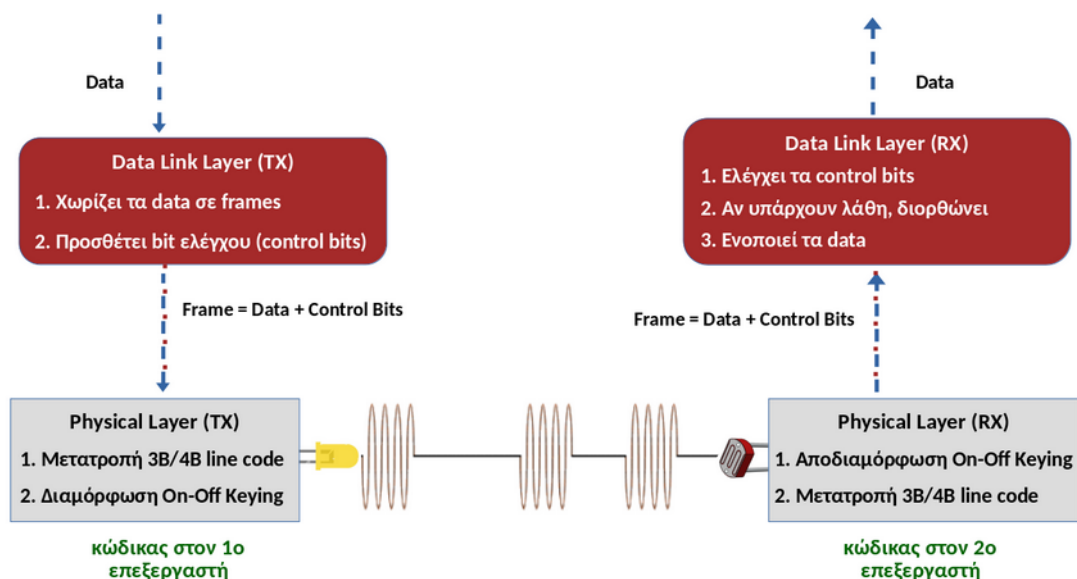
Επειδή δεν υπάρχουν (σε κόστος και μέγεθος που να μπορούν να ενσωματωθούν στο CanSat) πομποί που να διαμορφώνουν και να αποστέλλουν σήματα στο ορατό φάσμα, όπως επίσης δεν υπάρχουν αντίστοιχοι δέκτες που να λαμβάνουν και να αποδιαμορφώνουν τέτοια σήματα, αποφασίσαμε να τους υλοποιήσουμε συνδυάζοντας απλά υλικά και κώδικα.

Τα υλικά που χρησιμοποιούμε είναι ένα LED για τον πομπό και μια φωτοναντίσταση (LDR) για τον δέκτη. Το LED εκπέμπει φωτεινά σήματα καθοδηγούμενο από τον κώδικα του πρωτοκόλλου που σχεδιάσαμε. Η LDR λαμβάνει τα φωτεινά σήματα και μέσω κώδικα αυτά μετατρέπονται σε δεδομένα.

Το πρωτόκολλο που σχεδιάσαμε περιλαμβάνει λειτουργίες που γίνονται στο Physical Layer και στο Data Link Layer, όπως φαίνεται συνοπτικά στην Εικόνα 2. Τα βασικά χαρακτηριστικά του πρωτοκόλλου είναι τα εξής:

1. Τα δεδομένα χωρίζονται σε frames σταθερού μεγέθους.
2. Στην αρχή κάθε frame υπάρχει preamble (συγκεκριμένος δυαδικός κωδικός) που δείχνει την αρχή νέου frame και βοηθά στον συντονισμό μεταξύ πομπού κ δέκτη.
3. Μεταξύ των data bit παρεμβάλλονται control bits σε συγκεκριμένες θέσεις, τα οποία υπολογίζονται σύμφωνα με τον αλγόριθμο Hamming, ώστε να είναι εφικτή η ανακάλυψη (και πιθανώς η διόρθωση, χωρίς αναμετάδοση) λαθών που θα συμβούν κατά τη διάδοση.
4. Χρησιμοποιούμε Line Code 3B/4B για την μετατροπή των δεδομένων σε σειρές bit που δεν περιλαμβάνουν περισσότερα από δύο ίδια συνεχόμενα bit. Αυτό γίνεται για τον καλύτερο συγχρονισμό μεταξύ πομπού κ δέκτη (διότι δεν έχουν κοινό ρολόι).
5. Στο τελικό σήμα γίνεται διαμόρφωση πλάτους On-Off Keying.

