



Πανεπιστήμιο Πειραιώς – Τμήμα Πληροφορικής

Πρόγραμμα Μεταπτυχιακών Σπουδών

«Προηγμένα Συστήματα Πληροφορικής

Μεταπτυχιακή Διατριβή

Τίτλος Διατριβής	Ανάπτυξη συσκευής εισόδου με χρήση τεχνολογίας Bluetooth Low Energy και αισθητηρίων αδράνειας Input device development by the use of BLE Technology and inertia sensors.
Όνοματεπώνυμο Φοιτητή	Τζανιδάκης Μάριος
Πατρώνυμο	Τζανής
Αριθμός Μητρώου	ΜΠΣΠ12077
Επιβλέπων	Ψαράκης Μιχαήλ, Επίκουρος Καθηγητής

Ημερομηνία Παράδοσης **Ιούλιος 2015**

Τριμελής Εξεταστική Επιτροπή

(υπογραφή)

(υπογραφή)

(υπογραφή)

Μ. Ψαράκης
Επίκουρος Καθηγητής

Α. Πικράκης
Επίκουρος Καθηγητής

Π. Κοτζανικολάου
Επίκουρος Καθηγητής
Βαθμίδα

Περιεχόμενα

Περιεχόμενα.....	3
Περίληψη.....	96
Abstract.....	96
Εισαγωγή.....	107
1 Κεφάλαιο 1 - Bluetooth Low Energy (BLE).....	129
1.1 Bluetooth Core Specification.....	129
1.2 Bluetooth και Bluetooth Low Energy.....	1310
1.3 Κύρια δομικά στοιχεία του BLE.....	1411
1.3.1 Physical Layer.....	1512
1.3.2 Link Layer.....	1613
1.3.3 Host Controller Interface (HCI).....	1744
1.3.4 Logical Link Control and Adaptation Protocol (L2CAP).....	1815
1.3.5 Attribute Protocol (ATT).....	1916
1.3.6 Security Manager (SM).....	1916
1.3.7 Generic Attribute Profile (GATT).....	2118
1.3.8 Generic Access Profile (GAP).....	2118
1.4 Generic Access Profile Protocol (GAP).....	2118
1.4.1 Ρόλοι (Roles).....	2219
1.4.2 Λειτουργίες και διαδικασίες (Modes and Procedures).....	2219
1.4.3 Ασφάλεια.....	2421
1.4.4 Επιπλέον λειτουργίες του πρωτοκόλλου GAP.....	2623
1.5 Generic Attribute Profile Protocol (GATT).....	2724
1.5.1 Ρόλοι (Roles).....	2724
1.5.2 Γνωρίσματα (Attributes).....	2724
1.5.3 Ιεραρχία Δεδομένων.....	2926
1.5.4 Επιπλέον λειτουργίες γνωρισμάτων.....	3330
1.5.5 Λειτουργίες πρωτοκόλλου GATT.....	3330
2 Κεφάλαιο 2 – HID (Human Interface Device).....	3633
2.1 HID πάνω από USB.....	3633

2.2	Περιγραφείς αναφοράς (Report Descriptors).....	3835
2.3	HID με χρήση σύνδεσης Bluetooth.....	3936
2.3.1	Διαδικασίες HID με χρήση σύνδεσης Bluetooth.....	4037
2.3.2	Ανίχνευση υπηρεσιών (Service Discovery).....	4239
2.3.3	Ανίχνευση χαρακτηριστικών (Characteristic Discovery).....	4239
3	Κεφάλαιο 3 – Αισθητήρες Αδράνειας (Inertia Measurement Units).....	4542
3.1	Εξαρτήματα Αισθητήριων Αδράνειας.....	4643
3.1.1	Επιταχυνσιόμετρο (Accelerometer).....	4643
3.1.2	Γυροσκόπιο (Gyroscope).....	4845
3.1.3	Μαγνητόμετρο (Magnetic Field Sensor).....	5047
3.2	Αισθητήριο Αδράνειας GY-88.....	5249
3.3	Αισθητήριο MPU-6050.....	5350
3.3.1	MPU-6050 Χαρακτηριστικά Επιταχυνσιόμετρου.....	5350
3.3.2	MPU-6050 Χαρακτηριστικά Γυροσκοπίου.....	5350
3.3.3	MPU-6050 Sensor Fusion Engine.....	5451
3.3.4	MPU-6050 Επιπλέον χαρακτηριστικά.....	5451
4	Κεφάλαιο 4 – Ολοκληρωμένο DA14580 και Development Kit.....	5754
4.1	DA14580 chip.....	5754
4.1.1	Γενική Περιγραφή.....	5754
4.1.2	Επεξεργαστής ARM CORTEX M0.....	5855
4.1.3	Bluetooth Smart Τμήματα.....	5956
4.1.4	Μνήμες.....	5956
4.1.5	Λειτουργικές Καταστάσεις.....	5956
4.1.6	Ενεργειακές Καταστάσεις.....	6057
4.1.7	Διεπαφές (Interfaces).....	6057
4.1.8	Χρονιστές (Timers).....	6158
4.1.9	Ρολόι.....	6158
4.1.10	Διαχείριση Ενέργειας.....	6158
4.2	DA14580 Development Kit και Software Development Kit.....	6158
4.2.1	Hardware Development Kit.....	6158

4.2.2	Software Development Kit	6360
5	Κεφάλαιο 5 – Αναλυτική λειτουργία της συσκευής	6764
5.1	Υλοποίηση Υλικού	6764
5.2	Εκκίνηση εφαρμογής	6966
5.3	Κεντρικός Μηχανισμός Λειτουργίας	6966
5.4	Μηχανισμός Ουράς για τις αναφορές προς τον Host	7269
5.5	Μηχανισμός κίνησης του κέρσορα	7370
5.6	Χρονισμός συσκευής	7471
5.7	Μηχανισμός απομόνωσης και ανίχνευσης χειρονομιών	7471
5.8	Μηχανισμός μείωσης κατανάλωσης ενέργειας	7673
6	Κεφάλαιο 6 – Συμπεράσματα και μελλοντική έρευνα	8077
7	Βιβλιογραφία	8279
8	ΠΑΡΑΡΤΗΜΑ Α : User Manual DA14580 Gesture Ring	8481
9	ΠΑΡΑΡΤΗΜΑ Β : Hardware Overview	8582
10	ΠΑΡΑΡΤΗΜΑ Γ : Application Source Code	8784
11	ΠΑΡΑΡΤΗΜΑ Δ : Auxiliary Source	9693
12	ΠΑΡΑΡΤΗΜΑ Ε : Auxiliary Source Code Configurations	10198
13	ΠΑΡΑΡΤΗΜΑ ΣΤ : Auxiliary Source code Files	10299
	Περιεχόμενα	3
	Περίληψη	6
	Abstract	6
	Εισαγωγή	7
1	Κεφάλαιο 1 – Bluetooth Low Energy (BLE)	9
1.1	Bluetooth Core Specification	9
1.2	Bluetooth και Bluetooth Low Energy	10
1.3	Κύρια δομικά στοιχεία του BLE	11
1.3.1	Physical Layer	12
1.3.2	Link Layer	13
1.3.3	Host Controller Interface (HCI)	14
1.3.4	Logical Link Control and Adaptation Protocol (L2CAP)	15

