

CPS

CYBER-PHYSICAL
SYSTEMS



Πτυχιακή Εργασία

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΛΟΠΟΝΝΗΣΟΥ

ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ

ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΗΛΕΚΤΡΟΝΙΚΩΝ
ΥΠΟΛΟΓΙΣΤΩΝ

“ Μελέτη και Ανάπτυξη Cyber-Physical συστημάτων προσέγγισης στόχων από Ρομποτική πλατφόρμα μέσω BLE αισθητήρων και Διαχείριση Αντικειμένων μέσω Ενσωματωμένου Βραχίονα ”

Θεόδωρος-Βασίλειος Σκανδάμης

A.M. : 2317

Εισηγητής :

Επίκουρος Καθ. Κος Χρήστος Αντωνόπουλος

Πάτρα 2022



- © Θεόδωρος – Βασίλειος Σκανδάμης
- © Πανεπιστήμιο Πελοποννήσου – Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών
- © Εργαστήριο Σχεδιασμού Ενσωματωμένων Συστημάτων & Εφαρμογών

“Μελέτη και Ανάπτυξη Cyber-Physical συστημάτων προσέγγισης στόχων από Ρομποτική πλατφόρμα μέσω BLE αισθητήρων και Διαχείριση Αντικειμένων μέσω Ενσωματωμένου Βραχίονα”

“Study and Development of Cyber-Physical Target Approach Systems from Robotic Platform via BLE Sensors and Object Management via Embedded Arm”

ΕΞΕΤΑΣΤΙΚΗ ΕΠΙΤΡΟΠΗ

Βώρος Νικόλαος

Αντωνόπουλος Χρήστος

Καθηγητής

Επ. Καθηγητής



Πίνακας περιεχομένων

Πρόλογος	
Εισαγωγή	
1. Ρομποτικό Περιβάλλον	
1.1 <i>Ρομποτική Πλατφόρμα</i>	
1.1.1 <i>Kobuki Base platform</i>	
1.1.2 <i>Intel NUC PC</i>	
1.1.3 <i>Laser Scanner</i>	
1.1.4 <i>Orbbec Astra Camera</i>	10
1.1.5 <i>WidowX Robotic Arm</i>	10
1.2 <i>Ρομποτικό Λογισμικό Robot Operating System</i>	11
1.2.1 <i>Επίπεδα Robot Operating System</i>	13
1.2.2 <i>Επίπεδο συστήματος Αρχείων</i>	14
1.2.3 <i>Επίπεδο γραφικής παράστασης υπολογισμού</i>	14
1.2.4 <i>Επίπεδο κοινότητας</i>	14
2. Εξομοιωτής Σεναρίων	15
2.1 <i>Ρομποτική προσομοίωση</i>	15
2.2 <i>Προσομοίωση Ρομποτικών συστημάτων</i>	15
2.3 <i>Προσομοιωτής Ρομποτικών συστημάτων Gazebo</i>	15
3. Τεχνολογία Bluetooth Low Energy	17
3.1 <i>Bluetooth</i>	17
3.2 <i>Bluetooth Low Energy</i>	18
3.3 <i>Αρχιτεκτονική Bluetooth</i>	19
3.3.1 <i>Δίκτυο Piconet</i>	19



3.3.2	Δίκτυο Scatternet.....	20
3.4	Δομικά στοιχεία στοίβας Bluetooth.....	20
3.4.2	Host.....	21
3.4.2.1	Logical Link Control and Adaptation Protocol (L2CAP).....	21
3.4.2.2	Attribute Protocol (ATT).....	22
3.4.2.3	Security Manager (SM).....	22
3.4.2.4	Generic Attribute Profile (GATT).....	24
3.4.2.5	Generic Access Profile (GAP).....	24
3.4.3	Controller.....	24
3.4.3.1	Host Controller Interface (HCI).....	25
3.4.3.2	Link layer.....	26
3.5	Physical Layer.....	27
3.6	Ρόλοι συσκευών Bluetooth.....	28
3.7	Συσκευές Bluetooth.....	28
3.7.1	CC2650 SensorTag TI.....	29
3.7.2	ESP32 -Thing Platform - SparkFun.....	30
3.8	MQTT Πρωτόκολλο.....	32
3.8.1	Τι είναι το πρωτόκολλο MQTT.....	32
3.8.2	Χαρακτηριστικά πρωτοκόλλου MQTT.....	32
3.8.3	Δομή Μηνυμάτων (Πακέτων Ελέγχου) MQTT.....	33
3.8.4	Ασφάλεια Επικοινωνίας MQTT.....	34
3.8.5	MQTT Broker.....	35
3.9	ATLAS – ATLAS Gateway.....	35
4	Αρχιτεκτονική και Υλοποίηση.....	37
5	Υλοποίηση της Αρχιτεκτονικής.....	37
5.1	Υλοποίηση σε επίπεδο ROS.....	37
5.2	Αποστολή πληροφορίας.....	39
5.3	Localization api.....	40
5.3.1	Προσθήκη του βραχίονα.....	41
6	Μελλοντική εργασία.....	42



Πρόλογος

Η παρούσα πτυχιακή εργασία με θέμα “Μελέτη και Ανάπτυξη Cyberphysical συστημάτων προσέγγισης στόχων από Ρομποτική πλατφόρμα μέσω BLE αισθητήρων και Διαχείριση Αντικειμένων μέσω Ενσωματωμένου Βραχίονα”, εκπονήθηκε στα πλαίσια του προπτυχιακού προγράμματος σπουδών του πρώην τμήματος Μηχανικών Πληροφορικής ΤΕ, του Τεχνολογικού Εκπαιδευτικού Ιδρύματος Δυτικής Ελλάδας.

Σε αυτό το σημείο, θα ήθελα να ευχαριστήσω τον καθηγητή και μέντορά μου, κο. Αντωνόπουλο Χρήστο, για την πολυτιμότερη βοήθεια και υποστήριξη που μου παρείχε.

Λέξεις Κλειδιά: BLE, ROS, αισθητήρες BLE, Turtlebot2, Gazebo, Cloud, Localization





Εισαγωγή

Στόχος της παρούσας πτυχιακής εργασίας ήταν να μελετηθεί η επικοινωνία της ρομποτικής πλατφόρμας «Turtlebot 2» με την τεχνολογία Bluetooth Low Energy (BLE).

Για την επίτευξη της πτυχιακής εργασίας, θα πραγματοποιηθούν κατάλληλες βελτιστοποιήσεις στο μεταλειτουργικό σύστημα ROS του ρομπότ, έτσι ώστε να είναι πιο γρήγορο στην εκκίνηση, καθώς και στη λειτουργία του. Σε επόμενο στάδιο, θα πραγματοποιηθεί μια λειτουργική αναβάθμιση σε επίπεδο υλικού του ρομπότ, προσθέτοντας σε αυτό έναν ρομποτικό βραχίονα με σκοπό τη συλλογή αντικειμένων από έναν συγκεκριμένο στόχο που θα του έχει ορίσει ο χρήστης.

Για την επίτευξη του σκοπού αυτού, θα εμπλουτιστεί η ρομποτική πλατφόρμα τόσο από πλευράς hardware, όσο και από πλευράς Software (πέραν του βασικού).

Η υλοποίηση βασίστηκε στην επικοινωνία που εγκαθιδρύθηκε μεταξύ των συσκευών BLE και της ρομποτικής πλατφόρμας, όπου με κατάλληλα σχεδιασμένους αλγορίθμους, εντοπίζεται ο στόχος (συσκευή BLE), έτσι ώστε η ρομποτική πλατφόρμα να τον αναγνωρίζει και να πλησιάζει σε αυτόν. Η ρομποτική πλατφόρμα από τη δική της πλευρά, διαθέτει τους κατάλληλους αλγορίθμους ώστε να μπορεί να σχηματίζει την αποτύπωση του χώρου και να τον αποθηκεύει σε μορφή χάρτη έτσι ώστε να μπορεί να πλοηγηθεί αυτόνομα εντός του χαρτογραφημένου χώρου. Αλγόριθμοι κατάλληλοι υλοποιήθηκαν, έτσι ώστε ο ρομποτικός βραχίονας που θα είναι τοποθετημένος στο ανώτερο επίπεδο του ρομπότ, να μπορεί να εκτελέσει τη λειτουργία σύλληψης σταθερών αντικειμένων και την διαχείριση αυτών, με αυτοματοποιημένο τρόπο.

Οι δοκιμές έγιναν αρχικά στον προσομοιωτή Gazebo και ακολούθως (για την επιβεβαίωση της ορθής λειτουργίας των αλγορίθμων), στο πρότυπο περιβάλλον της

οικίας αυτόνομης διαβίωσης του εργαστηρίου σχεδιασμού ενσωματωμένων συστημάτων και εφαρμογών του τμήματος Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών, του Πανεπιστημίου Πελοποννήσου.

1. Ρομποτικό Περιβάλλον

1.1 Ρομποτική Πλατφόρμα

Για την προτεινόμενη λύση χρησιμοποιείται η ρομποτική πλατφόρμα τύπου **Turtlebot2** [1] (Εικόνα 1). Το Turtlebot είναι ένα χαμηλού κόστους Robot για ερευνητική χρήση το οποίο υποστηρίζει ανοιχτού τύπου λογισμικό, με το οποίο μπορεί να περιηγηθεί σε έναν χώρο, να δει τον κόσμο μέσω 3D κάμερας και να χρησιμοποιηθεί σε διάφορες ενδιαφέρουσες εφαρμογές.

Το ρομπότ είναι μια συλλογή στοιχείων υλικού, όπου ενορχηστρώνεται από το λογισμικό. Σε επίπεδο υλικού, η βάση του ρομπότ ενσωματώνει τους τροχούς, τις μπαταρίες, καθώς και το σύστημα ελέγχου κίνησης. Η κινητήρια δύναμη του Turtlebot είναι το Kobuki της εταιρίας Yujin Robot. Από την πλευρά επιπλέον αισθητηρίων και λοιπού υλικού, διαθέτει μια υπολογιστική μονάδα τύπου INTEL NUC, έναν αισθητήρα Laser Scanner τύπου Kobuki, έναν τρισδιάστατο οπτικό αισθητήρα τύπου Orbbec Astra, καθώς και έναν ρομποτικό βραχίονα WidowX Arm.



Εικόνα 1. Ρομποτική πλατφόρμα Turtlebot2



1.1.1 Kobuki Base platform

Το Kobuki [2] Το Kobuki είναι μια ρομποτική ερευνητική βάση χαμηλού κόστους όπου προσφέρει κίνηση, σχεδιασμένη για εκπαίδευση και έρευνα πάνω στην τελευταία λέξη της ρομποτικής. Παρέχει τροφοδοσία ρεύματος για έναν εξωτερικό υπολογιστή, καθώς και πρόσθετους αισθητήρες και ενεργοποιητές. Η εξαιρετικά ακριβής οδομετρία του από το εργοστασιακά βαθμονομημένο γυροσκόπιο, επιτρέπει την ακριβή πλοήγηση.

Τα λειτουργικά χαρακτηριστικά όπου διαθέτει, είναι τα εξής:

- Μέγιστη μεταφορική ταχύτητα: 70 cm/s
- Μέγιστη ταχύτητα περιστροφής: 180 deg/s
- Ωφέλιμο φορτίο: 5 kg (σε σκληρό δάπεδο), 4 kg (σε χαλί)
- Σκαλοπάτι: Μπορεί να κατέβει από σκαλοπάτι με βάθος μικρότερο από 5 cm.
- Αναρρίχηση κατωφλίων: αναρριχάται σε κατώφλια 12 mm ή χαμηλότερα.
- Αναρρίχηση χαλιών: αναρριχάται σε χαλιά 12 mm ή χαμηλότερα
- Αναμενόμενος χρόνος λειτουργίας: 3/7 ώρες (μικρή/μεγάλη μπαταρία)
- Αναμενόμενος χρόνος φόρτισης: 1,5/2,6 ώρες (μικρή/μεγάλη μπαταρία)

Τα χαρακτηριστικά του από πλευράς Hardware είναι:

- Ανίχνευση υπερφόρτωσης κινητήρα: απενεργοποιεί την τροφοδοσία κατά την ανίχνευση υψηλού ρεύματος (>3A)
- Οδομέτρηση: 2578,33 ticks/wheel
- Γυροσκόπιο: εργοστασιακή βαθμονόμηση, 1 άξονας (110 μοίρες/δευτερόλεπτο)
- Αισθητήρες κλίσης: αριστερά, κέντρο, δεξιά
- Υποδοχές τροφοδοσίας: 5V/1A, 12V/1.5A, 12V/5A
- Ήχος: διάφορες προγραμματιζόμενες ακολουθίες ηχητικών σημάτων
- Κουμπιά: 3 x κουμπιά αφής
- Μπαταρία: (4S1P - μικρή), 4400 mAh (4S2P - μεγάλη).
- Αναβάθμιση υλικολογισμικού: μέσω usb
- Ρυθμός δεδομένων αισθητήρα: 50Hz
- Δέκτης IR σύνδεσης: αριστερά, κέντρο, δεξιά
- Διάμετρος: 351.5mm / Ύψος: 124.8mm / Βάρος: 2.35kg (4S1P - μικρό)

Επίσης, παρέχει και επιπλέον αισθητήρες και ενεργοποιητές (actuators), καθώς και οδομετρία (μέσω γυροσκοπίου), κάτι το οποίο οδηγεί σε ακριβή πλοήγηση του ρομπότ μέσα στον χώρο. Περιλαμβάνει επίσης μια βάση φόρτισης δαπέδου όπου μέσω του εσωτερικού ελέγχου του επιπέδου της εναπομένουσας ενέργειας και μέσω του χάρτη, ο



οποίος αποθηκεύεται από τη χαρτογράφηση, το ρομπότ προσεγγίζει αυτόνομα στη βάση του μέσω των IR αισθητήρων του.

1.1.2 Intel NUC PC

Καθώς προορίζεται για συνεχή λειτουργία, το Kobuki παρέχει τροφοδοσίες για εξωτερικό προσωπικό υπολογιστή τύπου IntelNUC [3], ο οποίος είναι υπεύθυνος για την υλοποίηση των προγραμμάτων και των δυνατοτήτων δικτύωσης του ρομπότ.

Το Mini PC NUC (NUC5CPYH) από την Intel διαθέτει έναν διπύρηνο επεξεργαστή Intel Celeron 1,6 GHz, ο οποίος μπορεί να φτάσει έως και τα 2,16 GHz με Intel Turbo Boost. Διαθέτει μία κενή υποδοχή SODIMM που μπορεί να δεχτεί έως και 8 GB μνήμης RAM DDR3L 1333/1600 MHz. Για αποθήκευση, το Mini PC μπορεί να φιλοξενήσει έναν σκληρό δίσκο SATA III 2,5" ή SSD. Το NUC5CPYH διαθέτει θύρα HDMI και θύρα VGA για έξοδο βίντεο σε οθόνη ή τηλεόραση. Για τη σύνδεση στο Internet, διαθέτει Wi-Fi 802.11ac, καθώς και υποδοχή Gigabit Ethernet. Για πρόσθετη συνδεσιμότητα, διαθέτει επίσης Bluetooth 4.0 για τη σύνδεση περιφερειακών και άλλων εξωτερικών συσκευών.