Τα παραπάνω χαρακτηριστικά υλοποιούνται με κώδικα που τρέχει στους επεξεργαστές του CanSat. Λεπτομερής περιγραφή του πρωτοκόλλου υπάρχει στην ενότητα 3.3.



Εικόνα 2. Πρωτόκολλο ασύρματης οπτικής επικοινωνίας

2 ΣΧΕΔΙΑΣΜΟΣ ΕΡΓΟΥ

2.1 Χρονοδιάγραμμα

Χωρίσαμε την ανάπτυξη του project μας σε φάσεις και καθορίσαμε ποια είναι τα παραδοτέα κάθε φάσης, όπως περιγράφεται στο Παράρτημα Γ. Το χρονοδιάγραμμα των εργασιών έχει γίνει σε αυτή τη διαδοχή φάσεων. Κάθε εβδομάδα περιλαμβάνει τουλάχιστον μία δίωρη συνάντηση. Τις ημέρες των σχολικών διακοπών, οι συναντήσεις είναι πολύ περισσότερες. Οι αριθμοί στο πιο κάτω χρονοδιάγραμμα εκφράζουν διάρκεια σε ώρες.

Μήνας	ΟΚΤ				ΝΟΕ				ΔΕΚ				ΙΑΝ				ΦΕΒ				ΜΑΡ				ΑΠΡ			
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
ΟΡΙΣΜΟΣ PROJECT: 5h				1	2	2																						
Αίτηση Συμετοχής				1																								
Χρονοδιάγραμμα				1	1																							
Πλάνο Προώθησης				1	1																							
ΜΕΛΕΤΕΣ ΕΦΙΚΤΟΤ.: 22h				2	5	4	2	1	1	1	1	2	1	1	1	1												
Arduino, BME280				1	1																							
Arduino, LoRa (RX)					1	1																						
Teensy, BMP280				1	1																							
Teensy, Lora (TX)					1	1																						
Teensy, GPS					1	1																						
Teensy, SD						1	1																					
ESP32-CAM							1	1																				
ESP32, Servo									1	1																		
Duplex LoRa Comm											1	1																
Wireless VL Comm												1	1	1	1													
ΕΚΠΑΙΔΕΥΣΗ: 29h	4	3	3	3	2	2	1	1	1	2	2	2		1	1	1												

4. Εντολή διαχείρισης κάμερας GS → CS

Γενική μορφή: αύξων αριθμός, N ή F ή P, αριθμός κάμερας (1 ή 2)

##<sp>N# ή ##<sp>F# ή ##<sp>P#

5. Εγγραφή log file GS εντολής διαχείρισης κάμερας

Γενική μορφή: timestamp, αύξων αριθμός, N ή F ή P, αριθμός κάμερας (1 ή 2)

##:##<sp>##<sp>N# ή ##:##<sp>##<sp>F# ή ##:##<sp>##<sp>P#

6. Εγγραφή log file CS εντολής διαχείρισης κάμερας

Γενική μορφή: timestamp, ληφθείσα συμβολοσειρά

##:##<sp>συμβολοσειρά

3.3.2 Πρωτόκολλο ασύρματης οπτικής επικοινωνίας

Η ασύρματη οπτική επικοινωνία γίνεται μεταξύ των δύο βασικών επεξεργαστών του CanSat (Εικόνα 1 κ 3). Ο κεντρικός επεξεργαστής (Teensy 4.1) λαμβάνει από τον Επίγειο Σταθμό μέσω RF επικοινωνίας εντολή ενεργοποίησης ή απενεργοποίησης της 2ης κάμερας. Την εντολή αυτή την μεταδίδει ασύρματα μέσω οπτικών σημάτων στον 2ο επεξεργαστή του CanSat (ESP32-S3) ο οποίος ελέγχει τις κάμερες.

