



Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης

Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

Εφαρμογές Τηλεπικοινωνιακών Διατάξεων Τεχνική Αναφορά

8^ο εξάμηνο

Θανάσης Τσαρναδέλης 10388

Αλέξανδρος Φωτιάδης 10392

Δημήτρης Χρήστος Κυριάκου 10842

Μαμουγιώργη Μαρία 10533

17 Μαΐ 2024

Πίνακας Περιεχομένων

Πίνακας Περιεχομένων.....	2
Περιγραφή της ιδέας.....	3
Σταθμός βάσης - Δέκτης.....	3
Πομπός.....	3
Γραφικό περιβάλλον στον υπολογιστή - Python.....	4
Υλοποίηση σταθμού βάσης - δέκτη.....	4
Υλοποίηση πομπού.....	6
Υλοποίηση Python.....	8
Επιλογές αισθητήρων.....	10
Ανάλυση τηλεπικοινωνιακής κίνησης.....	10
Ανάλυση αναμενόμενης κάλυψης.....	11
Ανοιχτά Θέματα.....	11

Λίστα Σχημάτων

Σχήμα 1. Διάγραμμα σύνδεσης του σταθμού βάσης.....	5
Σχήμα 2. Διάγραμμα σύνδεσης του πομπού.....	7
Σχήμα 3. Γραφικό περιβάλλον διεπαφής.....	9

Λίστα Φωτογραφιών

Φωτ. 1. Φωτογραφία του υλοποιημένου σταθμού βάσης.....	5
Φωτ. 2. Τα photoresistor εγκατεστημένα στην κάτω πλευρά του χαρτονιού.....	6
Φωτ. 3. Φωτογραφία του υλοποιημένου σταθμού βάσης.....	7
Φωτ. 4. Το RFID reader εγκατεστημένο στην κάτω μεριά του χαρτονιού.....	8

Περιγραφή της ιδέας

Η αρχική ιδέα ήταν να δημιουργηθεί ένα σύστημα διαχείρισης θέσεων πάρκινγκ, χρησιμοποιώντας ως βάση το Arduino, πάνω στο οποίο θα υλοποιηθεί ένα ασύρματο δίκτυο αισθητήρων. Σε κάθε θέση τοποθετείται ένας αισθητήρας, που παρακολουθεί αν η θέση είναι άδεια ή κατειλημμένη. Έτσι, ο διαχειριστής του πάρκινγκ γνωρίζει ακριβώς ποιές θέσεις είναι κατειλημμένες και μπορεί να τις διαχειριστεί καλύτερα, με σκοπό την καλύτερη εξυπηρέτηση των πελατών.

Στην πορεία προστέθηκαν και άλλες λειτουργικότητες, για την πληρότητα του προϊόντος και την καλύτερη εξυπηρέτηση τόσο των πελατών, όσο και του διαχειριστή. Αυτές είναι ο έλεγχος της μπάρας εισόδου, η βοήθεια στον πελάτη κατά την διάρκεια του παρκαρίσματος, καθώς και αναγνώριση του χρήστη μέσω της κάρτας που του έχει δοθεί.

Σταθμός βάσης - Δέκτης

Ο σταθμός βάσης είναι υπεύθυνος να εκτελεί τα εξής:

- Έλεγχος μπάρας εισόδου. Ο σταθμός βάσης ανιχνεύει αν υπάρχει αυτοκίνητο που επιθυμεί να μπει στο πάρκινγκ και σηκώνει την μπάρα εισόδου. Αφού το αυτοκίνητο περάσει, κατεβάζει την μπάρα.
- Συγκέντρωση των δεδομένων από τους πομπούς. Ο κάθε πομπός στέλνει στον σταθμό βάσης την κατάσταση της θέσης του (άδεια θέση ή το ID του χρήστη που πάρκαραν αν είναι κατειλημμένη). Ο δέκτης συγκεντρώνει αυτή την πληροφορία και την μορφοποιεί κατάλληλα.
- Επικοινωνία με τον υπολογιστή και μετάδοση των δεδομένων που δέχτηκε από τους πομπούς στον υπολογιστή για επεξεργασία και απεικόνιση. Αφού τα δεδομένα έρθουν στην σωστή μορφή, το Arduino στέλνει τα δεδομένα στον υπολογιστή για περαιτέρω επεξεργασία και απεικόνιση.

Από τα παραπάνω είναι προφανές ότι το σταθμός βάσης είναι καλό, αλλά όχι απαραίτητο, να εγκατασταθεί κοντά στον μηχανισμό της μπάρας, και κοντά στον υπολογιστή που τρέχει το γραφικό περιβάλλον.

Πομπός

Ο πομπός είναι υπεύθυνος να εκτελεί τα εξής:

- Αναγνώριση κατάστασης θέσης. Ο κάθε πομπός χρησιμοποιεί έναν αισθητήρα απόστασης (θα αναλυθεί αργότερα γιατί επιλέχθηκε ο συγκεκριμένος αισθητήρας απόστασης και όχι κάποιος άλλος αισθητήρας απόστασης ή άλλου μεγέθους) για να αναγνωρίσει αν η θέση του είναι κατειλημμένη ή όχι.