1.1.3 Laser Scanner

Το ρομπότ είναι εξοπλισμένο επίσης με ένα σαρωτή λέιζερ της Hokuyo [4] (Εικόνα 2), ο οποίος χρησιμοποιείται για τη χαρτογράφηση του περιβάλλοντος. Το μοντέλο όπου χρησιμοποιήθηκε ήταν το UST-10LX όπου είναι ένας συμπαγής, ελαφρύς αισθητήρας LiDAR 2D όπου πέρα από τη χρήση για χαρτογράφηση, χρησιμοποιείται και για την ανίχνευση και τον εντοπισμό εμποδίων σε αυτόνομες ρομποτικές πλατφόρμες. Είναι εξοπλισμένο με μια διεπαφή Ethernet, με αποτέλεσμα να μπορεί να λάβει υψηλής ταχύτητας και ακρίβειας δεδομένα μέτρησης σε ένα οπτικό πεδίο 270° και σε απόσταση έως και 10 μέτρα. Λόγω της χαμηλής κατανάλωσης ενέργειας, αυτός ο σαρωτής είναι κατάλληλος για πλατφόρμες που λειτουργούν με μπαταρία.



Εικόνα 2. Σαρωτής Laser Hokuyo

1.1.4 Orbbec Astra Camera

Επίσης, διαθέτει μια τρισδιάστατη κάμερα Orbbec Astra [5] (Εικόνα 3) με τηλεσκοπικό βραχίονα, προσφέροντας την αναγκαία μηχανική όραση του ρομπότ. Η σειρά Astra σχεδιάστηκε για να βελτιώσει περαιτέρω τα χαρακτηριστικά που κάνουν τις 3D κάμερες της Orbbec να ξεχωρίζουν από τις υπόλοιπες 3D κάμερες. Οι κάμερες Astra 3D παρέχουν τη λειτουργία όρασης υπολογιστή με δεκάδες λειτουργίες, όπως η αναγνώριση προσώπου, η αναγνώριση χειρονομιών, η παρακολούθηση ανθρώπινου σώματος, τρισδιάστατη μέτρηση, αντίληψη περιβάλλοντος, καθώς και ανακατασκευή τρισδιάστατου χάρτη.

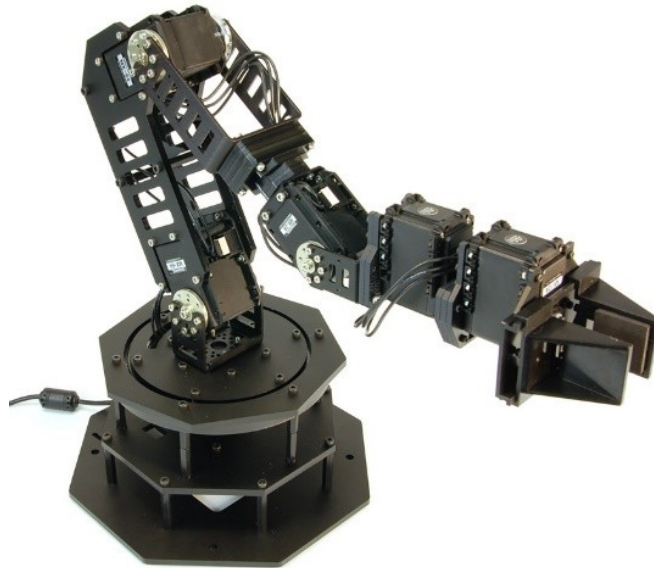


Εικόνα 3. Κάμερα Orbbec Astra

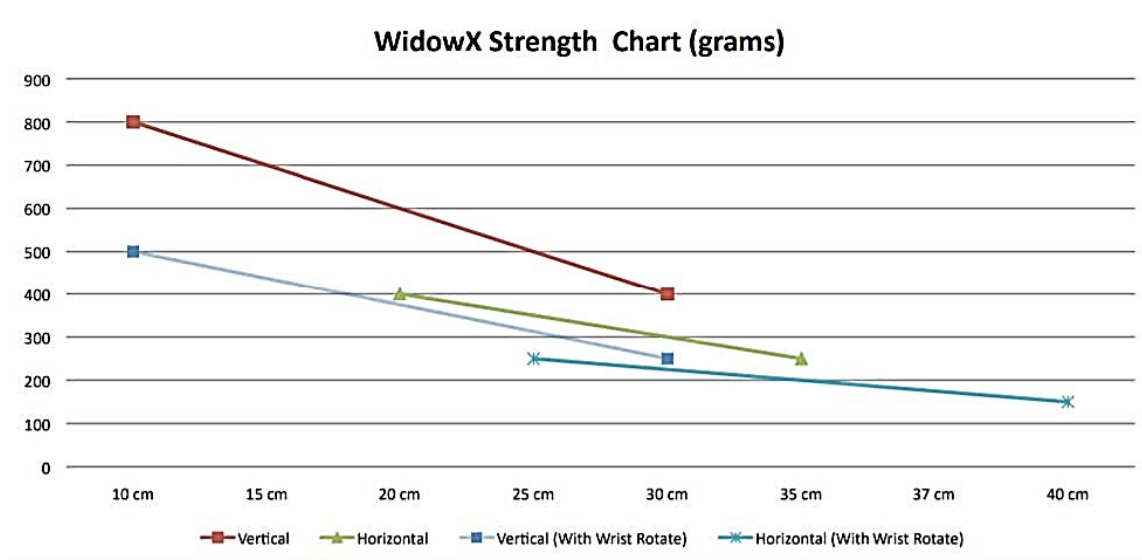
1.1.5 WidowX Robotic Arm

Επίσης, για την υλοποίηση της της διαχείρισης αντικειμένων, έχει προστεθεί ένας ρομποτικός βραχίονας WidowX [6] (Εικόνα 4)

Ο βραχίονας WidowX είναι κατασκευασμένος από τους σερβομηχανισμούς MX της DYNAMIXEL. Οι ενεργοποιητές της σειράς MX παρέχουν πλήρη ελευθερία κίνησης 360 μοιρών στη βάση, εξαιρετικά υψηλή ανάλυση 4096 θέσεων, καθώς και παραμέτρους PID που μπορούν να οριστούν από τον χρήστη. Στην (Εικόνα 5), φαίνονται οι αποστάσεις στις οποίες μπορεί να εκταθεί ο βραχίονας, σε σχέση με το βάρος.



Εικόνα 4. Ρομποτικός βραχίονας WidowX



Εικόνα 5. Διάγραμμα απόστασης



1.2 Ρομποτικό Λογισμικό *Robot Operating System*

Το ROS [7] είναι ένα μεταλειτουργικό σύστημα, το οποίο χρησιμοποιείται σήμερα από εκατοντάδες ερευνητικές ομάδες και εταιρείες του κλάδου της ρομποτικής. Επίσης λόγω της ευκολίας που παρέχει, το καθιστά ως το καλύτερο σημείο εισόδου στην ρομποτική για ερασιτέχνες αλλά και για εκπαιδευτικούς σκοπούς .

Το ROS, αναπτύχθηκε για πρώτη φορά το 2007 από το Stanford Artificial Intelligence Laboratory, με ονομασία Switchyard, ώστε να υποστηρίξει το έργο STAIR (STanford AI Robot), του ομώνυμου εργαστηρίου. Από το 2008 και μέχρι το 2013, η ανάπτυξη συνεχίστηκε αρχικά από το ερευνητικό ινστιτούτο ρομποτικής Willow Garage και εν συνεχεία με περισσότερα από είκοσι ερευνητικά εργαστήρια και ερευνητές, να συνεργάζονται με τους μηχανικούς της Willow Garage για τη περαιτέρω ανάπτυξή του. Το Φεβρουάριο του 2013, η ανάπτυξη του ROS πέρασε υπό την εποπτεία του Open Source Robotics Foundation (OSRF).

Το ίδρυμα ανοιχτού λογισμικού Ρομποτικής (OSRF), είναι μια ανεξάρτητη μη κερδοσκοπική εταιρεία Έρευνας & Ανάπτυξης. Η αποστολή της OSRF είναι να στηρίζει την ανάπτυξη, τη διανομή, και την υιοθέτηση του λογισμικού ανοικτού κώδικα, για χρήση σε ρομποτική έρευνα, την εκπαίδευση και την ανάπτυξη του τελικού προϊόντος.

Το ROS, προς το παρόν λειτουργεί μόνο σε πλατφόρμες που βασίζονται σε UNIX και κατά κύριο λόγο, σε συστήματα Ubuntu και Mac OS X. Άλλες πλατφόρμες που υποστηρίζουν το ROS και που έχουν αναπτυχθεί από την κοινότητα, είναι το Fedora, το Gentoo, το Arch κ.α.

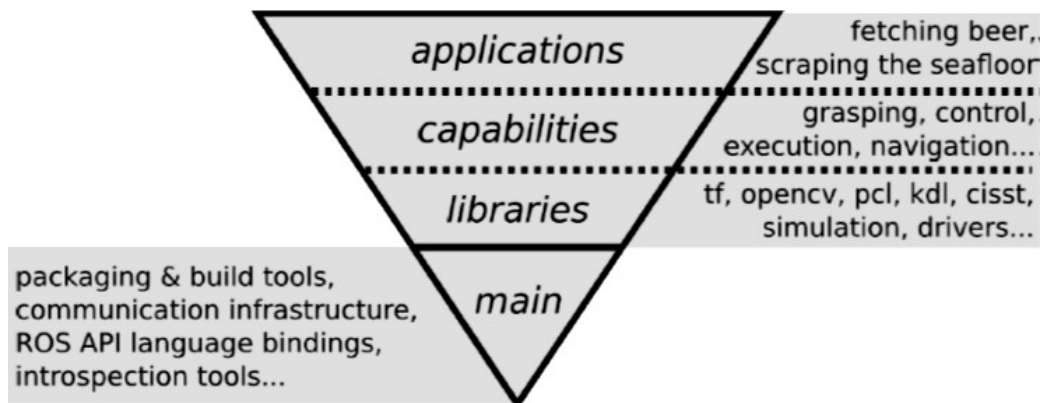
Ο πυρήνας του συστήματος ROS, μαζί με τα εργαλεία και τις βιβλιοθήκες, αναβαθμίζονται τακτικά και κυκλοφορούν ως διανομές ROS με διαφορετικά ονόματα.

Το ROS όπως προαναφέραμε, είναι ένα μεταλειτουργικό σύστημα ανοιχτού κώδικα για ρομπότ. Προσφέρει υπηρεσίες ενός κανονικού λειτουργικού συστήματος, συμπεριλαμβάνοντας επίσης διάφορα επίπεδα, όπως:

- Αφαίρεσης υλικού (hardware abstraction)
- Έλεγχο συσκευής χαμηλού επιπέδου (low-level device control)
- Εφαρμογή των λειτουργιών που χρησιμοποιούνται για τη λειτουργία του ρομπότ
- Ανταλλαγή μηνυμάτων μεταξύ των διεργασιών

- Διαχείριση των πακέτων

Μια συνοπτική απεικόνιση των επιπέδων δίνεται στο παρακάτω εικόνα (4) :



Εικόνα 4. Επίπεδα ROS

Το ROS, έχει ως πρωταρχικό του στόχο, την επαναχρησιμοποίηση του κώδικα στη ρομποτική έρευνα και ανάπτυξη. Αυτό το πετυχαίνει παρέχοντας μια σειρά από εργαλεία και βιβλιοθήκες, που χρησιμεύουν για τη λήψη, την κατασκευή, τη συγγραφή αλλά και την εκτέλεση κώδικα σε πολλούς διαφορετικούς υπολογιστές, λόγω της εφαρμογής τους κάτω από το πλαίσιο του ROS και που είναι κοινό για όλους τους υπολογιστές και τα λειτουργικά τους συστήματα.

Σχετικά με την επαναχρησιμοποίηση του κώδικα, το ROS υποστηρίζεται από ένα ομόσπονδο σύστημα αποθετηρίου κώδικα, επιτρέποντας έτσι την συνεργασία καθώς και τη διανομή του. Έτσι με αυτό το σχεδιασμό, από το επίπεδο συστήματος αρχείων, μέχρι το επίπεδο κοινότητας, επιτυγχάνεται η ανεξαρτησία αποφάσεων για τη συγγραφή κώδικα σε σχέση με την έρευνα και ανάπτυξη ρομποτικών εφαρμογών, όπου στο τέλος όλα μπορούν να συνενωθούν κάτω από την υποδομή του ROS.

1.2.1 Επίπεδα Robot Operating System

Η λειτουργία του ROS, είναι χωρισμένη σε επίπεδα. Έτσι υπάρχουν τρία εννοιολογικά επίπεδα, όπου είναι τα κάτωθι:

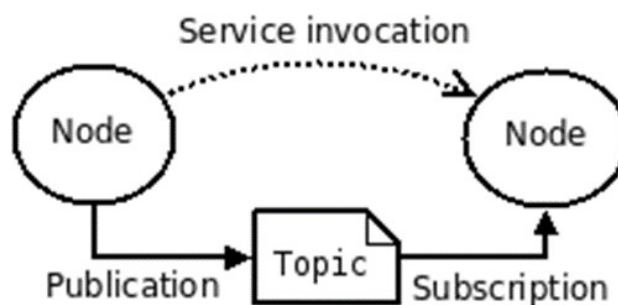
- Επίπεδο συστήματος Αρχείων
- Επίπεδο γραφικής παράστασης υπολογισμού
- Επίπεδο κοινότητας

1.2.2 Επίπεδο συστήματος Αρχείων

Στο επίπεδο συστήματος αρχείων, υπάρχουν οι πόροι του συστήματος που είναι αποθηκευμένοι στο δίσκο, όπως τα “πακέτα” (packages), τα “μεταπακέτα” (metapackages), τα “δηλωτικά αρχεία πακέτων” (Package Manifests), τα “μηνύματα” (message types, msg) και τέλος οι “υπηρεσίες” (service types, srv).

1.2.3 Επίπεδο γραφικής παράστασης υπολογισμού

Σε αυτό το επίπεδο, υπάρχουν οι “κόμβοι” (nodes), ο ROS Master, ο Parameter Server (κομμάτι του Master), τα “μηνύματα” (messages), τα Topics, τα Services και τέλος τα Bags που είναι σημαντικά για την δημιουργία αλλά και το τεστ των αλγορίθμων, καθώς καταγράφουν, αποθηκεύουν και αναπαραγάγουν τα δεδομένα. Ένα παράδειγμα του πως επικοινωνούν οι κόμβοι γραφικά, φαίνεται στην Εικόνα (5).