Formatted: Default Paragraph Font, Check spelling and grammar

Formatted: Default Paragraph Font, Check spelling and grammar

Formatted: Default Paragraph Font, Check spelling and grammar

Formatted: Default Paragraph Font, Check spelling and grammar

Formatted: Default Paragraph Font, Check spelling and grammar

Formatted: Default Paragraph Font, Check spelling and grammar

Formatted: Default Paragraph Font, Check spelling and grammar

Formatted: Default Paragraph Font, Check spelling and grammar

Formatted: Default Paragraph Font, Check spelling and grammar

Formatted: Default Paragraph Font, Check spelling and grammar

Formatted: Default Paragraph Font, Check spelling and grammar

Formatted: Default Paragraph Font, Check spelling and grammar

Formatted: Default Paragraph Font, Check spelling and grammar

Formatted: Default Paragraph Font, Check spelling and grammar

Formatted: Default Paragraph Font, Check spelling and grammar

Formatted: Default Paragraph Font, Check spelling and grammar

Formatted: Default Paragraph Font, Check spelling and grammar

Formatted: Default Paragraph Font, Check spelling and grammar

Formatted: Default Paragraph Font, Check spelling and grammar

Formatted: Default Paragraph Font, Check spelling and grammar

1.3.5	Attribute Protocol (ATT).....	16	Formatted	...
1.3.6	Security Manager (SM).....	16	Formatted	...
1.3.7	Generic Attribute Profile (GATT).....	18	Formatted	...
1.3.8	Generic Access Profile (GAP).....	18	Formatted	...
1.4	Generic Access Profile Protocol (GAP).....	18	Formatted	...
1.4.1	Ρόλοι (Roles).....	19	Formatted	...
1.4.2	Λειτουργίες και διαδικασίες (Modes and Procedures).....	19	Formatted	...
1.4.3	Ασφάλεια.....	21	Formatted	...
1.4.4	Επιπλέον λειτουργίες του πρωτοκόλλου GAP.....	23	Formatted	...
1.5	Generic Attribute Profile Protocol (GATT).....	24	Formatted	...
1.5.1	Ρόλοι (Roles).....	24	Formatted	...
1.5.2	Γνωρίσματα (Attributes).....	24	Formatted	...
1.5.3	Ιεραρχία Δεδομένων.....	26	Formatted	...
1.5.4	Επιπλέον λειτουργίες γνωρισμάτων.....	30	Formatted	...
1.5.5	Λειτουργίες πρωτοκόλλου GATT.....	30	Formatted	...
2	Κεφάλαιο 2 – HID (Human Interface Device).....	33	Formatted	...
2.1	HID πάνω από USB.....	33	Formatted	...
2.2	Περιγραφείς αναφοράς (Report Descriptors).....	35	Formatted	...
2.3	HID με χρήση σύνδεσης Bluetooth.....	36	Formatted	...
2.3.1	Διαδικασίες HID με χρήση σύνδεσης Bluetooth.....	37	Formatted	...
2.3.2	Ανίχνευση υπηρεσιών (Service Discovery).....	39	Formatted	...
2.3.3	Ανίχνευση χαρακτηριστικών (Characteristic Discovery).....	39	Formatted	...
3	Κεφάλαιο 3 – Αισθητήρες αδράνειας (Inertia Measurement Units).....	42	Formatted	...
3.1	Εξαρτήματα Αισθητήριων Αδράνειας.....	43	Formatted	...
3.1.1	Επιταχυνσιόμετρο (Accelerometer).....	43	Formatted	...
3.1.2	Γυροσκόπιο (Gyroscope).....	45	Formatted	...
3.1.3	Μαγνητόμετρο (Magnetic Field Sensor).....	47	Formatted	...
3.2	Αισθητήριο Αδράνειας GY 88.....	49	Formatted	...
3.3	Αισθητήριο MPU 6050.....	50	Formatted	...
3.3.1	MPU 6050 Χαρακτηριστικά Επιταχυνσιόμετρου.....	50	Formatted	...

3.3.2	MPU-6050 Χαρακτηριστικά Γυροσκοπίου	50	Formatted	...
3.3.3	MPU-6050 Sensor Fusion Engine	51	Formatted	...
3.3.4	MPU-6050 Επιπλέον χαρακτηριστικά	51	Formatted	...
4	Κεφάλαιο 4 – Ολοκληρωμένο DA14580 και Development Kit	54	Formatted	...
4.1	DA14580 chip	54	Formatted	...
4.1.1	Γενική Περιγραφή	54	Formatted	...
4.1.2	Επεξεργαστής ARM CORTEX M0	55	Formatted	...
4.1.3	Bluetooth Smart Τμήματα	56	Formatted	...
4.1.4	Μνήμες	56	Formatted	...
4.1.5	Λειτουργικές Καταστάσεις	56	Formatted	...
4.1.6	Ενεργειακές Καταστάσεις	57	Formatted	...
4.1.7	Διεπαφές (Interfaces)	57	Formatted	...
4.1.8	Χρονιστές (Timers)	58	Formatted	...
4.1.9	Ρολόι	58	Formatted	...
4.1.10	Διαχείριση Ενέργειας	58	Formatted	...
4.2	DA14580 Development Kit και Software Development Kit	58	Formatted	...
4.2.1	Hardware Development Kit	58	Formatted	...
4.2.2	Software Development Kit	60	Formatted	...
5	Κεφάλαιο 5 – Αναλυτική λειτουργία της συσκευής	64	Formatted	...
5.1	Υλοποίηση Υλικού	64	Formatted	...
5.2	Εκκίνηση εφαρμογής	67	Formatted	...
5.3	Κεντρικός Μηχανισμός Λειτουργίας	67	Formatted	...
5.4	Μηχανισμός Ουράς για τις αναφορές προς τον Host	70	Formatted	...
5.5	Μηχανισμός κίνησης του κέρσορα	71	Formatted	...
5.6	Χρονισμός συσκευής	72	Formatted	...
5.7	Μηχανισμός απομόνωσης και ανίχνευσης χειρονομιών	72	Formatted	...
5.8	Μηχανισμός μείωσης κατανάλωσης ενέργειας	74	Formatted	...
6	Κεφάλαιο 6 – Συμπεράσματα και μελλοντική έρευνα	78	Formatted	...
7	Βιβλιογραφία	80	Formatted	...
8	ΠΑΡΑΡΤΗΜΑ Α : User Manual DA14580 Gesture Ring	82	Formatted	...