- Αναγνώριση του χρήστη που πάρκαρει. Μέσω της κάρτας RFID, όταν κάποιος χρήστης παρκάρει, σκανάρει την κάρτα του στον αναγνώστη τοποθετημένο στην θέση πάρκινγκ για να αναγνωριστεί το ID του. Η κάρτα ή κάποιο αντίστοιχο RFID tag μπορεί να εγκατασταθεί και στο όχημα (όμοια με το demo) για την επίτευξη του ίδιου στόχου.
- Εκπομπή της κατάστασης της θέσης και του ID του χρήστη. Αν η θέση είναι άδεια, τότε εκπέμπεται E (empty), αλλιώς εκπέμπεται το ID του χρήστη. Έτσι, ο σταθμός βάσης ξέρει την κατάσταση και το ID του χρήστη που πάρκαρει.
- Βοήθεια κατά το παρκάρισμα. Όμοια με το σύστημα υποβοήθησης παρκαρίσματος που είναι εγκατεστημένο σε πολλά οχήματα, όταν ένα όχημα εισέρχεται στην θέση ακούγεται ένα επαναλαμβανόμενο buzzer. Το τελευταίο γίνεται εντονότερο όσο το όχημα εισέρχεται περισσότερο στην θέση. Αν το όχημα είναι πλέον πλήρως παρκαρισμένο, το buzzer ηχεί συνεχόμενα και στην συνέχεια απενεργοποιείται, δηλώνοντας στον χρήστη ότι πρέπει να σταματήσει.

Από τα παραπάνω είναι προφανές πως για την εύρυθμη λειτουργία χρειαζόμαστε έναν πομπό για κάθε θέση.

Γραφικό περιβάλλον στον υπολογιστή - Python

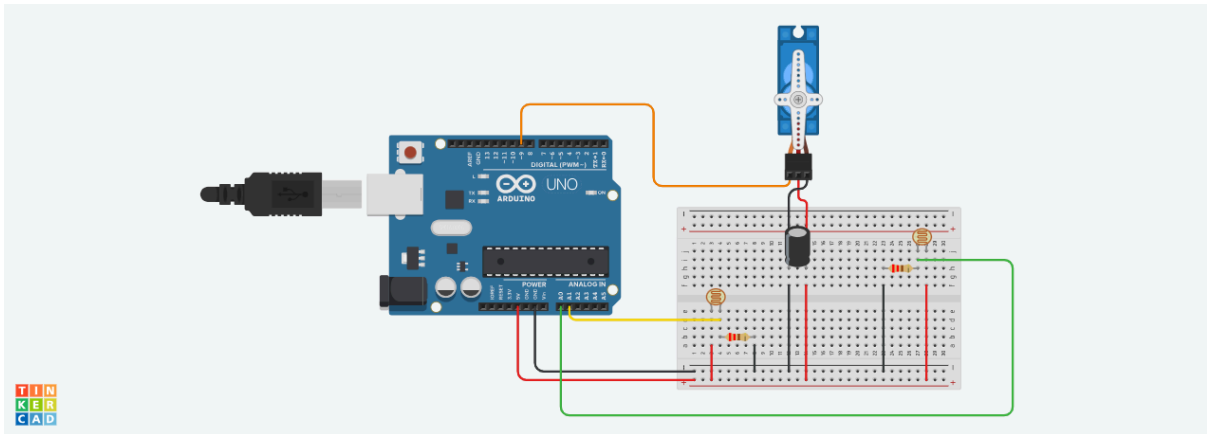
Ο υπολογιστής είναι υπεύθυνος για τα εξής:

- Παραλαβή των δεδομένων από τον σταθμό βάσης. Αυτό υλοποιείται μέσω USB, με ενσύρματη διασύνδεση τους.
- Επεξεργασία δεδομένων. Με βάση τα δεδομένα που έλαβε, αν η θέση είναι κατειλημμένη και χρησιμοποιώντας το ID του χρήστη, τον αναζητώ στην βάση δεδομένων. Θεωρώντας ότι ο χρήστης έχει κάνει εγγραφή γνωρίζω την πινακίδα του οχήματος και τον τύπο του οχήματος και τα τυπώνω στην οθόνη. Αν το ID δεν υπάρχει την βάση τυπώνω “UNKNOWN”.
- Γραφικό περιβάλλον - GUI. Για κάθε θέση (τέσσερις στο συγκεκριμένο demo) τυπώνω αν είναι “Empty” ή “Full”. Στην δεύτερη περίπτωση, τυπώνω την πινακίδα και τον τύπο του οχήματος αν είναι γνωστά, αλλιώς τυπώνω “UNKNOWN”.