Εικόνα 5. ROS Nodes Concept

1.2.4 Επίπεδο κοινότητας

Το επίπεδο κοινότητας του ROS, αποσκοπεί στην ένωση κάτω από μια πλατφόρμα, όλων των κοινοτήτων του ROS, με αποτέλεσμα το διαμοιρασμό εφαρμογών και τεχνογνωσίας. Σε αυτές ανήκουν οι “εκδόσεις” του ROS (Distributions), τα “αποθετήρια” εφαρμογών (Repositories), το ROS Wiki, οι ROS Mailing Lists, τα Q&A καθώς και το Blog (ros.org Blog).



Μέσω της συνεργασίας και των τριών επιπέδων, το ROS προσφέρει τη λειτουργικότητα που χρειάζεται μια ρομποτική συσκευή ή εφαρμογή.

2. Εξομοιωτής Σεναρίων

2.1 Ρομποτική προσομοίωση

Η ρομποτική προσομοίωση αφορά την εφαρμογή της προσομοίωσης στον κλάδο της ρομποτικής. Μέσω της ρομποτικής προσομοίωσης, είναι εφικτή η δημιουργία ρομποτικών συστημάτων ελέγχου τα οποία έχουν άμεση σύνδεση με τυχαίες σταδιακές ή όχι αλλαγές οι οποίες πραγματοποιούνται μέσα από ένα εύρος πολλών και συνεχών επαναλήψεων. Επίσης, με την ρομποτική προσομοίωση, το κόστος της διαδικασίας δημιουργίας ρομποτικών συστημάτων δεν είναι υψηλό και ο χρόνος υλοποίησης μειώνεται πολύ περισσότερο.

2.2 Προσομοίωση Ρομποτικών συστημάτων

Ο προσομοιωτής ρομποτικών συστημάτων, χρησιμοποιείται για την υλοποίηση εφαρμογών για ένα φυσικό ρομπότ, χωρίς να εξαρτάται από την φυσική μηχανή, μειώνοντας έτσι κόστος και χρόνο. Στις περισσότερες περιπτώσεις, οι εφαρμογές που υλοποιούνται μεταφέρονται στο ρομπότ χωρίς κάποιες τροποποιήσεις.

2.3 Προσομοιωτής Ρομποτικών συστημάτων Gazebo

Ο Gazebo [8] είναι ένας δυναμικός τρισδιάστατος προσομοιωτής με την ικανότητα να προσομοιώνει με ακρίβεια και αποτελεσματικότητα πολλαπλά ρομπότ και αισθητήρες σε πολύπλοκα εσωτερικά και εξωτερικά περιβάλλοντα. Παρόλο που είναι αρκετά παρόμοιος με μηχανές παιχνιδιών, ο Gazebo προσφέρει προσομοίωση φυσικών νόμων σε πολύ υψηλό βαθμό πιστότητας, αισθητήρων και διεπαφών τόσο για χρήστες όσο και για προγράμματα. Είναι ένας προσομοιωτής ανοιχτού κώδικα και ήταν μέρος του Player Project από το 2004 μέχρι το 2011. Από το 2011, ο Gazebo είναι ανεξάρτητο λογισμικό και υποστηρίζεται από τον Willow Garage. Περιλαμβάνει την φυσική μηχανή Open Dynamics Engine (ODE) [9] και την βιβλιοθήκη OpenGL [10].



Ο προσομοιωτής Gazebo παρουσιάζει κάποια σημαντικά χαρακτηριστικά και πλεονεκτήματα. Αρχικά, κάποιες από τις βασικές χρήσεις του προσομοιωτή είναι κυρίως για δοκιμές αλγόριθμων ρομποτικής, σχεδιασμό και μοντελισμό ρομπότ και έλεγχο πραγματικών σεναρίων μέσω ρομπότ και αισθητήρων, σχεδιασμένα από τον ίδιο τον προσομοιωτή. Γενικά, ο Gazebo περιέχει πολλαπλές φυσικές μηχανές, μια ολοκληρωμένη βιβλιοθήκη έτοιμων ρομποτικών μοντέλων και περιβαλλόντων, ένα σύνολο αισθητήρων και γραφικές, εύκολες προς τον χρήστη προγραμματιστικές διεπαφές που διευκολύνουν τον προγραμματιστή και ερευνητή στην χρήση του προσομοιωτή. Κάτι που παρουσιάζει μεγάλο ενδιαφέρον και θεωρείται τεράστιο συν του προσομοιωτή Gazebo είναι το γεγονός ότι μπορεί να αναβαθμιστεί με την χρήση και ενσωμάτωση του ROS framework.

Με την ενσωμάτωση του ROS στον Gazebo, ο Gazebo αναβαθμίζεται και αποκτάει επιπλέον χαρακτηριστικά. Αυτό επιτυγχάνεται μέσω ενός συνόλου πακέτων ROS που ονομάζονται *gazebo_ros_pkgs*. Παρέχουν κατάλληλα wrappers και διεπαφές για την ανάλογη προσομοίωση ρομπότ με χρήση ROS μηνυμάτων, υπηρεσιών και δυναμικής αναμόρφωσης. Είναι εφικτή η υλοποίηση με το εργαλείο catkin [11], ένα επίσημο build system του ROS και διάδοχος του rosbuilt, που είναι επίσης επίσημο build system του ROS. Μειώνεται ο διπλότυπος κώδικας καθώς και κώδικας από προηγούμενες εκδόσεις ROS και Gazebo. Τέλος, δίνεται καλύτερη υποστήριξη για controllers με την χρήση του *ros_control*.

Θα πρέπει να σημειωθεί πως υποστηρίζεται και ένα βασικό εργαλείο του ROS όπου είναι το Rviz. Το Rviz χρησιμοποιείται για να απαντάει στην βασικότερη ερώτηση του χρήστη: τί βλέπει – καταλαβαίνει η ρομποτική πλατφόρμα τώρα; Άρα πρόκειται για την οπτική αναπαράσταση των δεδομένων όπου συλλέγει από το περιβάλλον όσο το εξωτερικό όσο και από το εσωτερικό όπου είναι η πληροφορία των topics.





3. Τεχνολογία Bluetooth Low Energy

3.1 Bluetooth

Η τεχνολογία Bluetooth [12] είναι ένα πρωτόκολλο επικοινωνίας για ασύρματη ανταλλαγή δεδομένων σε μικρές αποστάσεις. Αναπτύχθηκε από την Ericsson το 1994 σαν μια ασύρματη έκδοση του πρωτοκόλλου RS-232. Η τεχνολογία Bluetooth Low Energy αναπτύχθηκε αρχικά από την Nokia και παρουσιάστηκε το 2006 υπό την ονομασία Wibree¹. Κατόπιν διαπραγματεύσεων με το Bluetooth Special Interest Group συμφωνήθηκε το νέο πρωτόκολλο Wibree να συμπεριληφθεί σε μετέπειτα εκδόσεις του Bluetooth Specification σαν μια έκδοση Ultra – Low – Power έκδοση Bluetooth.

Το 2011 το Bluetooth Special Interest Group εισήγαγε στην αγορά το Bluetooth Low Energy με την έκδοση 4.0 Bluetooth Core Specification. Η επόμενη μεγάλη αναβάθμιση έγινε τον Δεκέμβριο 2013 με την έκδοση 4.1 του Bluetooth Core όπου και υπήρξαν μεγάλες βελτιώσεις και αλλαγές. Όπως όλες οι προηγούμενες προδιαγραφές Bluetooth έτσι και η

4.1 είναι συμβατή με την 4.0 (όσον αφορά το BLE, εφόσον υπάρχει μέχρι στιγμής μόνο στην 4.1 και 4.0 έκδοση) εξασφαλίζοντας την διαλειτουργικότητα συσκευών βασισμένων σε προηγούμενες προδιαγραφές.

¹ Το Wibree είναι ένα σύνολο πρωτοκόλλων που παρέχονται για ασύρματη επικοινωνία μικρής εμβέλειας. Αυτή η τεχνολογία χρησιμοποιείται για μικρότερη κατανάλωση ενέργειας. Αναπτύχθηκε κυρίως για την ανταλλαγή δεδομένων Bluetooth. Πρόκειται για μια ελαφριά τεχνολογία που μπορεί να συνδυαστεί με μικροσκοπικούς πομπούς και μικρές μπαταρίες που καθιστούν ολόκληρη τη συσκευή πολύ ελαφριά και εύκολη στη μεταφορά.



Οι προδιαγραφές 4.1 και 4.0 συμπεριλαμβάνει και το κλασικό Bluetooth, αλλά και το Bluetooth Low Energy. Αυτά τα δύο πρότυπα δεν είναι άμεσα συμβατά μεταξύ τους και οποιαδήποτε συσκευή με πυρήνα παλαιότερο του 4.0 δεν μπορεί να επικοινωνήσει με κάποια άλλη BLE συσκευή. Τα πρωτόκολλα που χρησιμοποιούνται μεταξύ των δύο τεχνολογιών αλλά και οι εφαρμογές είναι τελείως διαφορετικές και ασύμβατες μεταξύ τους.

3.2 Bluetooth Low Energy

Η νέα τεχνολογία στο Bluetooth πρότυπο δεν είναι συμβατή με την ήδη υπάρχουσα. Παρόλα αυτά οι δύο τεχνολογίες μοιράζονται το ίδιο όνομα εφόσον πολλά από τα χαρακτηριστικά του κλασικού Bluetooth υιοθετηθήκαν από το Bluetooth Low Energy [13] όπως ο μηχανισμός προσαρμοστικής αναπήδησης συχνότητας (Adaptive Frequency Hopping AFH), καθώς και πλειάδα πρωτοκόλλων (LLC, L2CAP) που χρησιμοποιούνται στο κλασικό Bluetooth. Ακόμη το BLE ενσωματώνει τις ίδιες διαδικασίες για την ασφάλεια της σύνδεσης όπως αυτή της σύζευξης, της ασφαλούς πιστοποίησης και της κρυπτογράφησης. Τέλος η συχνότητα λειτουργίας και των δύο τεχνολογιών είναι στα 2.4GHz ISM Band και έχουν επίσης παρόμοια έξοδο ισχύος ραδιοσυχνότητας.

Παρότι το Bluetooth Low Energy εισήχθη στην αγορά σαν μια χαμηλής κατανάλωσης έκδοση του κλασικού Bluetooth, εφόσον έχουν πολλά κοινά χαρακτηριστικά, αυτό που κάνει το νέο πρότυπο να διαφέρει είναι ότι υποστηρίζει την αναστολή λειτουργίας. Η ικανότητα αυτή επιτρέπει να βρίσκεται σε μια κατάσταση χαμηλής κατανάλωσης ενέργειας και με αυτόν τρόπο εκπέμπει μόνο όταν χρειάζεται. Η μία περίπτωση είναι όταν αρχίζει μια σύνδεση ή όταν χρειάζεται να διατηρήσει κάποια σύνδεση. Η κατανάλωση ενέργειας βρίσκεται σε χαμηλά επίπεδα λόγω ότι οι χρονικές περίοδοι σύνδεσης διαρκούν μόλις λίγα δευτερόλεπτα. Η μέγιστη κατανάλωση ισχύος ανέρχεται στα 15mA και η μέση κατανάλωση στο 1uA.

Κάτι που πρέπει να αναφερθεί είναι ότι, η τεχνολογία που είναι πιο αποδοτική εξαρτάται κυρίως από την εφαρμογή που θα χρησιμοποιηθεί. Στο κλασικό Bluetooth οι ρυθμοί μετάδοσης που μπορούν να επιτευχθούν, μπορεί να ξεπερνούν τα 2Mb/s (ωφέλιμο φορτίο) σε αντίθεση με την νέα τεχνολογία που ο ρυθμός μετάδοσης είναι χαμηλότερος των 100kb/s. Από αυτό μπορούμε να αντιληφθούμε, πως αν ο σκοπός μίας εφαρμογής

είναι η μετάδοση δεδομένων με την χρήση BLE, όποια ενδεχόμενη εξοικονόμηση ενέργειας θα χαθεί, εφόσον η συσκευή θα χρειαστεί ανελλιπώς να εκπέμπει προκειμένου να ανταποκριθεί στους ρυθμούς μετάδοσης της εφαρμογής. Στην ακόλουθη (Εικόνα 6), παρουσιάζονται τα χαρακτηριστικά των δύο τεχνολογιών:

	Classic Bluetooth technology	Bluetooth low energy technology
Data payload throughput (net)	2Mbps	~100kbps
Robustness	Strong	Strong
Range	Up to 1000m	Up to 250m
Local system density	Strong	Strong
Large scale network	Weak	Good
Low latency	Strong	Strong
Connection set-up speed	Week	Strong
Power consumption	Good	Very strong
Cost	Good	Strong

Εικόνα 6. χαρακτηριστικά των δύο τεχνολογιών

Εντέλει, μπορεί οι δυο τεχνολογίες να έχουν κοινό όνομα και χαρακτηριστικά, είναι όμως πολύ διαφορετικά πρότυπα. Για κάποιες εφαρμογές ενδεχομένως η κλασική έκδοση Bluetooth να είναι η πιο κατάλληλη και για κάποιες άλλες το BLE να είναι η καλύτερη επιλογή. Σε καμία περίπτωση όμως το νέο αυτό πρότυπο δεν ήρθε για να αντικαταστήσει το κλασικό Bluetooth.