Σύμφωνα με την θεωρία, οι οπτικές επικοινωνίες χαρακτηρίζονται από πολύ μεγάλες ταχύτητες. Ο φυσικός περιορισμός που αντιμετωπίζουμε στην δική μας περίπτωση είναι η σχετικά αργή λειτουργία των οπτικών στοιχείων που χρησιμοποιούμε και ιδίως της φωτοαντίστασης. Για τον λόγο αυτό, θέτουμε ως αναγκαία συνθήκη την μετάδοση μικρού μεγέθους μηνυμάτων. Αυτή η συνθήκη όμως δεν περιορίζει την γενικότητα του σχεδιασμού μας, καθώς ο ίδιος σχεδιασμός μπορεί να εφαρμοστεί με μεγαλύτερα μηνύματα και ποιοτικότερα (και άρα ταχύτερα) οπτικά υλικά.

Κωδικοποίηση μηνυμάτων

Τα δυνατά μηνύματα που μπορούν να μεταδοθούν μέσω οπτικής επικοινωνίας είναι έξι: έναρξη συνεχούς λήψης εικόνων (N1), τερματισμός συνεχούς λήψης εικόνων (F1), λήψη μίας εικόνας (P1) για την 1η κάμερα, και ομοίως για την 2η κάμερα (N2, F2, P2).

Τα μηνύματα αυτά αναπαριστάνονται αρχικά με κωδικούς μήκους 3 bit, όπως φαίνεται στους παρακάτω πίνακες κωδικοποίησης.

Στις ασύρματες επικοινωνίες είναι επιθυμητή η συχνή εναλλαγή των bit, ώστε να αποφεύγονται οι μεγάλες ακολουθίες ίδιων bit, ιδίως όταν ο πομπός και ο δέκτης δεν λειτουργούν με συγχρονισμένα ρολόγια. Στην δική μας περίπτωση δεν επιθυμούμε να υπάρχουν περισσότερα από 2 συνεχόμενα ίδια bit. Η 3bit κωδικοποίηση δεν μας δίνει αυτή τη δυνατότητα, γιαυτό και θα χρησιμοποιήσουμε [Line Code 3B/4B](#). Θα μετατρέψουμε δηλ. τους 3 bit κωδικούς σε 4 bit κωδικούς.

Η μέθοδος αυτή βασίζεται στο σκεπτικό ότι η χρήση περισσότερων bit παράγει περισσότερους διαθέσιμους κωδικούς, άρα αν οι κωδικοί των x bit δεν επαρκούν (δηλ. δεν υπάρχουν αρκετοί κωδικοί με τα χαρακτηριστικά που επιθυμούμε) τότε οι κωδικοί των x+1 bit πολύ πιθανό να είναι αρκετοί καθώς είναι διπλάσιοι σε πλήθος. Αν και πάλι δεν επαρκούν, προχωράμε σε κωδικούς x+2 bit.

Στην δική μας περίπτωση, η 3bit κωδικοποίηση δίνει έως 8 διαφορετικούς κωδικούς, η προσθήκη 4ου bit αυξάνει τους διαθέσιμους κωδικούς σε 16. Μεταξύ αυτών επιλέγουμε έξι κωδικούς που δεν περιέχουν περισσότερα από 2 συνεχόμενα ίδια bit, για την αναπαράσταση των μηνυμάτων. Επιλέγουμε έναν ακόμη κωδικό για να χρησιμοποιηθεί ως preamble, που θα σηματοδοτεί την αρχή κάθε frame.