Υλοποίηση σταθμού βάσης - δέκτη

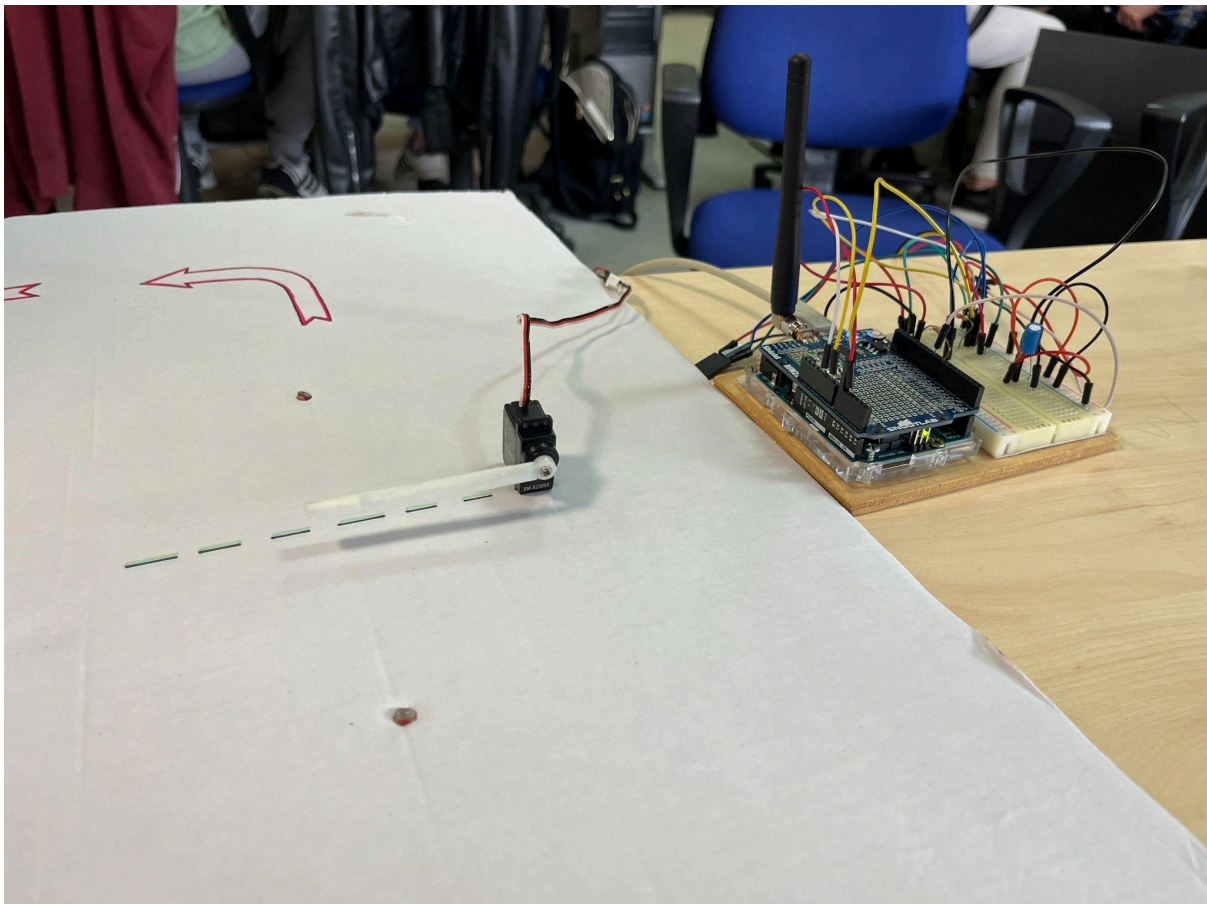
Ο σταθμός βάσης αποτελείται από:

- Arduino Uno
- SparkFun RFM22 Shield - 434MHz (ή αντίστοιχη της Snootlab)
- Πυκνωτής 100μF
- 2 Αντιστάσεις 220Ω
- Μοτέρ Servo