3.3 Αρχιτεκτονική Bluetooth

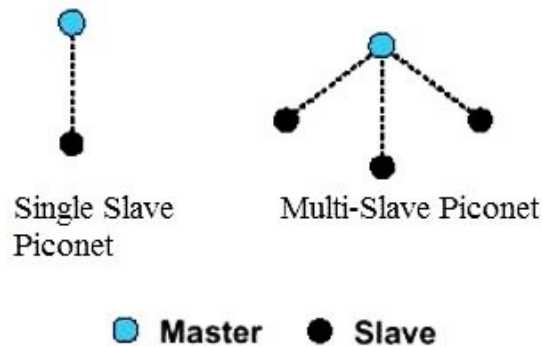
Η αρχιτεκτονική του Bluetooth ορίζει δύο τύπους δικτύων:

1. Δίκτυο Piconet
2. Δίκτυο Scatternet

3.3.1 Δίκτυο Piconet

Το Piconet [14] είναι ένας τύπος δικτύου Bluetooth που περιέχει έναν πρωτεύοντα κόμβο που ονομάζεται κύριος κόμβος (master node), καθώς και επτά ενεργούς δευτερεύοντες κόμβους όπου ονομάζονται δευτερεύοντες κόμβοι (slave nodes). Έτσι, μπορούμε να πούμε ότι υπάρχουν συνολικά 8 ενεργοί κόμβοι οι οποίοι βρίσκονται σε απόσταση 10 μέτρων. Η επικοινωνία μεταξύ του πρωτεύοντος και των δευτερευόντων κόμβων, μπορεί να είναι ένα προς ένα ή ένα προς πολλά. Η επικοινωνία είναι δυνατή μόνο μεταξύ του

κύριου και του δευτερεύοντος, ενώ η επικοινωνία μεταξύ δευτερευόντων δεν είναι δυνατή. Διαθέτει επίσης 255 σταθμευμένους κόμβους, οι οποίοι είναι δευτερεύοντες κόμβοι και δεν μπορούν να λάβουν μέρος στην επικοινωνία εκτός αν μετατραπούν σε ενεργή κατάσταση. Η σχηματική αναπαράσταση του δικτύου Piconet, παρουσιάζεται στην (Εικόνα 7)

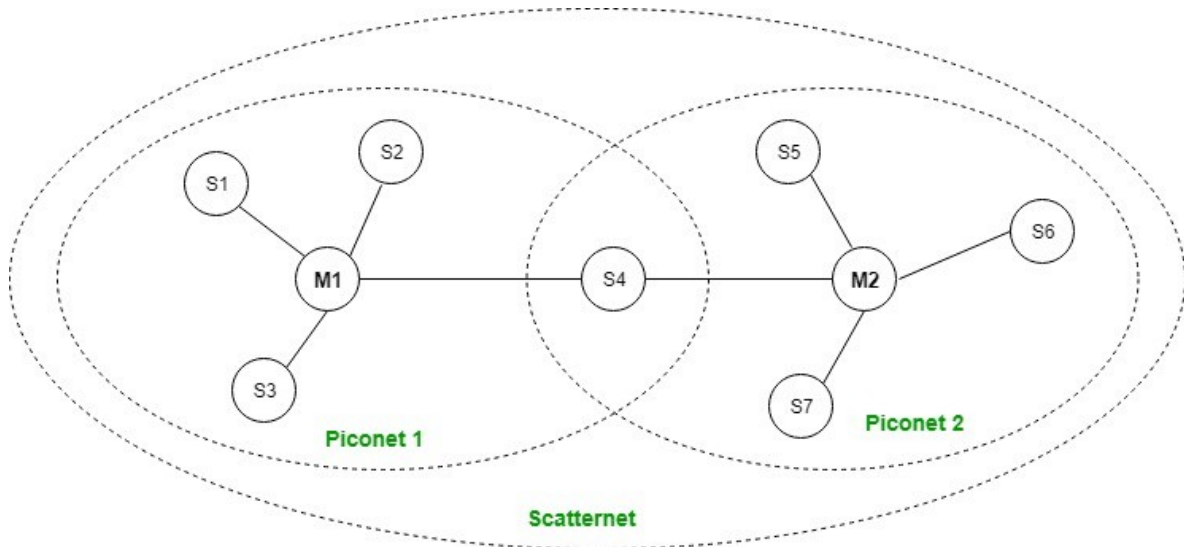


Εικόνα 7. Δίκτυο Piconet

3.3.2 Δίκτυο Scatternet

Η αρχιτεκτονική Scatternet [15], διαμορφώνεται με τη χρήση διαφόρων piconets. Ένας slave που υπάρχει σε ένα piconet μπορεί να ενεργεί ως master ή μπορούμε να πούμε ως primary σε ένα άλλο piconet. Αυτό το είδος κόμβου μπορεί να λάβει ένα μήνυμα από έναν κύριο σε ένα piconet και να παραδώσει το μήνυμα στον δευτερεύοντα που έχει στο άλλο piconet. Αυτός ο τύπος κόμβου αναφέρεται ως κόμβος γέφυρα. Ένας σταθμός δεν μπορεί να είναι master σε δύο piconets.

Η σχηματική αναπαράσταση του δικτύου Scatternet, παρουσιάζεται στην (Εικόνα 8)



Εικόνα 8. Δίκτυο Scatternet

3.4 Δομικά στοιχεία στοίβας Bluetooth

Η στοίβα πρωτοκόλλων του BLE [16] αποτελείται από 3 βασικά δομικά στοιχεία:

- Application
- Host
- Controller

3.4.1 Application

Το επίπεδο εφαρμογής, είναι το ανώτατο επίπεδο της στοίβας μιας BLE εφαρμογής. Περιλαμβάνει όλη την λογική της εφαρμογής, την διάδραση με τον χρήστη, τον χειρισμό των δεδομένων και γενικά ότι σχετίζεται με τον σκοπό για τον οποίο αναπτύχθηκε η εκάστοτε εφαρμογή.

3.4.2 Host

Είναι το ανώτερο επίπεδο της στοίβας του BLE, και συμπεριλαμβάνει τα ακόλουθα πρωτόκολλα:

- Generic Access Profile (GAP)
- Generic Attribute Profile (GATT)
- Logical Link Control and Adaptation Protocol (L2CAP)
- Attribute Protocol (ATT)
- Security Manager (SM)



- Host Controller Interface (HCI), η πλευρά του Host

3.4.2.1 Logical Link Control and Adaptation Protocol (L2CAP)

Το Logical Link Control and Adaptation Protocol (L2CAP) παρέχει δύο βασικές λειτουργίες. Αρχικά χρησιμοποιείται σαν πολυπλέκτης πρωτοκόλλων, λαμβάνει πολλαπλά πρωτόκολλα από τα πιο ψηλά στρώματα και στην συνέχεια τα ενθυλακώνει σε πακέτα BLE και το ανάποδο.

Ακόμη εκτελεί διάτμηση και επανασύνδεση, διαδικασία κατά την οποία παίρνει μεγάλα πακέτα από τα ψηλά επίπεδα και τα χωρίζει σε πακέτα των 27-byte προκειμένου να ταιριάξουν με το μέγιστο φόρτο των πακέτων του BLE προτύπου. Από την πλευρά της λήψης, λαμβάνει μικρά πακέτα από τα κατώτερα στρώματα και στην συνέχεια τα συνδέει και τα δρομολογεί στα αντίστοιχα υψηλά επίπεδα της στοίβας. Θα μπορούσαμε να κάνουμε μια αντιστοίχιση του πρωτοκόλλου L2CAP με το πρωτόκολλο TCP όπου επιτρέπει την συνύπαρξη πολλών πρωτοκόλλων σε ένα φυσικό μέσο, όπου το καθένα έχει διαφορετικό μέγεθος πακέτου και διαφορετικές απαιτήσεις.

Στο BLE το L2CAP επίπεδο είναι υπεύθυνο για την δρομολόγηση κυρίως δύο πρωτοκόλλων, του ATT (Attribute Protocol) και του SMP (Security Manager Protocol).

3.4.2.2 Attribute Protocol (ATT)

Το Attribute Protocol αποτελεί κατά κύριο λόγο την βάση δεδομένων που έχει κάθε BLE συσκευή. Είναι ένα Client/Server πρωτόκολλο βασισμένο σε κάποια γνωρίσματα (attributes). Στο BLE, κάθε συσκευή μπορεί να έχει το όρισμα του Client είτε του Server άσχετα αν είναι master ή slave. Ο Client «ζητά» δεδομένα από τον Server και στην συνέχεια τα αποστέλλει. Το παρόν πρωτόκολλο είναι αυστηρά ακολουθιακό, αν μια αίτηση δηλαδή από κάποιον client εκκρεμεί, ο server δεν ανταποκρίνεται στις υπόλοιπες αιτήσεις μέχρι να εξυπηρετηθεί η προηγούμενη αίτηση. Αυτό ισχύει και για τον ρόλο του Client αλλά και για τον Server.

Τα δεδομένα σε ένα server είναι οργανωμένα σε γνωρίσματα, κάθε γνώρισμα έχει:

- ένα μοναδικό 16-bit αριθμό που αποτελεί το attribute handle, χρησιμοποιείται σαν αναγνωριστικό για το συγκεκριμένο γνώρισμα προκειμένου να πάρουμε πρόσβαση στην τιμή του γνωρίσματος.



- ένα UUID, το οποίο δίνει τον τύπο και την φύση των δεδομένων που βρίσκονται μέσα στο γνώρισμα.
- ένα σύνολο δικαιωμάτων για το συγκεκριμένο γνώρισμα, ορίζει αν η τιμή μπορεί να γραφτεί ή να διαβαστεί από τον client.
- την τιμή που έχει το γνώρισμα.

Όταν ο Client θελήσει να διαβάσει ή να γράψει κάποιο από τα γνωρίσματα που παρέχονται σε έναν server γνωστοποιεί ένα αίτημα για διάβασμα ή για γράψιμο με την χρήση του attribute handle. Από τη μεριά του server, όταν ληφθεί μια αίτηση μηνύματος, τότε είτε θα επιστρέψει τα χαρακτηριστικά από αυτό που του ζητείται, είτε θα επιστρέψει έναν κωδικό επιβεβαίωσης παραλαβής - σφάλματος.

3.4.2.3 Security Manager (SM)

Το επίπεδο security manager λειτουργεί σαν πρωτόκολλο και αποτελείται από μια σειρά από αλγορίθμους ασφαλείας που επιτρέπουν στην τεχνολογία να ανταλλάσσει κλειδιά ασφαλείας, προκειμένου να γίνεται ασφαλής μεταφορά δεδομένων πάνω από μια κωδικοποιημένη σύνδεση.

Το επίπεδο security manager ορίζει δύο ρόλους:

1. τον Initiator – πάντα έχει τον ρόλο του master όσον αφορά το Link Layer επίπεδο.
2. Τον Responder – πάντα είναι ο slave σε Link Layer επίπεδο.

Το παρόν επίπεδο ορίζει τις τρεις ακόλουθες διαδικασίες ασφαλείας:

- **Pairing:** στην παρούσα διαδικασία παράγεται ένα προσωρινό κλειδί (Short Term Key -STK) που χρησιμοποιείται για την κρυπτογράφηση της σύνδεσης. Αυτό το προσωρινό κλειδί δεν αποθηκεύεται και συνεπώς δεν επαναχρησιμοποιείται σε μετέπειτα συνδέσεις.
- **Bonding:** είναι παρόμοια διαδικασία με αυτή του pairing και απλά τα κλειδιά που ανταλλάσσονται αποθηκεύονται σε μη-πτητικές μνήμες προκειμένου να επαναχρησιμοποιηθούν σε μετέπειτα συνδέσεις χωρίς να χρειάζεται να γίνει επανάληψη της διαδικασίας του bonding.



- **Encryption Re-establishment:** Μετά την διαδικασία του bonding, τα κλειδιά για την κρυπτογράφηση της σύνδεσης έχουν αποθηκευθεί στις συσκευές. Η παρούσα διαδικασία ορίζει πως θα χρησιμοποιηθούν αυτά τα κλειδιά σε επικείμενες επανασυνδέσεις.

Η διαδικασία του pairing δημιουργεί μια ασφαλή σύνδεση η οποία διαρκεί όσο οι συσκευές είναι συνδεδεμένες, ενώ το bonding δημιουργεί μόνιμο δεσμό στις συσκευές έως ότου κάποια πλευρά αποφασίσει και διαγράψει αυτόν τον μόνιμο δεσμό.

Επίσης, κατά την διάρκεια του bonding πάντα γίνεται αρχικά pairing. Σε μια διαδικασία όπου οι συσκευές θέλουν να δημιουργήσουν έναν μόνιμο δεσμό σε πρώτη φάση παράγουν και οι δύο τα προσωρινά κλειδιά (STK) και στην συνέχεια η σύνδεση κρυπτογραφείται με την χρήση αυτών των κλειδιών. Όταν η σύνδεση είναι ασφαλής τότε μόνο ανταλλάσσονται τα μόνιμα κλειδιά για αποθήκευση και επαναχρησιμοποίηση.

Για την παραγωγή αυτών των κλειδιών (STK) κατά την διάρκεια της ανταλλαγής πακέτων, κατά την εκτέλεση της pairing διαδικασίας, οι δύο συσκευές διαπραγματεύονται σχετικά με τον τρόπο που θα παράξουν τα STK κλειδιά. Οι διαδικασίες για την παραγωγή των κλειδιών είναι:

- **Just Works:** το κλειδί STK παράγεται και από τις δυο πλευρές βάση των πακέτων που έχουν ανταλλαχτεί. Αυτό δεν προσφέρει καμία προστασία σε MITM (Man In the Middle) επιθέσεις.
- **Passkey Display:** μια από τις συσκευές παράγει και γνωστοποιεί έναν 6-ψηφιο κωδικό και ζητά από το άλλο peer να τον εισάγει.
- **Out Of Band:** όταν χρησιμοποιείται αυτή η μέθοδος επιπλέον δεδομένα ανταλλάσσονται χωρίς την χρήση του BLE radio αλλά με άλλες ασύρματες τεχνολογίες.

3.4.2.4 Generic Attribute Profile (GATT)

Το Generic Attribute Profile πρωτόκολλο στηρίζεται πάνω στο Attribute Protocol (ATT). Στην ουσία κατά κάποιο τρόπο ομαδοποιεί τα γνωρίσματα (attributes) του πρωτοκόλλου ATT σε χαρακτηριστικά (characteristics). Με την σειρά τους τα χαρακτηριστικά αυτά ομαδοποιούνται σε διάφορες υπηρεσίες (services). Τα χαρακτηριστικά μπορούμε να τα

περιγράψουμε σαν ενότητες οι οποίες περιλαμβάνουν την τιμή των δεδομένων καθώς και την περιγραφή των δεδομένων που αντιπροσωπεύει αυτή η τιμή.

3.4.2.5 Generic Access Profile (GAP)

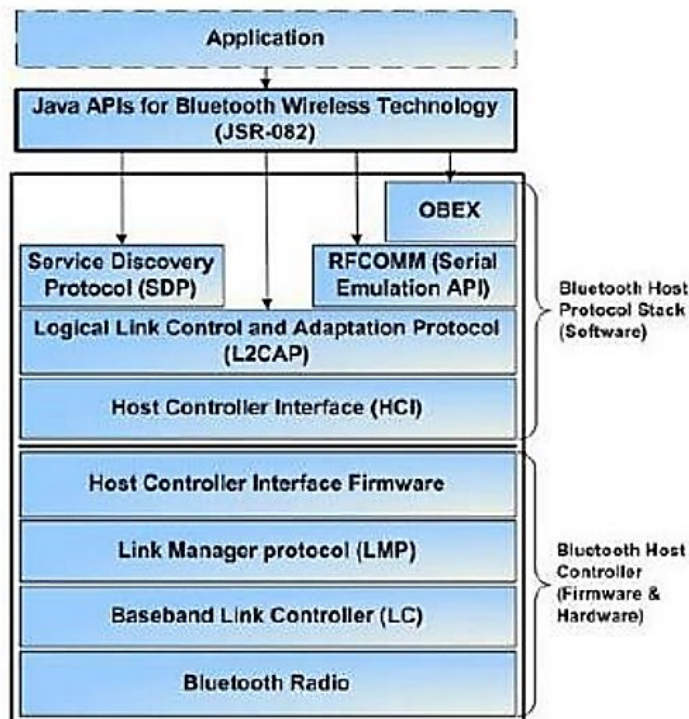
Το Generic Access Profile ορίζει πως οι συσκευές επικοινωνούν μεταξύ τους. Μπορούμε να θεωρήσουμε ότι το συγκεκριμένο πρωτόκολλο αποτελεί το υψηλότερο επίπεδο ελέγχου εφόσον προσδιορίζει το πώς οι συσκευές εκτελούν διαδικασίες όπως ανίχνευση συσκευών, σύνδεση, ασφάλεια, και άλλες διαδικασίες προκειμένου να εξασφαλιστεί η διαλειτουργικότητα και η ανταλλαγή δεδομένων μεταξύ συσκευών από διαφορετικούς κατασκευαστές.