9	ΠΑΡΑΡΤΗΜΑ Β : Hardware Overview	83
10	ΠΑΡΑΡΤΗΜΑ Γ : Application Source Code	85
11	ΠΑΡΑΡΤΗΜΑ Δ : Auxiliary Source	94
12	ΠΑΡΑΡΤΗΜΑ Ε : Auxiliary Source Code Configurations	99
13	ΠΑΡΑΡΤΗΜΑ ΣΤ : Auxiliary Source code Files	100

Formatted: Default Paragraph Font, Check spelling and grammar

Formatted: Default Paragraph Font, Check spelling and grammar

Formatted: Default Paragraph Font, Check spelling and grammar

Formatted: Default Paragraph Font, Check spelling and grammar

Formatted: Default Paragraph Font, Check spelling and grammar

Formatted: Default Paragraph Font, Check spelling and grammar

Formatted: Default Paragraph Font, Check spelling and grammar

Formatted: Default Paragraph Font, Check spelling and grammar

Formatted: Default Paragraph Font, Check spelling and grammar

Formatted: Default Paragraph Font, Check spelling and grammar

Περίληψη

Σκοπός της παρούσας εργασίας είναι η ανάπτυξη μίας συσκευής εισόδου κατάδειξης (ποντίκι) με την χρήση αισθητηρίων αδράνειας. Η συσκευή δεν φέρει κουμπιά για την εκτέλεση των εντολών καθώς επίσης δεν απαιτεί επιφάνεια κύλισης για την οδήγηση του κέρσορα. Έχει σχήμα δαχτυλιδιού προκειμένου να είναι εύκολο στην τοποθέτηση και ο χρήστης δίνει εντολές για κίνηση του κέρσορα απλά κουνώντας το χέρι. Για την εκτέλεση εντολών (πάτημα κουμπιού) το δαχτυλίδι είναι εξοπλισμένο με έναν αλγόριθμο αναγνώρισης κίνησης που επιτρέπει στον χρήστη απλά με ένα κούνημα του δαχτύλου να εκτελέσει την ενέργεια που θέλει. Γίνεται χρήση του πρωτοκόλλου HID προκειμένου η συσκευή να είναι συμβατή με όλα τα λειτουργικά συστήματα. Για την επικοινωνία με τον host χρησιμοποιήθηκε τεχνολογία BLE (Bluetooth Low Energy) προκειμένου να δώσει στην συσκευή εύκολη διασύνδεση καθώς και μεγάλη αυτονομία.

Abstract

The purpose of this paper is to describe the development of a pointing input device (mouse like device) by the use of inertia sensors. The device does not carry buttons in order to execute the ordinary commands of a mouse and also it doesn't require a solid surface for the movement of the cursor. It's a ring shaped gadget for ease in placement and the user can move the cursor of the host just by moving his arm. For command execution (left click) the ring is equipped with a gesture recognition algorithm which enables him to execute his command with a sudden movement of his finger. For the device's interface, HID protocol has been used in order to enhance the interoperability of the device with all the major operating systems. For the communication with the host, BLE Technology is used providing ease in interconnection and large self-sufficiency.

Εισαγωγή

Την τελευταία δεκαετία ένας από τους όρους που κυριαρχούν στον τομέα της τεχνολογίας είναι το Internet of things (IoT). Στην ουσία αυτό που περιγράφει ο τίτλος της νέας αυτής τάσης είναι ένα σενάριο σύμφωνα με το οποίο καθημερινές συσκευές, εξοπλισμένες με τα απαραίτητα ηλεκτρονικά εξαρτήματα, που χρησιμοποιούνται από ανθρώπους, από κλειδιά μέχρι αυτοκίνητα, θα έχουν την δυνατότητα να παράγουν δεδομένα και να τα προωθούν στο διαδίκτυο.



Το σενάριο αυτό υφίσταται από την δεκαετία του 80 περίπου, παρόλο που το συγκεκριμένο όνομα δόθηκε το 1999. Σαν μια πρώτη αναφορά του νέου αυτού σεναρίου θα μπορούσαμε να πούμε πως ήταν το μηχανήμα της Coca Cola στο πανεπιστήμιο του Carnegie Mellon στις αρχές του 80. Ο ενδιαφερόμενος μπορούσε να συνδεθεί στο μηχανήμα μέσω δικτύου και να ελέγξει την διαθεσιμότητα αναψυκτικών.

Σκοπός αυτής της νέας τεχνολογίας είναι να αντιμετωπίσει, κατά κάποιο τρόπο, το πρόβλημα της έλλειψης προσοχής σε λεπτομέρεια και ακρίβεια που αντιμετωπίζουν οι άνθρωποι λόγω περιορισμένου χρόνου. Εξοπλίζοντας καθημερινά αντικείμενα με τα κατάλληλα εξαρτήματα, τα οποία δεν θα χρειάζονταν καμία ανθρώπινη παρέμβαση για να συλλέξουν δεδομένα, θα μπορούσαμε να μετρήσουμε και να εντοπίσουμε οτιδήποτε. Θα ήμασταν σε θέση να μειώσουμε σημαντικά την σπατάλη, το κόστος καθώς και το πότε θα πρέπει κάτι να αντικατασταθεί ή να επισκευαστεί.

Στην ανάπτυξη αυτής της νέας τάσης έχουν συμβάλει και νέες τεχνολογίες όπως το BLE (Bluetooth Low Energy). Το γεγονός ότι προσφέρει ασύρματη επικοινωνία με μικρή κατανάλωση κάνει το BLE ιδανικό για μικροσυσκευές με περιορισμένους πόρους που χρειάζεται να λειτουργούν για μεγάλη χρονική περίοδο.