Κωδικός (3bit)	Μήνυμα
000	δεν χρησιμοποιείται
001	N1
010	F1

Κωδικός (4bit)	Μήνυμα
0000	απορρίπτεται
0001	απορρίπτεται
0010	N1

011	P1
100	δεν χρησιμοποιείται
101	N2
110	F2
111	P2

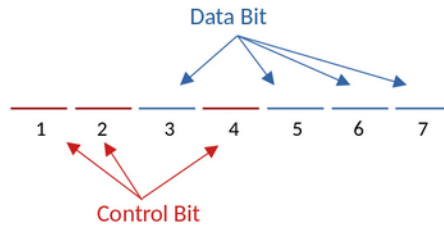
0011	διαθέσιμο
0100	F1
0101	P1
0110	N2
0111	απορρίπτεται
1000	απορρίπτεται
1001	διαθέσιμο
1010	preamble
1011	F2
1100	διαθέσιμο
1101	P2
1110	απορρίπτεται
1111	απορρίπτεται

Πίνακες κωδικοποίησης - Τελικά θα χρησιμοποιηθούν οι 4 bit κωδικοί

Control bits για ανίχνευση λαθών

Κατά την διάδοση πληροφοριών υπάρχει πάντοτε η πιθανότητα να συμβούν λάθη, ανεξάρτητα από το μέσο διάδοσης. Στην δική μας περίπτωση, επιθυμούμε να μπορεί ο δέκτης να ανιχνεύει τα λάθη αν συμβούν. Ως ένα βήμα ακόμη, θέλουμε να μπορεί να διορθώνει τα λάθη χωρίς να ζητά αναμετάδοση των δεδομένων που αλλοιώθηκαν. Για το σκοπό αυτό, υλοποιούμε μια μορφή του κώδικα Hamming, η οποία επιτρέπει την ανίχνευση και υπό συνθήκες τη διόρθωση λαθών στο δέκτη.

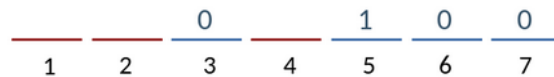
Συγκεκριμένα: μεταξύ των 4 bit που αναπαριστούν ένα μήνυμα, παρεμβάλλονται control bits σε συγκεκριμένες θέσεις και η τιμή τους υπολογίζεται με συγκεκριμένο αλγόριθμο.



Εικόνα 10. Θέσεις control και data bit στην κωδικοποίηση των μηνυμάτων

Εάν αριθμήσουμε τις θέσεις των bit από αριστερά προς τα δεξιά, ξεκινώντας από το 1, τότε τα control bits μπαίνουν στις θέσεις που είναι δυνάμεις του 2 ενώ στις υπόλοιπες θέσεις μπαίνουν με τη σειρά τα data bits. Η δομή αυτή φαίνεται στην Εικόνα 10.

Για παράδειγμα, το μήνυμα F1 που κωδικοποιείται ως 0100 (data bits) θα τοποθετηθεί όπως δείχνει η Εικόνα 11:



Εικόνα 11. Τα data bit του μηνύματος

Οι θέσεις των data bits μπορούν να εκφραστούν ως αθροίσματα δυνάμεων του 2, δηλ. ως αθροίσματα θέσεων control bits. Επομένως, κάθε control bit ελέγχει τις θέσεις data bits τις οποίες εκφράζει. Στην δική μας περίπτωση οι θέσεις των data bits εκφράζονται ως αθροίσματα θέσεων control bit: $3 = 1 + 2$, $5 = 1 + 4$, $6 = 2 + 4$, $7 = 1 + 2 + 4$. Άρα το control bit 1 ελέγχει τα data bits 3, 5, 7. Το control bit 2 ελέγχει τα data bits 3, 6, 7. Το control bit 4 ελέγχει τα data bits 5, 6, 7.

Κάθε control bit τίθεται 0 ή 1 έτσι ώστε να προκύπτει **άρτια ισοτιμία** μεταξύ αυτού και της ομάδας των data bits που ελέγχει. Στο παράδειγμα του μηνύματος F1, τα control bits θα υπολογιστούν όπως φαίνεται στην Εικόνα 12:

<u>1</u>	<u>0</u>	<u>0</u>	<u>1</u>	<u>1</u>	<u>0</u>	<u>0</u>
1	2	3	4	5	6	7

Εικόνα 12. Τα control bit του μηνύματος

Framing

Τα δεδομένα μεταδίδονται σε frames. Δημιουργούμε ένα frame ανά μήνυμα, αφού πρώτα γίνει η προσθήκη των control bits. Την αρχή κάθε frame δείχνει ο κωδικός **1010** (preamble) και το μέγεθος κάθε frame είναι σταθερό, συνολικά 11 bit.