3.4.3 Controller

Το κατώτερο επίπεδο της στοίβας, συμπεριλαμβανομένου και του πομπού συχνότητας καθώς και των ακόλουθων επιπέδων:

- Host Controller Interface (HCI), η πλευρά του Controller
- Link Layer (LL)
- Physical Layer (PHY)

Στην παρακάτω εικόνα, (Εικόνα 9), παρουσιάζεται η στοίβα πρωτοκόλλων του BLE



Εικόνα 9. Αρχιτεκτονική BLE (Στοίβα πρωτοκόλλων)



3.4.3.1 Host Controller Interface (HCI)

Η αρχιτεκτονική του BLE αποτελείται από 3 κύρια δομικά στοιχεία Application, Host και Controller. Αυτό επιτρέπει στην τεχνολογία να μπορεί να «μοιράσει» την εκτέλεση του κάθε επιπέδου σε διαφορετικό hardware. Για παράδειγμα μπορεί ο controller να εκτελείται σε ένα ολοκληρωμένο κύκλωμα και ο host μαζί με το application να τρέχει σε έναν άλλον επεξεργαστή. Οι δύο ξεχωριστές μονάδες μπορούν να επικοινωνούν μεταξύ τους φυσικά με κάποιο πρωτόκολλο σειριακής επικοινωνίας όπως UART, USB, SPI η κάτι άλλο. Η διεπαφή που επιτρέπει στον host και στον controller να ανταλλάσσουν μηνύματα λέγεται Host Controller Interface. Το Bluetooth πρότυπο ορίζει το HCI σαν έναν σύνολο εντολών και προ τυποποιημένων γεγονότων για τον Host και τον Controller που τους επιτρέπουν να συνδιαλέγονται μεταξύ τους. Επίσης ορίζει την μορφή των πακέτων που ανταλλάσσονται, καθώς και ένα σύνολο κανόνων για την επικοινωνία όπως και κάποιες επι πλέον διαδικασίες.

Το χαρακτηριστικό της τεχνολογίας να μπορεί να διαμοιράζει τον controller (σύστημα που έχει πολύ αυστηρούς χρονικούς περιορισμούς και είναι το μόνο σημείο που έρχεται σε επαφή με το φυσικό μέσο) και τον Host (σύστημα που περιλαμβάνει πολύπλοκες επεξεργασίες αλλά δεν έχει αυστηρούς χρονικούς περιορισμούς) την καθιστά ιδανική για εφαρμογές που τρέχουν σε smartphones, tablets και γενικά συσκευές που έχουν επεξεργαστική ισχύ. Έτσι η πιο συνηθισμένη αρχιτεκτονική μίας τέτοιας συσκευής υλοποιεί τον Host στην κεντρική μονάδα επεξεργασίας ενώ ο controller βρίσκεται σε ξεχωριστό υλικό.

3.4.3.2 Link layer

Το επίπεδο Link βρίσκεται σε άμεση επαφή με το physical επίπεδο και είναι συνδυασμός hardware και software. Στο hardware κομμάτι του εκτελούνται κοστοβόρες, όσον αφορά την επεξεργασία, ενέργειες, προκειμένου να μην επιβαρυνθεί επιπλέον η κεντρική μονάδα

επεξεργασίας (όπου και τρέχουν άλλα πρωτόκολλα άλλων επιπέδων). Η λειτουργία του hardware κομματιού του Link επιπέδου περιλαμβάνει:

- Preamble, διεύθυνση πρόσβασης και εναέρια διάτμηση πρωτοκόλλου



- CRC γεννήτρια, και επαλήθευση
- Data whitening
- Γεννήτρια τυχαίων αριθμών
- AES κρυπτογράφηση

Το software κομμάτι του επιπέδου διαχειρίζεται την κατάσταση σύνδεσης του ραδιοπομπού, πως δηλαδή μια συσκευή μπορεί να συνδεθεί με άλλες συσκευές. Μια συσκευή BLE μπορεί να λειτουργήσει είτε σαν master είτε σαν slave είτε και τα δύο, ανάλογα με τις απαιτήσεις της εφαρμογής. Οι συσκευές που κάνουν εκκίνηση συνδέσεων είναι masters και οι συσκευές που διαφημίζονται και δέχονται συνδέσεις είναι slaves. Ένας master μπορεί να συνδεθεί με πολλούς slaves και ένας slave μπορεί να συνδεθεί με πολλούς masters. Τυπικά οι συσκευές που συνήθως είναι masters έχουν μεγάλα αποθέματα σε ενεργειακούς και επεξεργαστικούς πόρους ενώ μικρές συσκευές με περιορισμένους πόρους συνήθως είναι slaves. Το BLE έχει υιοθετήσει μια ασυμμετρία στα κατώτερα στρώματα του, μεταξύ master και slave έτσι ώστε να απαιτούνται περισσότεροι πόροι για λειτουργήσει μια συσκευή σαν master. Αυτή η ασυμμετρία επιτρέπει φτηνά περιφερειακά να λειτουργούν με μικρούς επεξεργαστές και ραδιοπομπούς ενώ όλοι οι πολύπλοκοι υπολογισμοί των χαμηλότερων επιπέδων να γίνονται από συσκευές με περισσότερους πόρους όπως είναι τα smartphones και τα tablets.

Το Link επίπεδο ορίζει τους ακόλουθους ρόλους:

- Advertiser: η συσκευή που στέλνει διαφημιστικά πακέτα.
- Scanner: η συσκευή που ανιχνεύει διαφημιστικά πακέτα.
- Master: η συσκευή που αρχικοποιεί την διαδικασία για μια νέα σύνδεση.
- Slave: η συσκευή που δέχεται μια αίτηση σύνδεσης και ανταποκρίνεται στις οδηγίες του master.

Οι παραπάνω ρόλοι μπορούν να ομαδοποιηθούν στα ακόλουθα ζεύγη:

- Advertiser και Scanner: όταν δεν υπάρχει σύνδεση μεταξύ των δύο συσκευών.
- Master και Slave: όταν έχει δημιουργηθεί σύνδεση.



3.5 Physical Layer

Το Physical επίπεδο περιλαμβάνει όλα τα κυκλώματα που είναι υπεύθυνα για την διαμόρφωση και αποδιαμόρφωση των αναλογικών σημάτων και τον μετασχηματισμό τους σε ψηφιακές ακολουθίες.

Ο ραδιοπομπός που χρησιμοποιείται λειτουργεί στην μπάντα των 2.4GHz ISM και την διαιρεί σε 40 κανάλια, από 2.4000GHz έως και 2.4835. Από αυτά τα 40 κανάλια τα 37 είναι για μεταφορά δεδομένων σύνδεσης και τα υπόλοιπα 3 είναι αποκλειστικά και μόνο για διαφήμιση. Το πρότυπο προκειμένου να εξαλείψει οποιαδήποτε ραδιοπαρεμβολή σε κάποιο κανάλι (ιδίως για τον λόγο ότι χρησιμοποιεί την ίδια συχνότητα με το wi-fi) κάνει χρήση της προσαρμοστικής αναπήδησης συχνότητας. Το κανάλι που περιμένει ο πομπός να λάβει δεδομένα δίνεται από την σχέση $channel = (current_channel + hop) \bmod 37$ και η τιμή του hop γίνεται γνωστή στον δέκτη κατά την διάρκεια εκκίνησης μίας σύνδεσης. Η διαμόρφωση που χρησιμοποιείται για την ακολουθία σημάτων είναι Gaussian Frequency Shift Keying (GFSK) και ο ρυθμός διαμόρφωσης είναι στα 1Mbit/s, που είναι και το ανώτατο όριο που μπορεί να φτάσει η συγκεκριμένη τεχνολογία (αυτό το ανώτατο όριο βέβαια δεν επιτυγχάνεται ποτέ εφόσον κάθε επίπεδο πρωτοκόλλου προσθέτει επιπλέον κεφαλίδες άρα το ωφέλιμο φορτίο γίνεται όλο και πιο μικρό).

3.6 Ρόλοι συσκευών Bluetooth

Κάθε συσκευή μπορεί να λειτουργήσει υιοθετώντας περισσότερους από έναν ρόλους λειτουργίας. Ο κάθε ρόλος προϋποθέτει περιορισμούς και επιβάλλει συγκεκριμένη συμπεριφορά για την εκάστοτε συσκευή. Το πρωτόκολλο ορίζει τέσσερις διαφορετικούς ρόλους προκειμένου μια συσκευή να εισέλθει σε ένα BLE δίκτυο:

- **Broadcaster:** ο συγκεκριμένος ρόλος είναι βελτιστοποιημένος ως προς την αποστολή δεδομένων μόνο. Ο broadcaster κάνει περιοδική αποστολή δεδομένων με την χρήση διαφημιστικών πακέτων και τα δεδομένα είναι διαθέσιμα σε όποια συσκευή εκείνη την στιγμή βρίσκεται σε εμβέλεια.
- **Observer:** βελτιστοποιημένος περισσότερο στο να λαμβάνει δεδομένα από άλλες συσκευές που λειτουργούν σαν Broadcasters. Η λειτουργία του περιορίζεται στο να λαμβάνει πακέτα.



- **Central:** ο ρόλος του κεντρικού αντιστοιχεί στον ρόλο του master από την πλευρά του Link επιπέδου. Μια συσκευή με αυτόν τον ρόλο μπορεί να αρχικοποιεί συνδέσεις και να εγκρίνει την πρόσβαση συσκευών σε ένα BLE δίκτυο. Επειδή αυτός ο ρόλος προϋποθέτει την λειτουργία σαν master η συσκευή επιφορτίζεται με περισσότερες επεξεργαστικές υποχρεώσεις, για τον λόγο αυτό τέτοιες συσκευές είναι ως επί το πλείστον smartphones, tablets και γενικότερα συσκευές με μεγάλους επεξεργαστικούς και ενεργειακούς πόρους.
- **Peripheral:** ο ρόλος προϋποθέτει τον ρόλο του slave από την πλευρά του Link επιπέδου. Μια συσκευή επιφορτισμένη με τον συγκεκριμένο ρόλο στέλνει διαφημιστικά πακέτα προκειμένου να γίνει αντιληπτή από κάποια άλλη συσκευή central, προκειμένου να αρχίσει η διαδικασία της σύνδεσης.

3.7 Συσκευές Bluetooth

Για την υλοποίηση των σεναρίων, χρησιμοποιήθηκαν δύο συσκευές όπου υποστηρίζουν το πρωτόκολλο Bluetooth. Ως συσκευή εκπομπής σήματος, χρησιμοποιήθηκε το CC2650 SensorTag της εταιρείας Texas Instruments. Για τη λήψη του σήματος, έγινε χρήση της συσκευής ESP32 – Thing, της εταιρείας SparkFun.

3.7.1 CC2650 SensorTag TI

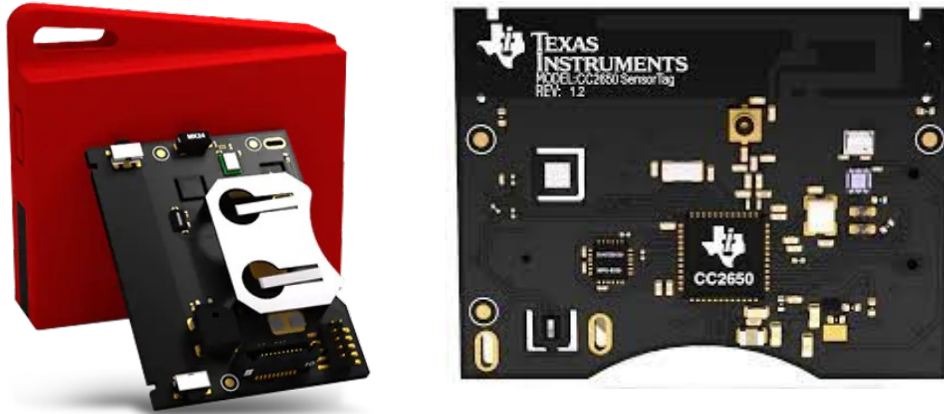
Το SensorTag [17], είναι μια συσκευή όπου περιλαμβάνει 10 αισθητήρες MEMS χαμηλής ισχύος σε μικρό μέγεθος. Το κιτ, είναι επεκτάσιμο μέσω του DevPacks, ώστε να μπορεί εύκολα να προστεθούν επιπλέον αισθητήρες ή ενεργοποιητές, (Εικόνα 10).

Μπορεί να συνδεθεί σε υποδομή νέφους μέσω Bluetooth®, ώστε να ληφθούν τα δεδομένα των αισθητήρων σε πραγματικό χρόνο. Το SensorTag είναι έτοιμο για χρήση με τις διαθέσιμες εφαρμογές τόσο σε iOS όσο και σε Android, χωρίς να απαιτείται εμπειρία προγραμματισμού.

Το SensorTag βασίζεται στον μικροελεγκτή (MCU) CC2650, προσφέροντας 75% χαμηλότερη κατανάλωση ενέργειας σε σχέση με άλλα προϊόντα Bluetooth. Αυτό επιτρέπει την πολυετή διάρκεια λειτουργίας, με μία μόνο μπαταρία τύπου cell. Το Bluetooth SensorTag περιλαμβάνει την τεχνολογία iBeacon, επιτρέποντας μέσω των προαναφερόμενων εφαρμογών, να εκκινεί εφαρμογές και να προσαρμόζει το

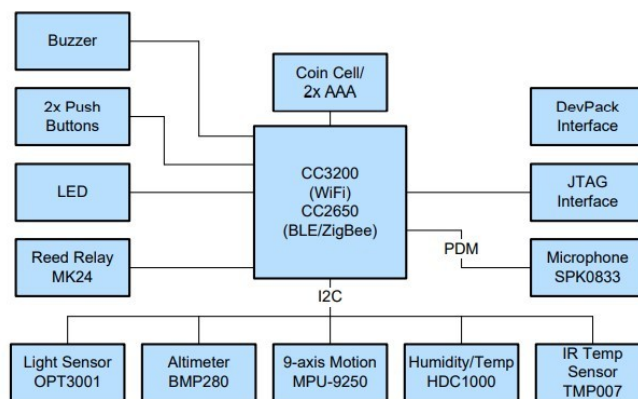
περιεχόμενο με βάση τα δεδομένα που λαμβάνει από το SensorTag, καθώς και τη φυσική του θέση.

Επιπλέον το SensorTag, μπορεί να χρησιμοποιηθεί και με τις τεχνολογίες ZigBee® και 6LoWPAN.