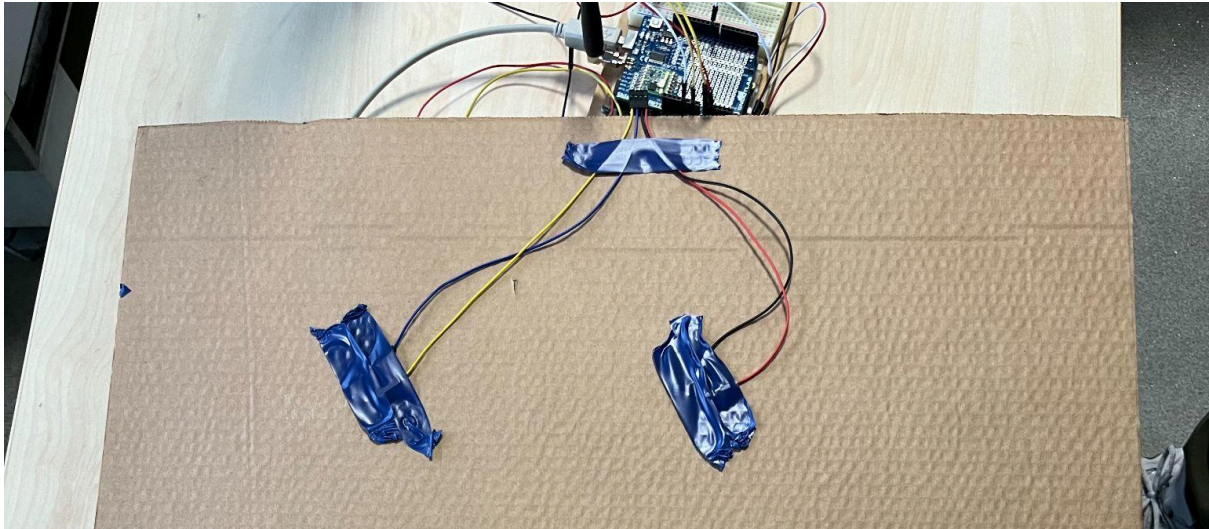


Σχήμα 1: Διάγραμμα σύνδεσης του σταθμού βάσης

Στο σχήμα δεν φαίνεται η ασύρματη κάρτα, η οποία είναι σε μορφή hat.



Φωτ. 1: Φωτογραφία του υλοποιημένου σταθμού βάσης



Φωτ. 2: Τα photoresistor εγκατεστημένα στην κάτω πλευρά του χαρτονιού.

Όταν το photoresistor πριν την μπάρα είναι σκοτεινό, τότε σημαίνει ότι υπάρχει αυτοκίνητο στην είσοδο που περιμένει να μπει, και η μπάρα ανοίγει. Αν το φωτορεσίστορ μετά την μπάρα είναι σκοτεινό, τότε σημαίνει ότι το αυτοκίνητο έχει περάσει την μπάρα. Μετά το πέρας 1.5 δευτερολέπτων η μπάρα κλείνει.

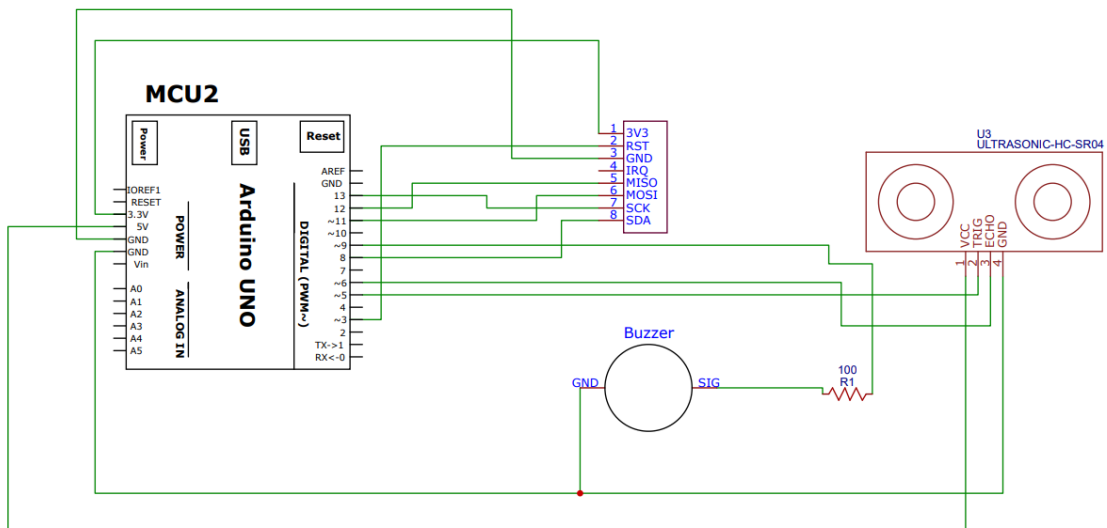
Ο δέκτης δέχεται από τους πομπούς είτε τον χαρακτήρα "E", αν η θέση είναι κενή, είτε τους χαρακτήρες του ID της RFID κάρτας πχ. " 53 03 31 0f". Γνωρίζοντας ποιος δέκτης αντιστοιχεί σε κάθε θέση, και τα δεδομένα του, ο σταθμός βάσης γράφει στην σειριακή χαρακτήρες της μορφής: "S1:E,S2:E,S3: 53 03 31 0f,S4:E,end", δεδομένα που είναι σε κατάλληλη μορφή για επεξεργασία από την Python.

Για την ασύρματη κάρτα χρησιμοποιούνται οι βιβλιοθήκες RF22 και RF22Router, ενώ για τον σερβοκινητήρα η βιβλιοθήκη Servo.