Στηρίζόμενοι στην τάση του IoT σκοπός της εργασίας είναι η ανάπτυξη μιας συσκευής εισόδου που θα λειτουργεί σαν «ποντίκι». Η συσκευή μπορεί να φορευθεί ως δαχτυλίδι και δεν φέρει καθόλου κουμπιά όπως ένα συνηθισμένο «ποντίκι». Η κίνηση στον κέρσορα δίνεται από ένα γυροσκόπιο, έτσι ακολουθεί την κίνηση του χεριού του χρήστη. Όσον αφορά τις διάφορες ενέργειες που θέλει να εκτελέσει ο χρήστης, το δαχτυλίδι είναι εξοπλισμένο με ένα αισθητήριο επιτάχυνσης και καταγράφει όποια κίνηση του δαχτύλου. Με την χρήση ενός αλγορίθμου αναγνώρισης χρονοσειρών, η συσκευή μπορεί να εντοπίσει και να ταυτοποιήσει αν η χειρονομία που έκανε ο χρήστης ταιριάζει με κάποια από τις χειρονομίες που μπορεί να αναγνωρίσει το δαχτυλίδι και στην συνέχεια να εκτελέσει την ενέργεια που είναι «δεμένη» με την συγκεκριμένη χειρονομία.

Η συσκευή χρησιμοποιεί το πρωτόκολλο HID (Human Interface Device) για διάδραση με άλλες συσκευές κάτι που την καθιστά συμβατή με τα πιο γνωστά λειτουργικά συστήματα Windows 8, iOS και Android 4.3 και πάνω.

Τέλος προκειμένου να μπορεί η συσκευή να λειτουργεί για μεγάλα χρονικά διαστήματα με περιορισμένα αποθέματα ενέργειας, μετά από ένα διάστημα ακινησίας απενεργοποιείται αυτόματα καταναλώνοντας μικρές ποσότητες ενέργειας. Η συσκευή μπορεί να ξυπνήσει και να επανασυνδεθεί απαιτώντας από τον χρήστη μόνο ένα απότομο κούνημα του δαχτύλου προκειμένου να τεθεί και πάλι σε πλήρη λειτουργία.

Στο παρόν κείμενο θα αναφερθούν αναλυτικά οι τεχνολογίες και ο εξοπλισμός που χρησιμοποιήθηκε στην ανάπτυξη της συσκευής. Στην συνέχεια θα περιγραφούν τα διάφορα υποσυστήματα που περιλαμβάνει η συσκευή και τέλος θα αναφερθούμε σε ελαττώματα, μειονεκτήματα και προβλήματα της, στην παρούσα ανάπτυξη της καθώς και πως θα μπορούσαμε να τα αντιμετωπίσουμε σε κάποια μελλοντική εργασία.

Κεφάλαιο 1 - Bluetooth Low Energy (BLE)

1.1 Bluetooth Core Specification

Η τεχνολογία Bluetooth 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 Wibree. 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 (τρέχουσα έκδοση του πυρήνα) 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 δια – λειτουργικότητα συσκευών βασισμένων σε προηγούμενες προδιαγραφές.

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.

Device	BR/EDR(classic Bluetooth) support	BLE(Bluetooth Low Energy) support
Pre-4.0 Bluetooth	Yes	No
4.x Single-Mode (Bluetooth Smart)	No	Yes
4.x Dual-Mode (Bluetooth Smart Ready)	Yes	Yes

Πίνακας 1: Είδη συσκευών που κυκλοφορούν στην αγορά.

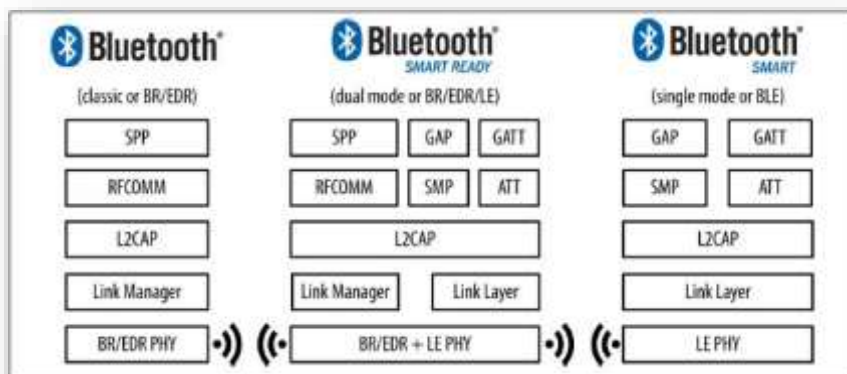
Όπως φαίνεται στον ΠΙΝΑΚΑ 1 οι προδιαγραφές για τον πυρήνα Bluetooth ορίζουν δύο ασύρματες τεχνολογίες.

- BR/EDR (κλασικό Bluetooth).
Το κλασικό πρότυπο όπως εξελίχθηκε από τον πυρήνα 1.0
- BLE (Bluetooth Low Energy)
Το χαμηλής κατανάλωσης πρότυπο που εισήχθη στις προδιαγραφές του Bluetooth πυρήνα από την έκδοση 4.0.

Οι συσκευές που υποστηρίζουν τις παραπάνω τεχνολογίες είναι δύο ειδών:

- Single Mode (BLE, Bluetooth Smart) συσκευή.
Είναι συσκευή που υιοθετεί μόνο την τεχνολογία BLE και μπορεί να συνδιαλεγεί είτε με single – mode είτε με dual – mode συσκευές αλλά όχι με συσκευές που υποστηρίζουν BR/EDR πρωτόκολλο μόνο.
- Dual Mode (BR/EDR/LE, Bluetooth Smart Ready) συσκευή.

Συσκευή που υιοθετεί και το κλασικό Bluetooth αλλά και Bluetooth Low Energy, και μπορεί να επικοινωνήσει με οποιαδήποτε Bluetooth συσκευή.



Εικόνα 1: Πιθανές ρυθμίσεις μεταξύ Bluetooth συσκευών διαφορετικών εκδόσεων.