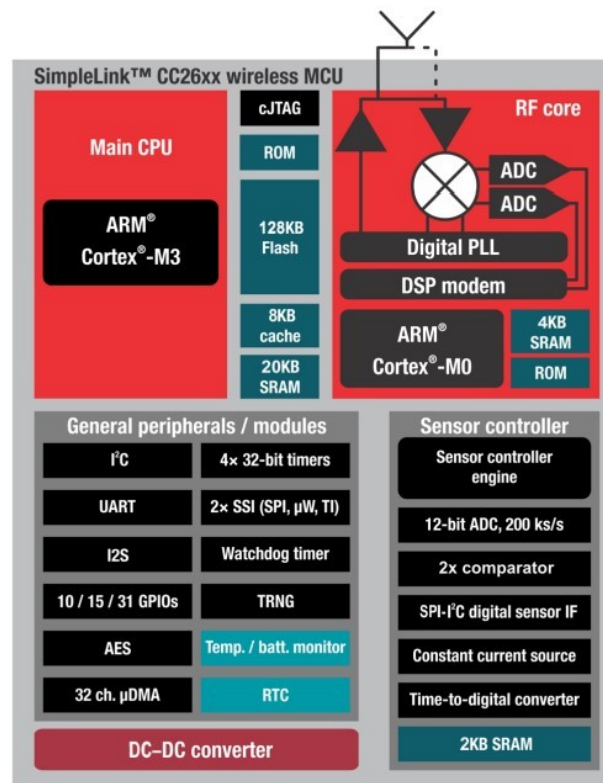


Εικόνα 10, Συσκευή SensorTag CC2650 TI

Στην εικόνα (Εικόνα 11), παρουσιάζεται το διάγραμμα των blocks που έχει το SensorTag, ενώ στην εικόνα (Εικόνα 12), παρουσιάζεται το λειτουργικό διάγραμμα των blocks που διαθέτει η συσκευή.



Εικόνα 11. Διάγραμμα των blocks στο SensorTag



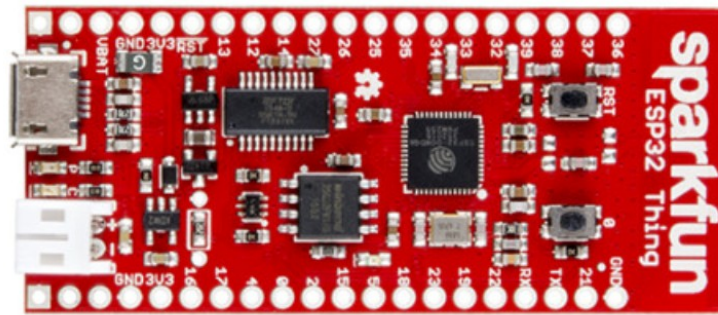
Εικόνα 12. Λειτουργικό διάγραμμα των blocks στο SensorTag

3.7.2 ESP32 -Thing Platform - SparkFun

Το ESP32 [18] (Εικόνα 13), είναι ένα chip 2,4 GHz Wi-Fi και Bluetooth που έχει σχεδιαστεί με TSMC στα 40 nm και τεχνολογία εξαιρετικά χαμηλής ισχύος. Είναι σχεδιασμένο και βελτιστοποιημένο για την καλύτερη δυνατή απόδοση ισχύος, απόδοση RF, στιβαρότητα, ευελιξία, χαρακτηριστικά και αξιοπιστία, για μια μεγάλη ποικιλία εφαρμογών και διαφορετικά προφίλ ισχύος.

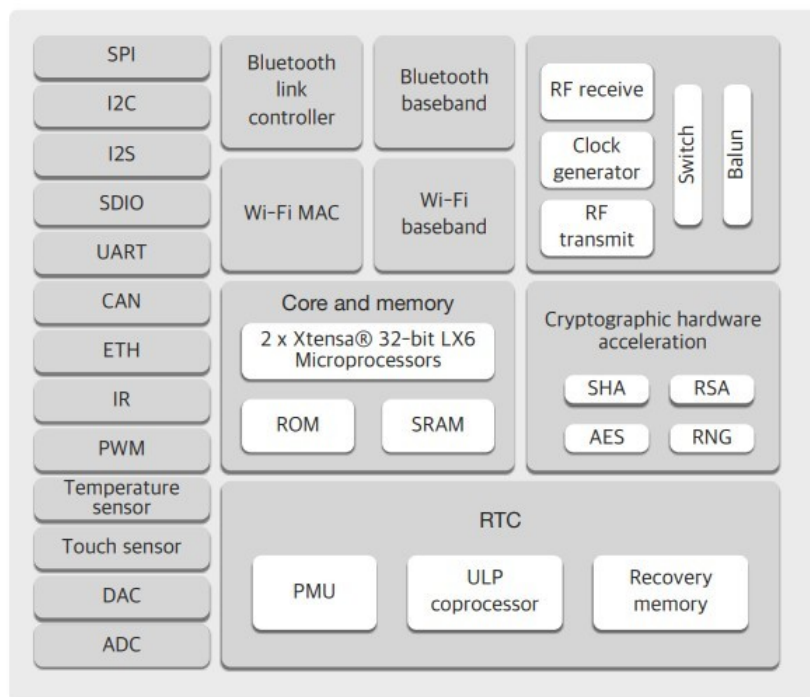
Το ESP32 είναι η πιο ολοκληρωμένη λύση για εφαρμογές Wi-Fi + Bluetooth στη βιομηχανία με λιγότερα από 10 εξωτερικά εξαρτήματα. Το ESP32 ενσωματώνει τον διακόπτη κεραίας, τον εξισωτή RF, τον ενισχυτή ισχύος, τον ενισχυτή λήψης χαμηλού θορύβου, τα φίλτρα και τις μονάδες διαχείρισης ισχύος. Ως εκ τούτου, ολόκληρη η λύση καταλαμβάνει ελάχιστη επιφάνεια σε πλακέτα τυπωμένου κυκλώματος (PCB).

Το ESP32 χρησιμοποιεί CMOS για πλήρως ενσωματωμένο ραδιόφωνο και βασική ζώνη ενός chip και ενσωματώνει επίσης προηγμένα κυκλώματα βαθμονόμησης που επιτρέπουν στη λύση να προσαρμόζεται δυναμικά για να απομακρύνει τις εξωτερικές ατέλειες του κυκλώματος ή να προσαρμόζεται στις αλλαγές των εξωτερικών συνθηκών.



Εικόνα 13. ESP32 SparkFun Thing

Στην παρακάτω εικόνα (Εικόνα 14), παρουσιάζεται το διάγραμμα των blocks του ESP32



Εικόνα 14. Διάγραμμα των blocks στο ESP32

3.8 MQTT Πρωτόκολλο

3.8.1 Τι είναι το πρωτόκολλο MQTT

Το πρωτόκολλο Message Queuing Telemetry Transport (MQTT) αποτελεί το πρότυπο ISO / IEC 20922: 2016 και είναι ένα πρωτόκολλο μεταφοράς μηνυμάτων για το μοντέλο εξυπηρετητή – πελάτη βασισμένο στο TCP/IP. Είναι ελαφρύ, ανοιχτό, απλό και σχεδιασμένο έτσι, ώστε να είναι εύκολο στην εφαρμογή. Αυτά τα χαρακτηριστικά το καθιστούν ιδανικό για χρήση σε πολλές περιπτώσεις, όπως περιορισμένα περιβάλλοντα, για επικοινωνία σε περιβάλλοντα μηχανής-με-μηχανή και IoT, όπου απαιτείται μικρό αποτύπωμα κώδικα, ή περιορισμένο εύρος ζώνης δικτύου. Το μοντέλο μηνυμάτων



δημοσίευσης / εγγραφής που χρησιμοποιεί, απαιτεί ένα ενδιάμεσο διαχειριστή μηνυμάτων (broker). Ο broker είναι υπεύθυνος για τη διανομή των μηνυμάτων στους πελάτες, με βασικό κριτήριο το θέμα του μηνύματος. Το πρωτόκολλο ανακαλύφθηκε από τους Δρ Andy Stanford – Clark της IBM και Arlen Nipper της Arccom το 1999. Το 2013, η IBM υπέβαλλε στο OASIS το MQTT v3.1, ενώ η τελευταία έκδοση του πρωτοκόλλου είναι η v5 (OASIS, 2019) [20].

3.8.2 Χαρακτηριστικά πρωτοκόλλου MQTT

Τα χαρακτηριστικά και οι κύριες δυνατότητες του πρωτοκόλλου MQTT είναι τα παρακάτω:

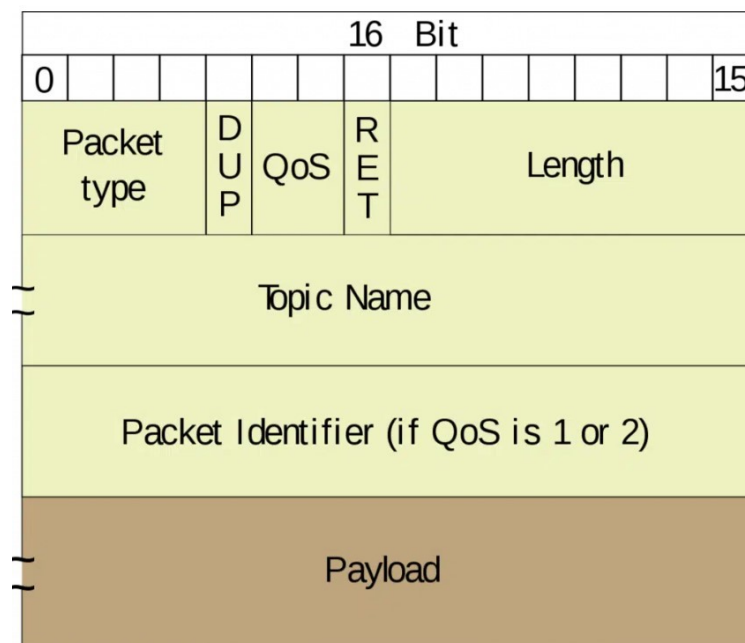
- Χρήση του μοντέλου «δημοσίευσης / εγγραφής μηνύματος», που παρέχει κατανομή μηνυμάτων «ένα-προς-πολλά» και αποσύνδεση των εφαρμογών.
- Διατήρηση της κατάστασης της συνεδρίας(session) μεταξύ πελάτη και εξυπηρετητή για όσο διάστημα διαρκεί η συνεδρία
- Αγνωστικιστική μεταφορά μηνυμάτων (που δεν έχει γνώση του περιεχόμενου των μεταδιδόμενων δεδομένων).
- Τρία επίπεδα αξιοπιστίας παράδοσης των μηνυμάτων:
 - "Το πολύ μία φορά", όπου ένα μήνυμα μεταδίδεται μια φορά και δεν εξασφαλίζεται η παράδοσή του, οπότε μπορεί να παρουσιαστεί απώλεια μηνυμάτων. Αυτό το επίπεδο θα μπορούσε να χρησιμοποιηθεί, για παράδειγμα, με δεδομένα αισθητήρων περιβάλλοντος, όπου δεν έχει σημασία αν χάνονται μεμονωμένες μετρήσεις, καθώς το επόμενο μήνυμα θα δημοσιευθεί σύντομα μετά.
 - "Τουλάχιστον μία φορά", όπου εξασφαλίζεται ότι ένα μήνυμα θα παραδοθεί, αλλά μπορεί να εμφανιστούν διπλότυπα.
 - "Ακριβώς μία φορά", όπου εξασφαλίζεται ότι ένα μήνυμα θα παραδοθεί ακριβώς μία φορά. Αυτό το επίπεδο θα μπορούσε να χρησιμοποιηθεί, για παράδειγμα, με συστήματα χρέωσης όπου διπλά ή χαμένα μηνύματα θα μπορούσαν να οδηγήσουν σε εσφαλμένες χρεώσεις
- Χρησιμοποιεί το πρωτόκολλο δικτύου επιπέδου μετάδοσης TCP/IP.
- Υποστηρίζει την ασύγχρονη ανταλλαγή μηνυμάτων.

- Παρέχει ασφαλείς μεταδόσεις μέσω του Transport Layer Security (TLS).

3.8.3 Δομή Μηνυμάτων (Πακέτων Ελέγχου) MQTT

Το πρωτόκολλο MQTT ορίζει συνολικά 14 διαφορετικά είδη πακέτων ελέγχου (μηνυμάτων), εκ των οποίων μόνο ένα (πακέτο PUBLISH) αφορά τη μετάδοση μηνυμάτων σε επίπεδο εφαρμογής (Application Messages). Ένα πακέτο ελέγχου (Εικόνα 15), αποτελείται από τρία τμήματα, με την ακόλουθη σειρά εμφάνισης:

- Σταθερή κεφαλίδα, υπάρχει σε όλα τα πακέτα ελέγχου
- Μεταβλητή κεφαλίδα, υπάρχει σε ορισμένα πακέτα ελέγχου
- Δεδομένα (payload), υπάρχει σε ορισμένα πακέτα ελέγχου



Εικόνα 15. Δομή Μηνυμάτων MQTT

3.8.4 Ασφάλεια Επικοινωνίας MQTT

Το πρωτόκολλο MQTT παρέχει πεδία ονόματος χρήστη και κωδικού πρόσβασης στο μήνυμα CONNECT για τον έλεγχο της ταυτότητας του πελάτη. Ο πελάτης έχει τη δυνατότητα να στείλει ένα όνομα χρήστη και έναν κωδικό πρόσβασης όταν συνδέεται με έναν εξυπηρετητή. Το όνομα χρήστη είναι μια κωδικοποιημένη συμβολοσειρά UTF-8. Ο κωδικός πρόσβασης είναι δυαδικά δεδομένα με μέγιστο μήκος 65535 bytes. Η προδιαγραφή MQTT δηλώνει ότι μπορείτε να σταλεί ένα όνομα χρήστη χωρίς κωδικό πρόσβασης, αλλά δεν είναι δυνατή η αποστολή ενός κωδικού πρόσβασης χωρίς όνομα



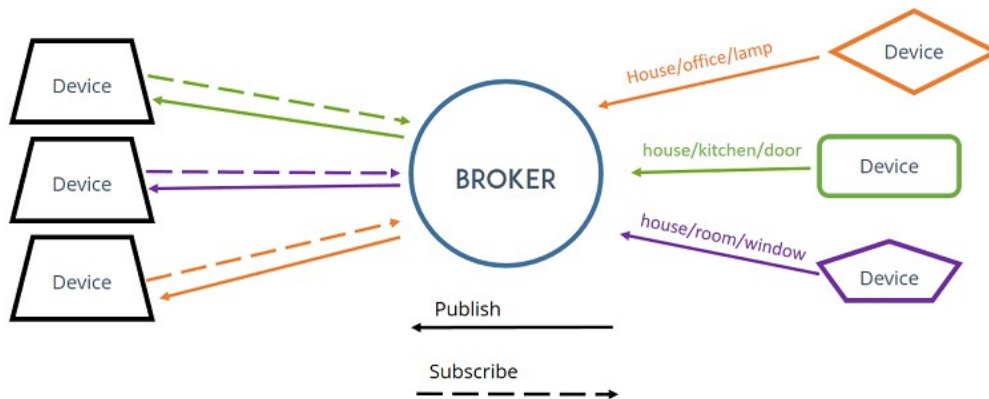
χρήστη. Σε προηγούμενες εκδόσεις του MQTT υπήρχε η σύσταση για κωδικούς πρόσβασης 12 χαρακτήρων, αλλά στην τρέχουσα έκδοση έχει καταργηθεί. Όταν χρησιμοποιείται ο ενσωματωμένος έλεγχος ταυτότητας με όνομα χρήστη και κωδικό πρόσβασης, ο εξυπηρετητής αξιολογεί τα διαπιστευτήρια βάσει του μηχανισμού ελέγχου ταυτότητας που εφαρμόζεται και επιστρέφει έναν από τους ακόλουθους κωδικούς επιστροφής:

1. 0 (0x00) – Η σύνδεση είναι αποδεκτή
2. 1 (0x01) – Η σύνδεση απορρίφθηκε, μη αποδεκτή έκδοση πρωτοκόλλου
3. 2 (0x02) – Η σύνδεση απορρίφθηκε, το αναγνωριστικό πελάτη δεν είναι αποδεκτό
4. 3 (0x03) – Η σύνδεση απορρίφθηκε, ενώ η διαδικτυακή σύνδεση ήταν επιτυχής, δεν υπάρχει MQTT εξυπηρετητής διαθέσιμος
5. 4 (0x04) – Η σύνδεση απορρίφθηκε, το όνομα χρήστη ή / και ο κωδικός πρόσβασης δεν είναι αποδεκτά
6. 5 (0x05) – Η σύνδεση απορρίφθηκε, ο πελάτης δεν είναι εξουσιοδοτημένος να συνδεθεί

Όταν το όνομα χρήστη και ο κωδικός πρόσβασης έχουν οριστεί στον πελάτη, οι πληροφορίες αποστέλλονται στον εξυπηρετητή σε μορφή απλού κείμενου. Αυτό το κείμενο είναι ευάλωτο στην υποκλοπή και παρέχει έναν εύκολο τρόπο για τους κακόβουλους να αποκτήσουν τα διαπιστευτήρια (credentials). Η ασφαλής μετάδοση των ονομάτων χρήστη και των κωδικών πρόσβασης απαιτεί κρυπτογράφηση σε επίπεδο μετάδοσης. Το MQTT ορίζει ότι μπορεί να χρησιμοποιηθεί κρυπτογράφηση μετάδοσης TLS (SSL), ώστε οι πληροφορίες που ανταλλάσσονται μεταξύ πελάτη και εξυπηρετητή (των συμπεριλαμβανομένων των διαπιστευτηρίων) να μην είναι εύκολο να υποκλαπούν. Εκτός από την κρυπτογράφηση των δεδομένων, το TLS προστατεύει και από την αλλοίωση των δεδομένων. Επιπλέον είναι εφικτό οι πελάτες να χρησιμοποιούν SSL πιστοποιητικά, ώστε να ταυτοποιούνται μέσω αυτών από τον εξυπηρετητή.

3.8.5 MQTT Broker

Ένας MQTT Broker είναι μια κεντρική οντότητα λογισμικού στην αρχιτεκτονική MQTT. Λειτουργεί ακριβώς όπως ένας μεσίτης ακινήτων, ο οποίος πρώτα κάνει ελέγχους ιστορικού για τα εμπλεκόμενα μέρη και στη συνέχεια, αφού βεβαιωθεί ότι εφαρμόζονται οι σχετικοί κανόνες, ο μεσίτης ξεκινά μια συναλλαγή όπου στη δική μας περίπτωση είναι η ανταλλαγή μηνυμάτων (Εικόνα 16).



Εικόνα 16. MQTT Broker

3.9 ATLAS – ATLAS Gateway

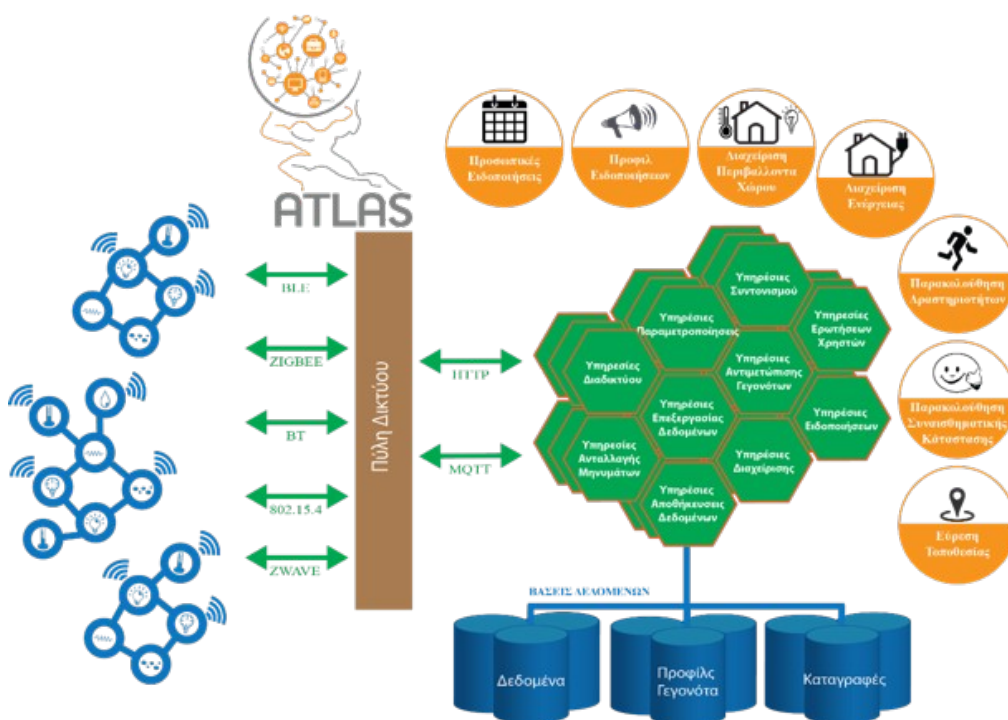
Η πλατφόρμα ATLAS [21], είναι μια ολοκληρωμένη Πλατφόρμα, που έχει ως σκοπό την από-άκρο-σε-άκρο διασύνδεση ετερογενών δικτύων αισθητήρων και την ανάπτυξη υπηρεσιών προς τους τελικούς χρήστες και έχει αναπτυχθεί από το Εργαστήριο Σχεδιασμού Ενσωματωμένων Συστημάτων και Εφαρμογών του τμήματος Μηχανικών Πληροφορικής του πρώην ΤΕΙ Δυτικής Ελλάδας.

Σε αντίθεση από τις κλασικές πλατφόρμες όπου η κύρια υλοποίηση είναι στην πλευρά του υπολογιστικού νέφους (cloud), η πλατφόρμα ATLAS (Εικόνα 17), αναπτύσσεται με σκοπό να παρέχει υλοποιήσεις όσο πιο κοντά γίνεται στο δίκτυο ασύρματων αισθητήρων, το οποίο επιτρέπει την άμεση ανίχνευση και αντιμετώπιση διαφόρων γεγονότων. Με την χρήση τοπικών συστάδων (clusters) που αποτελούνται από έξυπνες πύλες (gateways), η πλατφόρμα υποστηρίζει ετερογενή ασύρματα δίκτυα αισθητήρων, όπως για παράδειγμα, Bluetooth Low Energy, Zigbee.

Η επικοινωνία μεταξύ των πυλών (gateways) και του νέφους γίνεται με το πρωτόκολλο MQTT (Message Queuing Telemetry Transport), το οποίο είναι σχεδιασμένο για χαμηλών προδιαγραφών συσκευές και για επικοινωνίες με μικρές ταχύτητες. Επιπλέον, υπάρχει

και η δυνατότητα για την επικοινωνία μέσω του παραδοσιακού διαδικτυακού πρωτοκόλλου HTTP (Hypertext Transfer Protocol). Στην πλευρά του υπολογιστικού νέφους, η πλατφόρμα αποτελείται από μικρές υπηρεσίες διασυνδεδεμένες μεταξύ τους, βασισμένη στην αρχιτεκτονική μικρο-υπηρεσιών (microservices). Η συγκεκριμένη αρχιτεκτονική παρέχει στην πλατφόρμα ευελιξία, ως προς την υλοποίηση, το σχεδιασμό και την εύκολη και γρήγορη προσθήκη και αφαίρεση υπηρεσιών.

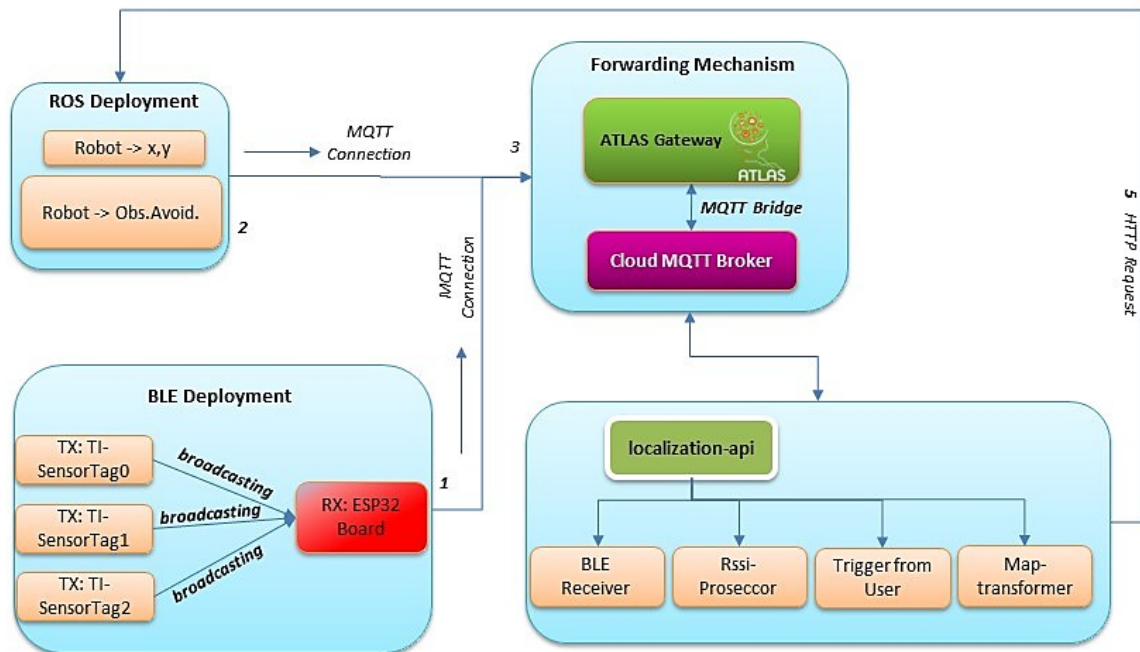
Τέλος η πλατφόρμα ATLAS παρέχει ολοκληρωμένα APIs (Application Programming Interfaces) και διαδικτυακές εφαρμογές, για τους τελικούς χρήστες ώστε να μπορούν χρησιμοποιήσουν την πλατφόρμα, να δημιουργήσουν καινούργιες υπηρεσίες, χωρίς να έχουν εξειδικευμένες γνώσεις προγραμματισμού, καθώς επίσης και να συνδέσουν την πλατφόρμα με άλλες παρόμοιες πλατφόρμες.



Εικόνα 17. Πλατφόρμα ATLAS

4 Αρχιτεκτονική και Υλοποίηση

Στην ακόλουθη εικόνα (Εικόνα 18), παρουσιάζεται η αρχιτεκτονική του υλοποιούμενου συστήματος.



Εικόνα 18. Αρχιτεκτονική συστήματος

5 Υλοποίηση της Αρχιτεκτονικής

5.1 Υλοποίηση σε επίπεδο ROS

Μια από τις πιο δημοφιλείς εφαρμογές του ROS είναι το SLAM [22] (ταυτόχρονος εντοπισμός και χαρτογράφηση). Ο στόχος του SLAM στη ρομποτική, είναι η κατασκευή και η ενημέρωση του χάρτη ενός ανεξερεύνητου περιβάλλοντος με τη βοήθεια των διαθέσιμων αισθητήρων που είναι προσαρτημένοι στο ρομπότ το οποίο θα χρησιμοποιηθεί για την εξερεύνηση. Οι αισθητήρες όπου χρησιμοποιούνται για την δημιουργία ενός χάρτη και αλλά και για την περιήγηση της ρομποτικής πλατφόρμας στον χώρο είναι είτε η κάμερα είτε το λέιζερ.

Από το ρομποτικό μηχανισμό και των πακέτων *Navigation* [23] και *gmapping* όπου υλοποιείται το *SLAM (Simultaneous Localization And Mapping)* χρησιμοποιώντας τον αισθητήρα του λέιζερ και κατ' επέκταση το *Laser Scanner* [24] πακέτο, συνδυάζουμε και εξάγουμε πληροφορία όπου μας χρησιμεύει για δύο πράγματα όπως:

1. Την τοποθεσία όπου βρίσκεται το ρομπότ επάνω στο χάρτη. Για την εξαγωγή της πληροφορίας αυτής, χρησιμοποιούμε το *topic: /odom*, όπου και παίρνουμε την πληροφορία για τις συντεταγμένες μέσω του χάρτη όπου έχει σχεδιάσει το

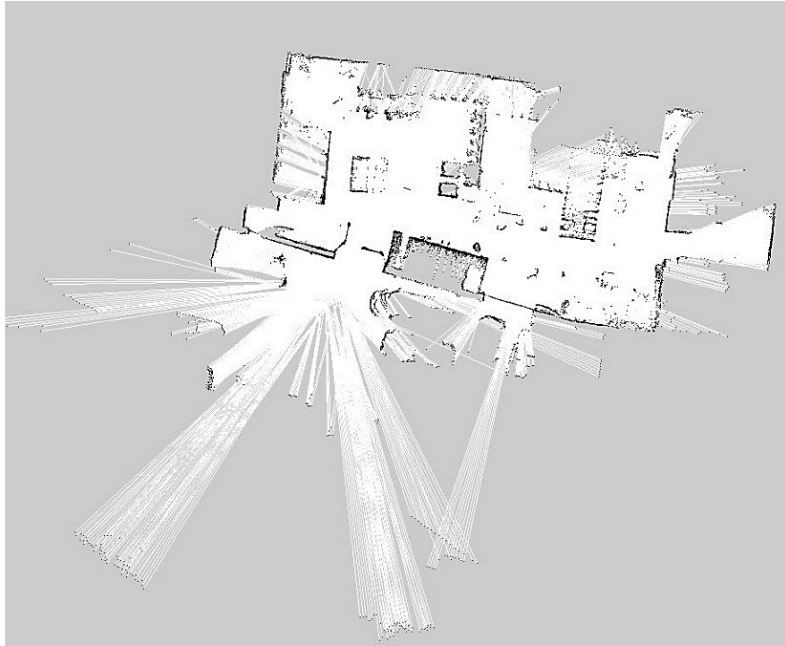
ρομπότ για το περιβάλλον όπου είναι τοποθετημένο. Με αυτόν τον τρόπο έχουμε την πληροφορία για το πού βρίσκεται το ρομπότ στον χώρο, (Εικόνα 19).

```
^Cheader:
  seq: 52387
  stamp:
    secs: 5238
    nsecs: 800000000
  frame_id: "odom"
  child_frame_id: ''
  pose:
    position:
      x: 9.98966876268
      y: -0.646511593475
      z: 0.0
    orientation:
      x: 0.0
      y: 0.0
      z: -0.0399893341866
      w: 0.999200106661
  covariance: [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0]
  twist:
    twist:
      linear:
        x: 0.1
        y: 0.0
        z: 0.0
      angular:
        x: 0.0
        y: 0.0
        z: 0.0
    covariance: [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0]
```

Εικόνα 19. Πληροφορία συντεταγμένων x,y

2. Βάση του σχεδιασμένου χάρτη η ρομποτική πλατφόρμα μας μπορεί να πλοηγείτε στον χώρο και να αποφεύγει τα εμπόδια όπου υπάρχουν σε αυτόν με την βέλτιστη διαδρομή όπου υπολογίζει. Αυτή η διαδικασία υποστηρίζεται σε δυναμικό περιβάλλον έτσι ώστε να αποφεύγονται και εμπόδια όπου μπορούν να εμφανιστούν – τοποθετηθούν την στιγμή όπου κινείται σε προδιαγεγραμμένη πορεία.