Υλοποίηση πομπού

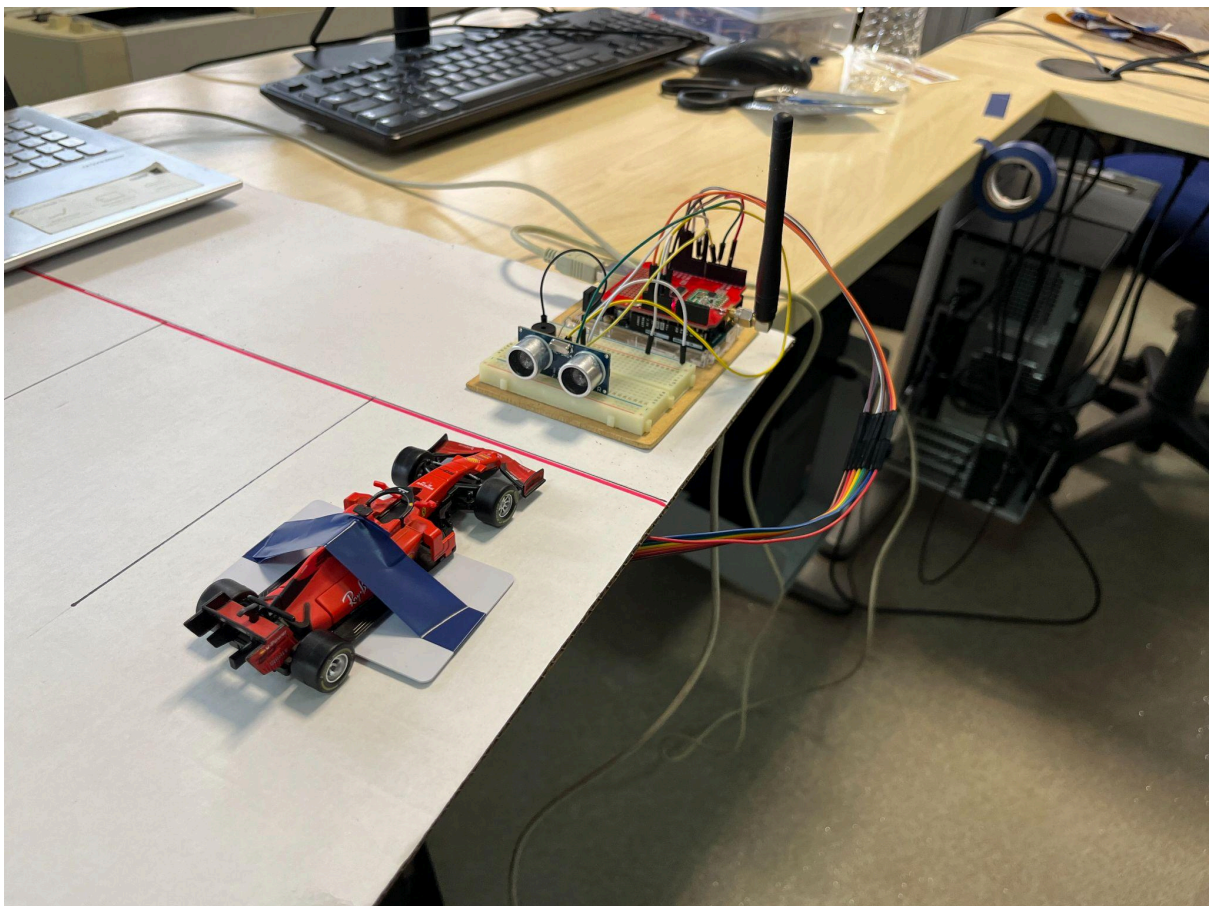
Ο πομπός αποτελείται από:

- Arduino Uno
- SparkFun RFM22 Shield - 434MHz (υποχρεωτικά)
- Buzzer
- Αντίσταση 100Ω
- Αισθητήρας απόστασης ultrasonic HC-SR04
- RFID Mifare RC522 μαζί με μια κάρτα RFID στα 13.56MHz

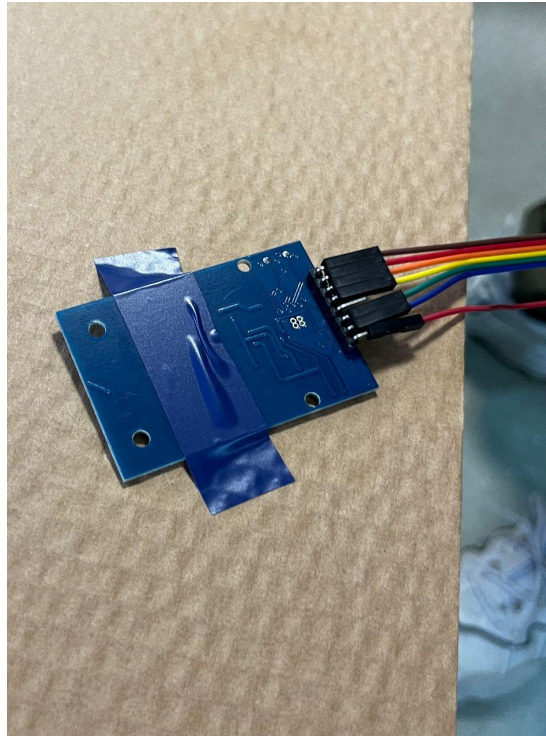


Σχήμα 2: Διάγραμμα σύνδεσης του πομπού

Το module δεξιά του Arduino είναι το RFID reader.