1.2 Bluetooth και Bluetooth Low Energy

Όπως επώθηκε και στην προηγούμενη ενότητα η νεοεισαχθείσα 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 κλασσικού Bluetooth υιοθετήθηκαν από το Bluetooth Low Energy 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 κλασσικού Bluetooth, εφόσον έχουν πολλά κοινά χαρακτηριστικά, αυτό που κάνει το νέο πρότυπο να διαφέρει είναι ότι μπορεί να «κοιμάται». Η ικανότητα αυτή του BLE του επιτρέπει να βρίσκεται σε μια κατάσταση χαμηλής κατανάλωσης ενέργειας και στην ουσία να εκπέμπει μόνο όταν χρειάζεται (όταν αρχίζει μια σύνδεση ή όταν χρειάζεται να διατηρήσει κάποια σύνδεση). Η κατανάλωση ισχύος βρίσκεται σε χαμηλά επίπεδα εφόσον οι χρονικές περίοδοι σύνδεσης διαρκούν μόλις λίγα ms. Η μέγιστη κατανάλωση ισχύος είναι μόλις 15mA και η μέση κατανάλωση ισχύος είναι 1uA.

Κάτι που πρέπει να τονιστεί είναι ότι, το ποια τεχνολογία είναι πιο αποδοτική εξαρτάται αποκλειστικά από την εφαρμογή που θα χρησιμοποιηθεί. Το κλασικό Bluetooth μπορεί να επιτύχει ρυθμούς μετάδοσης που να ξεπερνούν τα 2Mb/s (ωφέλιμο φορτίο) σε αντίθεση με την νέα τεχνολογία όπου ο ρυθμός μετάδοσης είναι χαμηλότερος των 100kb/s. Από αυτό μπορούμε να αντιληφθούμε πως αν ο σκοπός μίας εφαρμογής είναι το streaming δεδομένων, με την

χρήση BLE, όποια ενδεχόμενη εξοικονόμηση ενέργειας θα χαθεί εφόσον η συσκευή θα πρέπει συνεχώς να εκπέμπει για να ανταποκριθεί στους ρυθμούς μετάδοσης της εφαρμογής.

	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

Πίνακας 2: Χαρακτηριστικά των τεχνολογιών Bluetooth και Bluetooth low Energy

Συνοψίζοντας, 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.

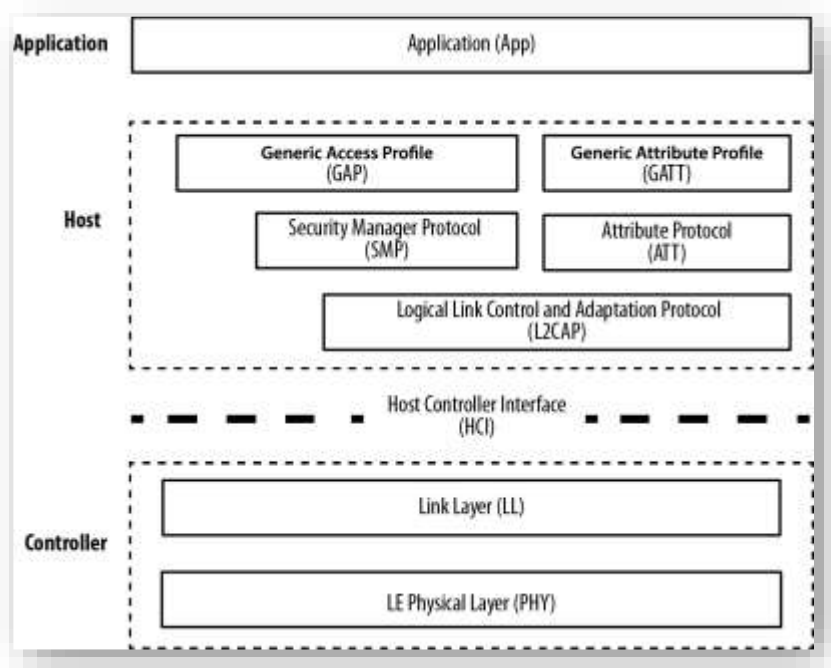
1.3 Κύρια δομικά στοιχεία του BLE.

Η αρχιτεκτονική του BLE αποτελείται από 3 βασικά δομικά στοιχεία:

- **Application:** 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.
- **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
- **Controller:** Το κατώτερο επίπεδο της στοίβας, συμπεριλαμβανομένου και του πομπού συχνοτήτων καθώς και των ακόλουθων επιπέδων.
 - Host Controller Interface (HCI), η πλευρά του Controller
 - Link Layer (LL)
 - Physical Layer (PHY)

Παρακάτω γίνεται μια περίληψη των λειτουργιών του καθενός και σε επόμενες παραγράφους θα αναλυθούν τα πρωτόκολλα GAP και GATT όπως και η έννοιες των profile

εφόσον αποτελούν τα πιο σημαντικά μέρη της τεχνολογίας που έρχονται σε επαφή με την ανάπτυξη της εκάστοτε εφαρμογής.



Εικόνα 22: Στοιβα πρωτοκόλλου BLE.

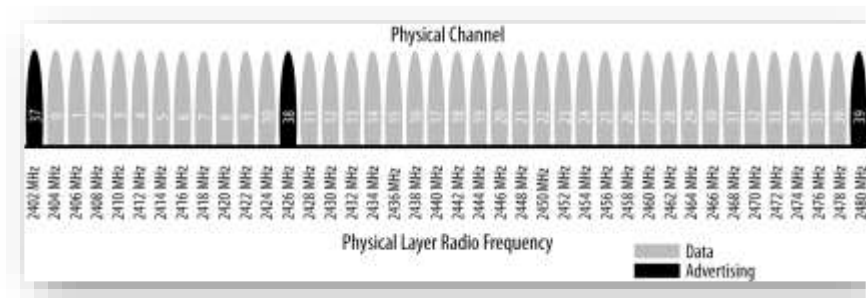
1.3.1 Physical Layer

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 όπως φαίνεται στην **ΕΙΚΟΝΑ 3**. 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 βεβαια δεν επιτυγχάνεται

Commented [M1]: Με κεφαλαίο το Εικόνα. Σε όλα τα σημεία που κάνεις reference σε εικόνα.

ποτέ εφόσον κάθε επίπεδο πρωτοκόλλου προσθέτει επι πλέον κεφαλίδες άρα το ωφέλιμο φορτίο γίνεται όλο και πιο μικρό).



Εικόνα 33: Κανάλια συχνότητας που χρησιμοποιούνται από το BLE πρωτόκολλο.

1.3.2 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 επι πλέον 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 απόθεμα 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 σιαφημιστικά 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.