Στο παράδειγμά μας, το μήνυμα F1 με αρχική κωδικοποίηση **0100** (data bits), γίνεται **1001100** μετά την προσθήκη των control bits. Το frame που τελικά θα μεταδοθεί μετά και την προσθήκη του preamble φαίνεται στην Εικόνα 13.

<u>1</u>	<u>0</u>	<u>1</u>	<u>0</u>	<u>1</u>	<u>0</u>	<u>0</u>	<u>1</u>	<u>1</u>	<u>0</u>	<u>0</u>
1	2	3	4	5	6	7	8	9	10	11

Εικόνα 13. Το frame του μηνύματος

Το τέλος του frame αναγνωρίζεται από σταθερό μήκος των 11 bit. Δεν προσθέτουμε κάποια άλλη πληροφορία σε αυτό (πχ κάποιο sequence number), διότι δεν θα γίνονται αναμεταδόσεις.

Ανίχνευση και διόρθωση λαθών

Στην πλευρά του δέκτη αφού αναγνωριστεί η αρχή ενός frame από το preamble, απομονώνονται τα bit του μηνύματος. Για κάθε control bit μαζί με την ομάδα των data bits που ελέγχει, υπολογίζεται ξανά η άρτια ισοτιμία. Αν όλα τα αποτελέσματα είναι 0, τότε γνωρίζουμε ότι δεν συνέβη λάθος κατά την διάδοση του μηνύματος. Αν υπάρχουν μη μηδενικά αποτελέσματα, τότε μπορούμε να συμπεράνουμε (αθροίζοντας του αριθμούς των control bits) ποιο data bit αλλοιώθηκε κατά την διάδοση του μηνύματος, και άρα να το διορθώσουμε. Δεν είναι δυνατή η διόρθωση εάν έχουν αλλοιωθεί περισσότερα του ενός data bits.

Διαμόρφωση και Αποδιαμόρφωση

Το τελευταίο βήμα μετά τη δημιουργία του frame, είναι η μετατροπή του σε αναλογικό οπτικό σήμα. Η διαμόρφωση του οπτικού σήματος που επιλέξαμε να υλοποιήσουμε είναι η On-Off Keying, η οποία είναι διαμόρφωση πλάτους. Το LED θα ανάβει σε όλη την έντασή του όταν πρέπει να μεταδοθεί το 1 και σε σημαντικά μικρότερη (έως μηδενική) ένταση όταν πρέπει να μεταδοθεί το 0.

Όλες οι λειτουργίες του πρωτοκόλλου ασύρματης οπτικής επικοινωνίας που υλοποιούμε φαίνονται στην Εικόνα 2 και περιγράφονται συνοπτικά στην ενότητα 1.2.3.

Σε επίπεδο κώδικα, ο πομπός και ο δέκτης υλοποιούνται σαν μηχανές καταστάσεων (state machines), οι οποίες περνούν διαδοχικά από τις φάσεις του συγχρονισμού, της αναγνώρισης του preamble και της επεξεργασίας των δεδομένων.

3.3.3 Διαδικασίες του συστήματος

Οι διαδικασίες του συνολικού μας συστήματος χωρίζονται σε αυτές που θα γίνονται στο CanSat και σε αυτές που θα γίνονται στον Επίγειο Σταθμό. Πρόκειται για ένα καταναμημένο σύστημα, διότι για να ολοκληρωθεί η αποστολή, απαιτείται να συνεργαστούν προγράμματα που θα τρέχουν σε διαφορετικούς επεξεργαστές.

Στο Παράρτημα Δ υπάρχει η ανάλυση των διαδικασιών καθώς και η περιγραφή κάθε διαδικασίας σε βήματα. Εδώ συμπεριλάβαμε μόνο τα αντίστοιχα διαγράμματα ροής.