Φωτ. 3: Φωτογραφία του υλοποιημένου σταθμού βάσης. Φαίνεται επίσης το δοκιμαστικό όχημα που φέρει την κάρτα RFID.



Φωτ. 4: Το RFID reader εγκατεστημένο στην κάτω μεριά του χαρτονιού.

Ο πομπός χρησιμοποιεί τον αισθητήρα απόστασης για να προσδιορίσει αν η θέση είναι κενή, αν υπάρχει αυτοκίνητο που παρκάρει εκείνη την στιγμή ή αν υπάρχει πλήρως παρκαρισμένο αυτοκίνητο. Αν η έξοδος του αισθητήρα απόστασης είναι μεγαλύτερη του μήκους της θέσης τότε είναι κενή, και εκπέμπεται προς τον σταθμό βάσης ο χαρακτήρας “E” (empty). Αν η έξοδος του αισθητήρα απόστασης είναι περίπου στην μέση του μήκους της θέσης, τότε ένα όχημα προσπαθεί να παρκάρει. Το buzzer ηχεί διακεκομμένα, έως ότου το αυτοκίνητο εισέλθει πλήρως. Όταν γίνει το τελευταίο, το buzzer ηχεί συνεχόμενα για 1.5 δευτερόλεπτο και μετά απενεργοποιείται. Στην συνέχεια ενεργοποιείται ο αναγνώστης RFID, και περιμένει να διαβάσει την κάρτα ή το tag του πελάτη. Αυτό γίνεται είτε χειροκίνητα, με τον πελάτη να τοποθετεί την κάρτα του στον αναγνώστη, είτε αυτόματα, όπως στο βίντεο του demo. Αν η κάρτα τοποθετηθεί στο αυτοκίνητο, τότε η ανάγνωση γίνεται αυτόματα και το μόνο που έχει να κάνει ο πελάτης είναι να παρκάρει. Όταν αναγνωστεί η κάρτα, εκπέμπεται προς το σταθμό βάσης το ID της κάρτας. Το ID συνεχίζει να εκπέμπεται έως ότου το όχημα ξεπαρκάρει.

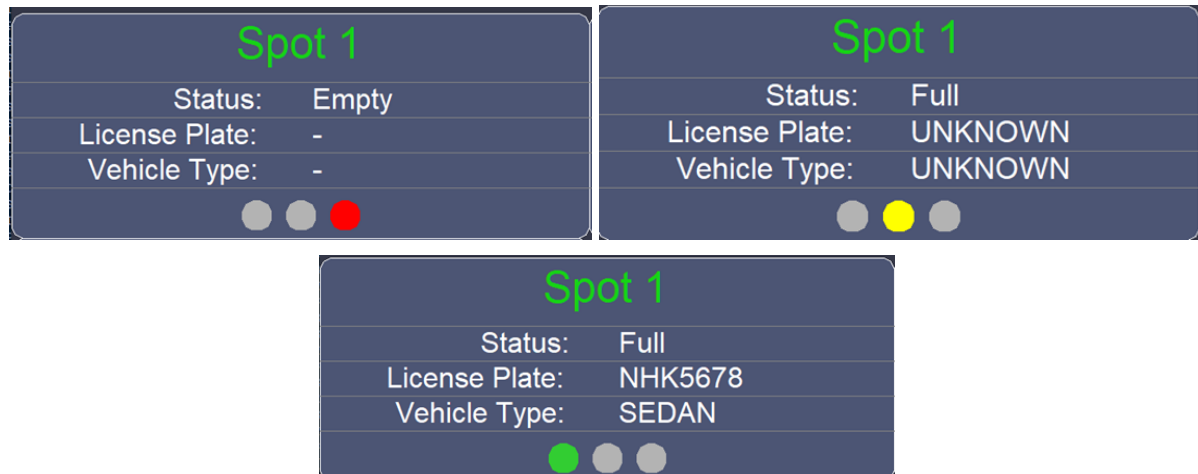
Για την ασύρματη κάρτα χρησιμοποιούνται οι βιβλιοθήκες RF22 και RF22Router, για το RFID η βιβλιοθήκη MFRC522 και η βιβλιοθήκη SPI, που χρησιμοποιείται και από τα δύο modules.