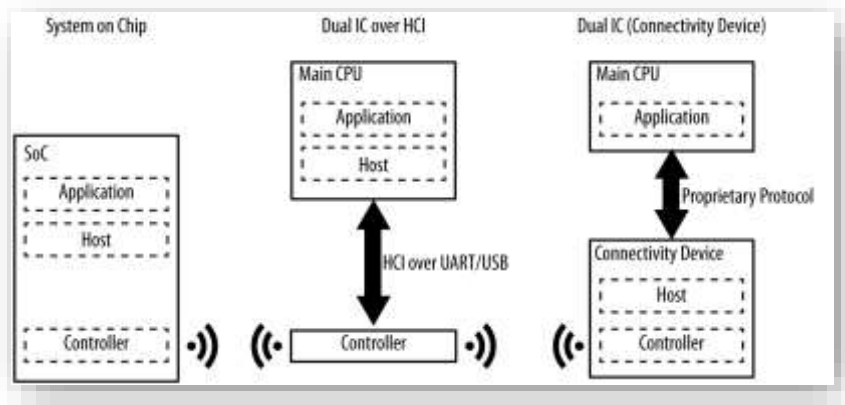
Οι παραπάνω ρόλοι μπορούν να ομαδοποιηθούν στα ακόλουθα ζεύγη:

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

1.3.3 Host Controller Interface (HCI)

Όπως έχει ήδη περιγραφεί, 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 προτυποποιημένων 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 συνήθισμένη αρχιτεκτονική μίας τέτοιας συσκευής υλοποιεί τον Host στην κεντρική μονάδα επεξεργασίας ενώ ο controller βρίσκεται σε ξεχωριστό υλικό.



Εικόνα 44: Ρυθμίσεις υλικού για BLE συσκευές.

Η παραπάνω αρχιτεκτονική δομή με το Application και τον Host να εκτελούνται στο ίδιο κομμάτι ενώ ο controller να εκτελείται σε ξεχωριστό κομμάτι λέγεται *Dual IC over HCI*.

Υπάρχουν άλλες δύο επί πλέον αρχιτεκτονικές, η *Dual IC with connectivity device* όπου σε ένα κομμάτι hardware υλοποιείται το Application και σε άλλο κομμάτι υλοποιείται ο Host και ο Controller. Το πρωτόκολλο που ακολουθείται για την διεπαφή μεταξύ Host και του Application εξαρτάται από τον εκάστοτε κατασκευαστή του ολοκληρωμένου. Τέλος η πιο απλή αρχιτεκτονική που μπορεί να ακολουθηθεί είναι αυτή του SoC όπου όλα τα κομμάτια του προτύπου εκτελούνται στο ίδιο chip.

1.3.4 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).

1.3.5 Attribute Protocol (ATT)

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

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

- ένα μοναδικό 16-bit αριθμό που αποτελεί το attribute handle, χρησιμοποιείται να 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 γραφεί ή να διαβαστεί από τον client.
- την τιμή που έχει το γνώρισμα.

Όταν ένας Client θέλει να διαβάσει ή να γράψει κάποιο από τα γνωρίσματα που υπάρχουν σε έναν server γνωστοποιεί ένα αίτημα για διάβασμα ή για γράψιμο με την χρήση του attribute handle. Ο server θα ανταποκριθεί με την τιμή που έχει το γνώρισμα που αντιστοιχεί στο attribute handle ή με κάποια επιβεβαίωση ότι η αίτηση έχει ληφθεί.

Οι κατηγορίες των εντολών που διαθέτει το πρωτόκολλο είναι:

- Εντολές για χειρισμό σφαλμάτων (Error Handling) – Χρησιμοποιούνται από τον server για να γνωστοποιήσουν τυχόν σφάλματα.
- Ρυθμίσεις server (Server Configuration) – Χρησιμοποιείται για την ρύθμιση του ίδιου του πρωτοκόλλου.
- Εύρεση πληροφοριών (Find Information) – Χρησιμοποιούνται από τον Client για πληροφορίες και για την αναγνώριση των γνωρισμάτων του Server.
- Λειτουργίες διαβάσματος (Read Operations) – Χρησιμοποιούνται από τον client για την λήψη της τιμής των γνωρισμάτων.
- Λειτουργίες εγγραφής (Write operations) – Χρησιμοποιούνται από τον client για θέσουν την τιμή ενός γνωρίσματος.
- Εγγραφές ουράς (Queued writes) – Χρησιμοποιούνται από τον client για να γίνουν εγγραφές στις τιμές των γνωρισμάτων με τιμές που είναι μεγαλύτερες από ότι χωράει ένα κανονικό πακέτο μεταφοράς.
- Λειτουργίες που εκκινούνται από τον server (Server Initiated) – Χρησιμοποιούνται από τον server προκειμένου να αποσταλούν ασύγχρονα τιμές γνωρισμάτων στον client.

1.3.6 Security Manager (SM)

Το επίπεδο security manager λειτουργεί σαν πρωτόκολλο και αποτελείται από μια σειρά από αλγορίθμους ασφαλείας που επιτρέπουν στην τεχνολογία να ανταλλάσει κλειδιά ασφαλείας

προκειμένου να γίνεται ασφαλής μεταφορά δεδομένων πάνω από μια κωδικοποιημένη σύνδεση.

Το επίπεδο security manager ορίζει δύο ρόλους:

- Τον Initiator – πάντα έχει τον ρόλο του master όσον αφορά το Link Layer επίπεδο.
- 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.

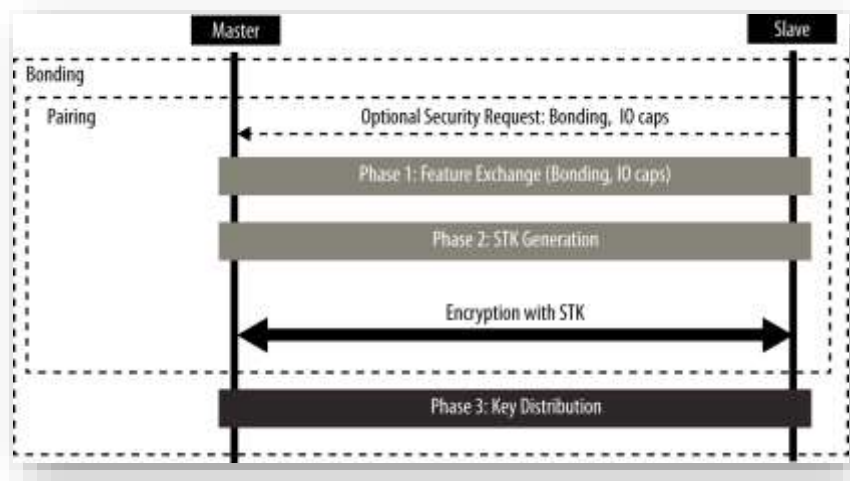
Όπως αντιλαμβανόμαστε η διαδικασία του pairing δημιουργεί μια ασφαλή σύνδεση η οποία διαρκεί όσο οι συσκευές είναι συνδεδεμένες, ενώ το bonding δημιουργεί μόνιμο δεσμό στις συσκευές έως ότου κάποια πλευρά αποφασίσει και διαγράψει αυτόν τον μόνιμο δεσμό.