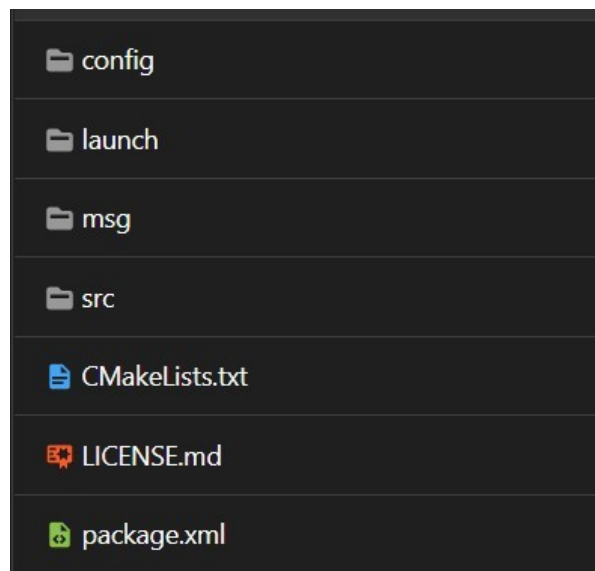
Ο λόγος όπου χρειαζόμαστε αυτή την λειτουργία, είναι για να μπορεί η ρομποτική πλατφόρμα μας να μπορεί να αποσταλεί οπουδήποτε στον χώρο και ταυτόχρονα να υπολογίζει αν είναι εφικτή η ολοκλήρωση της μετακίνησης. Αν δεν είναι εφικτό να πάει στην τοποθεσία όπου θέλουμε, τότε, η πλατφόρμα θα φτάσει μέχρι το κοντινότερο σημείο και θα σταματήσει, ελέγχοντας για πλησιέστερες διεξόδους (βάση της αρχικής γνώσης που διαθέτει από τον αρχικό χάρτη). Ένα παράδειγμα χάρτη που έχει ληφθεί από το ρομποτικό μηχανισμό, παρουσιάζεται ακολούθως στην (Εικόνα 20).



Εικόνα 20. Αποτύπωση χάρτη (TOBEA)

5.2 Αποστολή πληροφορίας

Για την αποστολή πληροφορίας από το ρομπότ στο υπόλοιπο σύστημα, χρησιμοποιήθηκε το πρωτόκολλο MQTT. Για τον σκοπό αυτό, υλοποιήθηκε κατάλληλο πακέτο ROS, με την ονομασία `ros_mqtt_subscriber`. Η δομή του πακέτου παρουσιάζεται στην (Εικόνα 21).



Εικόνα 21. Δομή πακέτου `ros_mqtt_subscriber`

5.3 Localization api

Το localization api είναι αυτό που αναλαμβάνει την διαχείριση των τιμών όπου συλλέγονται από το BLE Deployment. Από το σύνολο των τιμών όπου λαμβάνει,



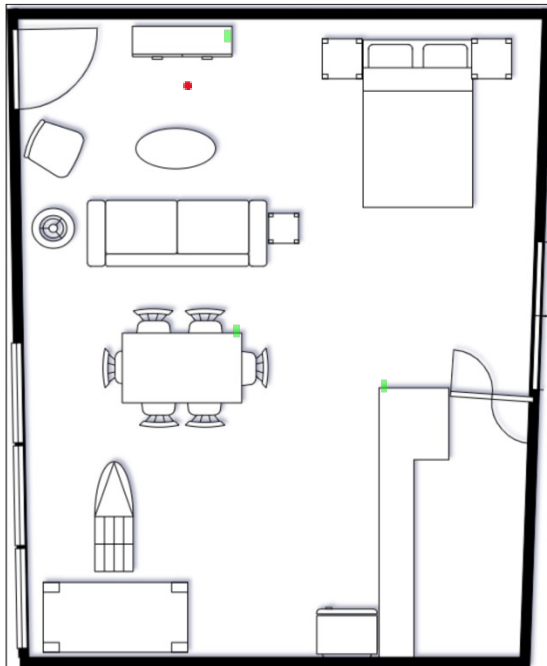
διαχωρίζει την πληροφορία σε αποδεκτή και μη. Οι αποδεκτές τιμές είναι αυτές όπου εμείς έχουμε ορίσει ότι αναζητούμε σε συγκεκριμένο χώρο, δηλαδή οι MAC διευθύνσεις από τα τρία BLE μαζί και οι τιμές του σήματος (RSSI) όπου έχουν.

Θα πρέπει να σημειωθεί πως τα BLE έχουν τοποθετηθεί σε συγκεκριμένα σημεία στον χώρο. Αυτά τα σημεία έχουν οριστεί και στο Localization api, με σκοπό να υπάρχει σημείο αναφοράς και για τα τρία σε σχέση με τη συσκευή ESP32 που χρησιμοποιείται για την αναγνώριση αυτών και που είναι προσαρτημένη στη ρομποτική πλατφόρμα όπου κινείται στον χώρο.

Αφού εκτελεστεί η εφαρμογή περιμένει από τον χρήστη να επιλέξει ένα από τα τρία BLE όπου θέλει η ρομποτική πλατφόρμα μας να προσεγγίσει. Όταν επιλέξει ο χρήστης το BLE, το σύστημα ελέγχει αν το RSSI του συγκεκριμένου λαμβάνεται από το σύστημα μας. Αν βρεθεί αυτή η τιμή στην λίστα με τις τιμές όπου λαμβάνει στέλνεται ένα αίτημα (command), μέσω του πρωτοκόλλου HTTP, έτσι ώστε να αποσταλεί το ρομπότ στην τοποθεσία όπου είναι η Bluetooth συσκευή μας.

Τέλος για να μπορούμε να έχουμε μία οπτική απεικόνιση σε δισδιάστατο περιβάλλον σε διάφορες συσκευές σχετικά με το πού βρίσκεται το ρομπότ μας στον χώρο, σχεδιάσαμε μια εικονική αναπαράσταση του χώρου. Για την δυναμική αναπαράσταση της ρομποτικής πλατφόρμας επάνω στο σχέδιο, χρησιμοποιήθηκε η τεχνική του Canvas element [25].

Η αναπαράσταση αυτή παρουσιάζεται στην παρακάτω εικόνα (Εικόνα 22). Οι πράσινες στίξεις αναπαριστούν τα σημεία όπου είναι τοποθετημένα τα BLE.



Device Type	Device ID	Position
robot	urn:esda:atlas:robot:gazebo001	{"device": "urn:esda:atlas:robot:gazebo001", "type": "robot", "position": {"x": 889, "y": 730}}

Εικόνα 22. Αναπαράσταση οικίας σε δισδιάστατο περιβάλλον

Στον πίνακα κάτω από την αναπαράσταση του χώρου λαμβάνουμε από το ΑΤΛΑΣ την πληροφορία για το ποια συσκευή είναι συνδεδεμένη, όπως επίσης και τις τιμές που μας επιστρέφει. Στην συγκεκριμένη περίπτωση αναφέρεται πως η συσκευή που έχει συνδεθεί είναι το ρομπότ και η τοποθεσία που βρίσκεται είναι πάνω στον χώρο όπου έχει σχεδιάσει είναι στο $x:889$ και $y:730$. Αυτή η τιμή στον χάρτη μας αναπαρίσταται από την κόκκινη κουκίδα.

5.3.1 Προσθήκη του βραχίονα

Για την ολοκλήρωση της λειτουργικότητας του συστήματος, προστίθεται ο ρομποτικός βραχίονας WidowX Arm. Η τοποθέτηση του έγινε στο ανώτερο επίπεδο της ρομποτικής πλατφόρμας και δόθηκε τροφοδοσία από την υπάρχουσα αναμονή της βάσης, (kobuki). Για τη λειτουργικότητα του ρομποτικού βραχίονα, υλοποιήθηκε κατάλληλο ROS πακέτο. Η συνδυασμένη χρήση του με την λαμβανόμενη πληροφορία από την εφαρμογή όπου αποτυπώνει την τοποθεσία του ρομπότ σε σχέση με το περιβάλλον, ενεργοποιεί τη χρήση του, όταν επιτευχθεί η προσέγγιση του στόχου όπου επιλέχθηκε από τον χρήστη.



Συνοψίζοντας, όταν φτάσει η πλατφόρμα στον προορισμό όπου έχει αποσταλεί δίνει εντολή στον βραχίονα να ξεκινήσει το σενάριο όπου από την αρχική θέση όπου είναι τοποθετημένος, εκτείνεται για να προσεγγίσει και τέλος να πιάσει το αντικείμενο όπου είναι τοποθετημένο μπροστά του. Αφού έχει πιάσει το αντικείμενο το σηκώνει και επανέρχεται στην αρχική του θέση κρατώντας το. Έπειτα το ρομπότ ξεκινάει να πάει στον επόμενο προορισμό. Όταν φτάσει δίνεται ξανά εντολή στον βραχίονα με την διαφορά πως σε αυτή την περίπτωση αφήνει το αντικείμενο όπου έχει συλλέξει από το προηγούμενο σενάριο που εκτέλεσε για τη λήψη του αντικειμένου.

Θα πρέπει να σημειωθεί πως μετά από πολλές δοκιμές παρατηρήθηκε πως λόγω της μειούμενης ισχύος όπου παρείχε το kobuki, ο βραχίονας άρχιζε και υπολειτουργούσε. Επίσης, παρατηρήθηκε πως ο βραχίονας κινούταν πιο αργά, και πως όταν έπιανε το αντικείμενο για να το σηκώσει από την επιφάνεια η κίνηση δεν ήταν ομαλή όπως όταν ήταν συνδεδεμένος με το τροφοδοτικό του.

6 Μελλοντική εργασία

Το επόμενο βήμα σε αυτή την υλοποίηση είναι με την υπάρχουσα αρχιτεκτονική όπου λαμβάνονται τα δεδομένα από τις BLE συσκευές να χρησιμοποιούμε την πληροφορία αυτή για να μπορέσουμε να προσεγγίζουμε τις συσκευές ανάλογα με την ένταση του σήματός τους. Έτσι έχοντας ως σημείο αναφοράς τα σημεία όπου είναι καταχωρημένες οι συσκευές μας, μπορούμε να τις αναζητήσουμε στον χώρο με την ρομποτική μας πλατφόρμα ανεξαρτήτου το πού βρίσκονται στον χώρο μετατοπισμένες.

Επίσης θα πρέπει να λυθεί το πρόβλημα με την παροχή ενέργειας στον ρομποτικό βραχίονα για να μπορεί να λειτουργεί σωστά παρέχοντάς μας την πλήρη λειτουργικότητά του. Τέλος είτε με την κάμερα όπου διαθέτει το turtlebot είτε με μία επιπλέον κάμερα να ξεχωρίζονται τα αντικείμενα από το ρομπότ έτσι ώστε η πίεση όπου θα ασκείτε από το βραχίονα για να τα πιάσει και να τα διαχειριστεί, να μην είναι λιγότερη αλλά και σίγουρα όχι περισσότερη μιας και αυτό το ενδεχόμενο μπορεί να προκαλέσει ζημιά στο αντικείμενο αλλά και στον βραχίονα.



Βιβλιογραφία – Πηγές

[1] Open-source robot development kit for apps on wheels, TurtleBot2,
<https://www.turtlebot.com/turtlebot2/>

[2] iClebo Kobuki, <http://kobuki.yujinrobot.com/about2/>

[3] Intel® NUC, <https://www.intel.com/content/www/us/en/products/details/nuc.html>



- [4] Compact, lightweight obstacle detection sensor Hokuyo UST10LX, <https://hokuyo-usa.com/products/lidar-obstacle-detection/ust-10lx>
- [5] Orbbec Astra Camera, <https://orbbec3d.com/index/products.html>
- [6] WidowX Robot Arm Kit, <https://www.trossenrobotics.com/widowxrobotarm>
- [7] Robotic Operating System, <http://www.ros.org/>
- [8] Gazebo Simulator, <http://gazebosim.org/>
- [9] Open Dynamics Engine, <https://www.ode.org/>
- [10] OpenGL (Open Graphics Library), <https://www.opengl.org/>
- [11] Catkin build, <https://catkin-tools.readthedocs.io/en/latest/>
- [12] Bluetooth Protocol, <https://www.bluetooth.com/>
- [13] Bluetooth Low Energy Protocol, https://en.wikipedia.org/wiki/Bluetooth_Low_Energy
- [14] Piconet in Bluetooth, <https://en.wikipedia.org/wiki/Piconet>
- [15] Scatternet in Bluetooth, <https://en.wikipedia.org/wiki/Scatternet>
- [16] Protocols in the Bluetooth Protocol Stack, <https://www.tutorialspoint.com/the-bluetooth-protocol-stack>
- [17] Texas Instruments CC2650 SensorTag, <https://www.ti.com/tool/TIDC-CC2650STK-SENSORTAG#description>
- [18] ESP32 Thing, https://cdn.sparkfun.com/datasheets/IoT/esp32_datasheet_en.pdf
- [19] MQTT: The Standard for IoT Messaging, <https://mqtt.org/>
- [20] MQTT Version 5.0, <https://docs.oasis-open.org/mqtt/mqtt/v5.0/mqtt-v5.0.html>
- [21] ATLAS Platform, <https://esdalab.ece.uop.gr/index.php/el/research-and-development/research-platforms>
- [22] Simultaneous localization and mapping (SLAM), https://en.wikipedia.org/wiki/Simultaneous_localization_and_mapping
- [23] Navigation in ROS, <https://navigation.ros.org/>
- [24] Laser-based SLAM (Simultaneous Localization and Mapping), <http://wiki.ros.org/gmapping>
- [25] Canvas Element, [https://en.wikipedia.org/wiki/Canvas_\(GUI\)](https://en.wikipedia.org/wiki/Canvas_(GUI))
- [26] AAL House, <https://aalhouse.esdalab.ece.uop.gr/>
- [27] Εργαστήριο Σχεδιασμού Ενσωματωμένων Συστημάτων και εφαρμογών, <https://www.esdalab.ece.uop.gr/index.php/el/>