Υλοποίηση Python

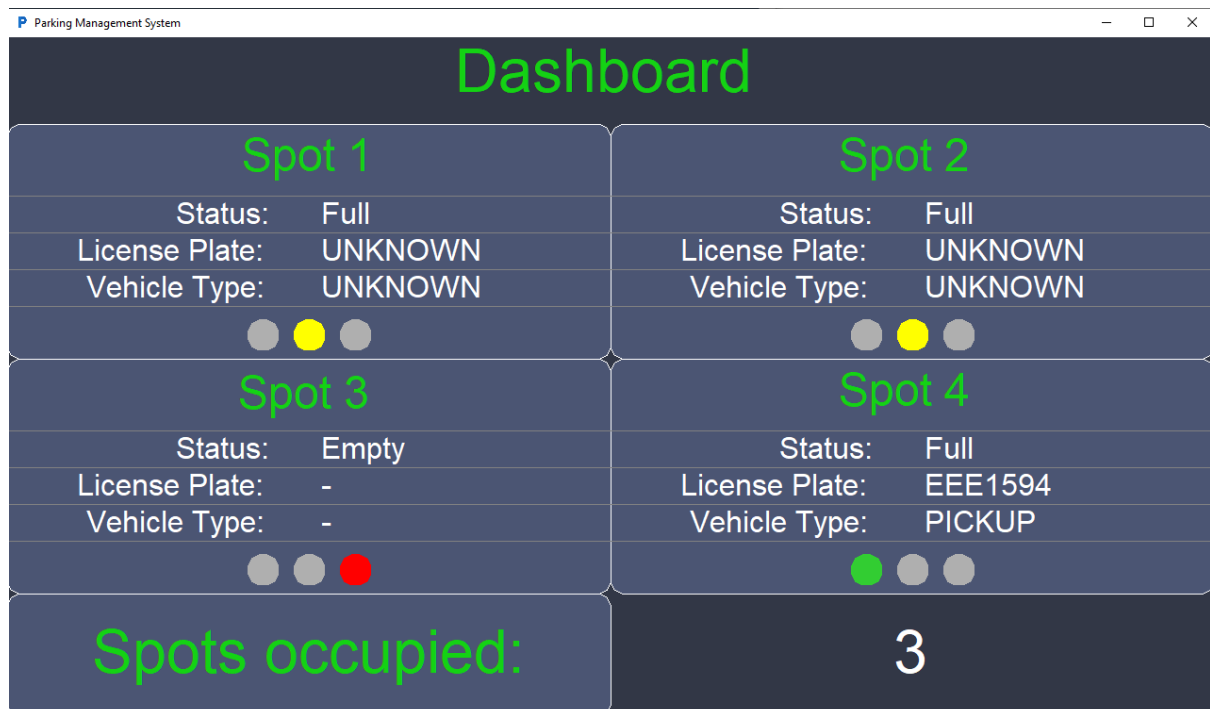
Η python παίρνει τα δεδομένα στην μορφή “S1:E,S2:E,S3: 53 03 31 0f,S4:E,end” και χωρίζει τα δεδομένα της κάθε θέσης. Υπάρχουν αποθηκευμένα τα δεδομένα των θέσεων στην προηγούμενη λήψη του πομπού. Αν κάποια θέση αλλάξει κατάσταση,

ανανεώνεται η διεπαφή χρήστη. Αν η θέση είναι άδεια, εμφανίζεται Status:Empty, License Plate:-, Vehicle Type:-. Αν η θέση είναι γεμάτη, γνωρίζω το ID του χρήστη και αναζητώ το ID του στην βάση δεδομένων του συστήματος. Αν ο πελάτης έχει κάνει εγγραφή, τότε γνωρίζω τα στοιχεία του και τα δείχνω στην οθόνη. Αν ο χρήστης δεν έχει κάνει εγγραφή, τότε δεν γνωρίζω τα στοιχεία του και εμφανίζεται στην οθόνη UNKNOWN.

Για την επεξεργασία και αποθήκευση των δεδομένων χρησιμοποιείται η βιβλιοθήκη Numpy, για την επικοινωνία με το Arduino η Pyserial, και για το γραφικό περιβάλλον η Tkinter.



Τρεις πιθανές καταστάσεις της θέσης.



Σχήμα 3: Γραφικό περιβάλλον διεπαφής.

Επιλογές αισθητήρων και ασύρματης κάρτας

- HC-SR04: Ο συγκεκριμένος αισθητήρας επιλέχθηκε έναντι άλλων διαθέσιμων, καθώς δίνει πληροφορίες απόστασης και όχι απλά αν η θέση είναι κατειλημμένη ή όχι. Έτσι μπορώ να υλοποιήσω το σύστημα υποβοήθησης παρκαρίσματος.
- SparkFun RFM22 Shield - 434MHz: Αυτή η ασύρματη κάρτα (κόκκινο χρώμα) πρέπει να χρησιμοποιηθεί υποχρεωτικά στον πομπό, καθώς η άλλη διαθέσιμη (Snootlab χρώματος μπλε) δεν λειτουργεί σε συνδυασμό με το RFID reader. Για τον δέκτη, μπορεί να χρησιμοποιηθεί οποιαδήποτε από τις δύο επιλογές.
- RFID Mifare MFRC522: Το RFID module επιλέχτηκε καθώς το είχαμε διαθέσιμο, με σκοπό την ταυτοποίηση του χρήστη. Σε περίπτωση που υπάρχει κάποιο άλλο module που υλοποιεί την ίδια δουλειά, μπορεί να χρησιμοποιηθεί στην θέση του RFID.