Αυτό που αξίζει να σημειώσουμε είναι ότι 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.

Η μέθοδος Just Works και η μέθοδος Out Of Band προσφέρουν προστασία από MITM επιθέσεις.



Εικόνα 55: Διαδικασίες pairing και bonding.

1.3.7 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. Περισσότερες λεπτομέρειες σχετικά με το πρωτόκολλο GATT αναφέρονται στο κεφάλαιο 1.5.

1.3.8 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. Περισσότερες λεπτομέρειες σχετικά με το κεφάλαιο GAP αναφέρονται στο κεφάλαιο 1.4.

1.4 Generic Access Profile Protocol (GAP)

Όπως αναφέραμε και σε προηγούμενη παράγραφο το πρωτόκολλο GAP θέτει κάποιους κανόνες στους οποίους θα πρέπει οι συσκευές που ενσωματώνουν την BLE τεχνολογία να ανταποκρίνονται προκειμένου να μπορούν να ανιχνεύουν, να εκπέμπουν δεδομένα, να θέτουν ασφαλείς συνδέσεις και να εκτελούν άλλες λειτουργίες όπως ορίζει το πρότυπο του BLE.

Το πώς ορίζει το πρότυπο την λειτουργία μιας BLE συσκευής σε διαφορετικές καταστάσεις λειτουργίας συνοψίζεται στους ακόλουθους τρόπους αλληλεπίδρασης.

1.4.1 Ρόλοι (Roles)

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.

1.4.2 Λειτουργίες και διαδικασίες (Modes and Procedures)

Οι λειτουργίες είναι καταστάσεις που μπορεί να εισέλθει μια συσκευή για συγκεκριμένο χρονικό διάστημα προκειμένου να επιτευχθεί κάποιος σκοπός ή ακόμα καλύτερά, προκειμένου να επιτραπεί σε κάποιο peer να εκτελέσει κάποια συγκεκριμένη διαδικασία. Οι διάφορες λειτουργίες είναι άρρηκτα συνδεδεμένες με τις αντίστοιχες διαδικασίες. Παρακάτω αναφέρονται οι διάφορες λειτουργίες και οι αντίστοιχες διαδικασίες:

- **Λειτουργία Broadcast και διαδικασία Observation:** θέτει την δομή μέσω της οποίας μια συσκευή μπορεί να στείλει δεδομένα (Broadcaster) σε οποιαδήποτε άλλη συσκευή ή περισσότερες που βρίσκονται εντός εμβέλειας.
- **Λειτουργία και διαδικασίες ανίχνευσης:** Οι ακόλουθες λειτουργίες ανίχνευσης επιτρέπουν την ευελιξία που ενδεχομένως απαιτείται από κάποια εφαρμογή.
 - **Non-discoverable mode:** η συσκευή δεν μπορεί να ανιχνευθεί από άλλες συσκευές.
 - **Limited Discoverable mode:** η συσκευή είναι ανιχνεύσιμη για συγκεκριμένο χρονικό διάστημα.

- General Discoverable mode: η συσκευή είναι ανιχνεύσιμη έως ότου αυτό κριθεί επαρκές.
- Limited Discovery procedure: επιτρέπει στην συσκευή να είναι ανιχνεύσιμη για περιορισμένο χρόνο. Μια συσκευή με τον ρόλο central αρχίζει την διαδικασία ενεργής εύρεσης συσκευών, ότι διαφημιστικό πακέτο βρεθεί φιλτράρεται και αν η peer συσκευή διαφημίζεται με limited discoverable mode ενεργοποιημένο τότε αρχίζουν οι εκάστοτε διαδικασίες της εφαρμογής.
- General Discovery procedure: η central συσκευή αντιδρά σε όποια διαφημιστικό πακέτο εντοπιστεί είτε αυτό είναι από peer συσκευή που λειτουργεί με Limited Discoverable mode είτε με General Discoverable mode.
- **Λειτουργίες και διαδικασίες αρχικοποίησης σύνδεσης:** Για οποιοδήποτε central peer προκειμένου να αρχίσει την διαδικασία σύνδεσης θα πρέπει, το client peer με το οποίο θα θελήσει να συνδεθεί, να βρίσκεται σε connectable κατάσταση.
 - Non-Connectable mode: η συσκευή δεν είναι σε κατάσταση που μπορεί να συνδεθεί.
 - Directed connectable mode: η συσκευή μπορεί να συνδεθεί μόνο με το κεντρικό peer του οποίου η διεύθυνση βρίσκεται στο διαφημιστικό πακέτο που έχει σταλεί.
 - Undirected connectable mode: το περιφερειακό διαφημίζεται σε οποιαδήποτε συσκευή προκειμένου να αρχίσει η διαδικασία της σύνδεσης.
 - Auto connection establishment διαδικασία: το central κρατά μια λίστα με τις γνωστές συσκευές (white list) και στην συνέχεια συνδέεται με το πρώτο περιφερειακό που ανιχνεύει και βρίσκεται σε αυτήν την λίστα.
 - General connection establishment διαδικασία: το central ανιχνεύει και εμφανίζει όποιο περιφερειακό βρίσκεται στην εμβέλεια του. Αν το κεντρικό θα συνδεθεί με το εκάστοτε περιφερειακό εξαρτάται από την εφαρμογή. Για την σύνδεση χρησιμοποιείται η Direct connection establishment λειτουργία.
 - Selective connection establishment διαδικασία: η διαδικασία είναι ίδια με την παραπάνω με την διαφορά ότι εδώ χρησιμοποιείται η white list και ο χρήστης μπορεί να επιλέξει από όσες συσκευές διαφημίζονται μόνο αυτές που είναι ήδη γνωστές.
 - Direct connection establishment διαδικασία: με αυτήν την διαδικασία ένα κεντρικό συνδέεται μόνο με ένα περιφερειακό.

Αξίζει να σημειωθεί ότι υπάρχουν δύο τρόποι αρχικοποίησης σύνδεσης. Στην πρώτη μέθοδο, πρώτα η συσκευή γίνεται γνωστή μέσω της διαδικασίας του scanning και έπειτα ακολουθεί σύνδεση στην συσκευή που έγινε γνωστή μέσω του scanning. Στην δεύτερη μέθοδο η συσκευή προσπαθεί να συνδεθεί απευθείας με μια συσκευή που είναι γνωστή ανεξαρτήτως αν αυτή είναι κοντά.