Ανάλυση τηλεπικοινωνιακής κίνησης

Αναλύοντας τον κωδικά του Arduino, εξάγω τα εξής δεδομένα για να κάνω ανάλυση κίνησης:

- Τα δεδομένα που στέλνω είναι οι χαρακτήρες του ID της κάρτας αυξημένο κατά ένα. Συνολικά αυτά είναι $12+1=13$ χαρακτήρες(συμπεριλαμβανομένου και των κενών χαρακτήρων) από 8 bit/χαρακτήρα, συνολικά 104 bits
- Χρησιμοποιώ διαμόρφωση GFSK με 125kbps Bandwidth
- Στέλνω δεδομένα ανά 200msec άρα $\lambda=1/200\text{msec}=5$ πακέτα/sec

Μπορώ έτσι να υπολογίσω την μέση διάρκεια πακέτου $T=104 \text{ bits}/125\text{kbps}=0.832\text{msec}$. Για το συγκεκριμένο παράδειγμα θεωρώ ότι έχω 4 πομπούς άρα $k=4$ και άρα $r=kl=20$ πακέτα/sec. Τελικά το $rT=20*0.832=0.01664$ πακέτα. Από τους πίνακες και τους τύπους βρίσκω πως $RT=0.01692$ Erl. Γνωρίζοντας ότι χρησιμοποιώ ALOHA, ο κάθε κόμβος παράγει κίνηση $\lambda T=5*0.832=0.00461$ πακέτα. Άρα με μοναδικό σταθμό βάσης θα μπορούσα να υποστηρίξω το πολύ $0.186/0.00461=44.7$, δηλαδή 44 πομπούς. Συγκεντρωτικά:

Μέγεθος	Τιμή
Πρωτόκολλο επικοινωνίας	ALOHA
Μέγεθος πακέτου	104 bit
Εύρος ζώνης	125 kbps
Διάρκεια πακέτου T	0.832 msec
Ρυθμός παραγωγής πακέτων ανά πομπό λ	5 πακέτα/sec

Αριθμός Πομπών κ	4
Συνολικός ρυθμός παραγωγής πακέτων r	20
Channel Utilisation rT	0.01664 πακέτα
Αληθινή μέση κίνηση RT	0.01692 Erlang
Παραγωγή κίνησης ανα κόμβο	0.00416 πακέτα
Μέγιστος αριθμός πομπών που υποστηρίζει το σύστημα ανα σταθμό βάσης.	44

Ανάλυση αναμενόμενης κάλυψης

Από τον κώδικα του Arduino και το datasheet της ασύρματης κάρτας RF22 εξάγω το Psensitivity και την ισχύ εκπομπής Ptx, και χρησιμοποιώ τον τύπο του Friis για να υπολογίσω την αναμενόμενη απόσταση κάλυψης. Θεωρώ τους συντελεστές Gt και Gr ίσους με 1, δηλαδή τις κεραίες ιστροπικές. Επίσης θεωρώ ότι η ισχύς αποσβένει συναρτήσει της 4ης δύναμης της απόστασης, και χρησιμοποιώ το υπολογισμένο λ από την παράγραφο Ανάλυση τηλεπικοινωνιακής κίνησης. Συγκεντρωτικά:

Μέγεθος	Τιμή
Psensitivity (από datasheet)	-118 dbm
Ptx (από κώδικα)	20 dbm
λ	5
Gt	1 (ιστροπική)
Gr	1 (ιστροπική)
n	4
Απόσταση κάλυψης	1777 m

Συμπεραίνω ότι η απόσταση κάλυψης είναι εξαιρετική. Πάραυτα, η παραπάνω είναι ιδανική περίπτωση και σε πραγματική εφαρμογή αναμένω η απόσταση αυτή να έχει μικρή ελάττωση. Σε κάθε περίπτωση, ο αριθμός των πομπών που μπορεί να υποστηρίξει ένας δέκτης είναι πιο περιοριστικό μέγεθος από την απόσταση κάλυψης. Θεωρώντας μια τυπική θέση πάρκινγκ $5 \times 2.30 \text{ m} = 11.5 \text{ m}^2$, με μέγιστο αριθμό πομπών ανά δέκτη ίσο με 44, έχω εμβαδόν 506 m^2 , ή περίπου $220 \times 2.30 \text{ m}$, μέγεθος πολύ μικρότερο της αναμενόμενης απόστασης κάλυψης.