- **Επιπλέον λειτουργίες GAP:** Το πρωτόκολλο επίσης ορίζει κάποιες επί πλέον λειτουργίες που απευθύνονται σε καταστάσεις όπου ήδη υπάρχει σύνδεση:
 - Name discovery διαδικασία: σε κάποιες περιπτώσεις το όνομα κάποιας συσκευής δεν εμφανίζεται στα διαφημιστικά πακέτα, για τον λόγο αυτό υπάρχει η αντίστοιχη διαδικασία για την ανάκτηση του.

- Connection parameter update διαδικασία: κάθε σύνδεση γίνεται βάση κάποιων παραμέτρων σύνδεσης, υπεύθυνος γι' αυτές τις παραμέτρους είναι πάντα ο master μιας σύνδεσης. Παρόλα' αυτά ένα περιφερειακό μπορεί να αιτηθεί για αλλαγή των παραμέτρων σύνδεσης και είναι στην ευχέρεια του κεντρικού αν θα ανταποκριθεί στο αίτημα ή όχι. Ακόμα όμως και αν το κεντρικό αλλάξει τις παραμέτρους μπορεί οι νέες παράμετροι να μην είναι αυτοί που έχει αιτηθεί το περιφερειακό αλλά αυτές που θεωρεί το κεντρικό σαν βέλτιστες για την σύνδεση.
- Terminate connection διαδικασία: με αυτήν την διαδικασία οι συσκευές τερματίζουν την σύνδεσή τους.

Mode	Applicable Role(s)	Applicable Peer Procedure(s)
Broadcast	Broadcaster	Observation
Non-discoverable	Peripheral	N/A
Limited discoverable	Peripheral	Limited and General discovery
General discoverable	Peripheral	General discovery
Non-connectable	Peripheral, Broadcaster, Observer	N/A
Any connectable	Peripheral	Any connection establishment

Πίνακας 33: Λειτουργίες και οι αντίστοιχες διαδικασίες του πρωτοκόλλου GAP.

Procedure	Applicable Role(s)	Applicable Peer Mode(s)
Observation	Observer	Broadcast
Limited discovery	Central	Limited discoverable
General discovery	Central	Limited and General discoverable
Name discovery	Peripheral, central	N/A
Any connection establishment	Central	Any connectable
Connection parameter update	Peripheral, central	N/A
Terminate connection	Peripheral, central	N/A

Πίνακας 34: Διαδικασίες και οι αντίστοιχες λειτουργίες που απαιτούνται.

1.4.3 Ασφάλεια

Το GAP βασίζεται πάνω στην λειτουργία του Security Manager προκειμένου να ορίσει κάποιες λειτουργίες και διαδικασίες ασφάλειας. Ο ρόλος που έχει η συσκευή στην σύνδεση είναι άμεσα συνδεδεμένος με τις λειτουργίες ασφάλειας που πρέπει να εκτελέσει. Παρακάτω αναφέρονται οι διάφορες πτυχές ασφαλείας που παρέχει το πρωτόκολλο.

- **Τύποι διευθύνσεων:** το BLE πρότυπο ορίζει δύο τύπους διευθύνσεων. Την public διεύθυνση, που είναι η διεύθυνση που έχει η συσκευή από το εργοστάσιο παραγωγής και η Random διεύθυνση που είτε μπορεί να οριστεί στην συσκευή από τον προγραμματιστή ή να δημιουργηθεί δυναμικά κατά την διάρκεια λειτουργίας της συσκευής. Το GAP υποδιαιρεί ακόμα περισσότερο τον τύπο της Random διεύθυνσης στις αντίστοιχες κατηγορίες.
 - Static address: χρησιμοποιείται σαν υποκατάστατο της public όταν ο κατασκευαστής δεν επιθυμεί να την χρησιμοποιήσει. Είναι απλά ένας τυχαίος αριθμός που παράγεται κατά την εκκίνηση του συστήματος.

- Non – Resolvable private address: είναι ένας τυχαίος αριθμός και χρησιμοποιείται σαν μια προσωρινή διεύθυνση για περιορισμένο χρονικό διάστημα.
- Resolvable private address: παράγεται από το ένα identity resolving key (IRK) και έναν τυχαίο αριθμό και μπορεί να μεταβάλλεται συχνά. Χρησιμοποιείται για να αποφευχθεί η ανίχνευση και αναγνώριση της συσκευής από ανεπιθύμητους ανιχνευτές. Μόνο οι συσκευές που έχουν το IRK που έχει διανεμηθεί κατά την διάρκεια της σύνδεσης μπορούν να αναγνωρίσουν την συσκευή.
- **Λειτουργίες ασφαλείας:** κάθε σύνδεση λειτουργεί κάτω από κάποια κατάσταση ασφαλείας. Η κατάσταση ασφαλείας ορίζει και το επίπεδο ασφαλείας της σύνδεσης, το οποίο κάποια στιγμή μπορεί να αλλάξει ανάλογα με τις απαιτήσεις που έχουν τα peer που βρίσκονται σε σύνδεση, οδηγώντας στην ενεργοποίηση διαδικασιών για την αύξηση του επιπέδου ασφαλείας. Το GAP ορίζει δύο καταστάσεις ασφαλείας και συγκεκριμένα επίπεδα ασφαλείας ανά κατάσταση.
 - Security mode 1: Αυτή η κατάσταση ενδυναμώνει την ασφάλεια με την χρήση κρυπτογράφησης και περιέχει τρία επίπεδα.
 - Level 1: καμία ασφάλεια, η σύνδεση δεν είναι κρυπτογραφημένη.
 - Level 2: μη πιστοποιημένη κρυπτογράφηση.
 - Level 3: πιστοποιημένη κρυπτογράφηση.
 - Security mode 2: Αυτή η κατάσταση ενδυναμώνει την ασφάλεια με την χρήση του data signing και περιλαμβάνει δύο επίπεδα.
 - Level 1: μη πιστοποιημένο data signing.
 - Level 2: πιστοποιημένο data signing.

Με την έννοια της πιστοποίησης εννοούμε ότι η κρυπτογράφηση ή το data signing έγιναν με την χρήση κλειδιών που για την παραγωγή τους χρησιμοποιήθηκαν οι μέθοδοι Passkey Display ή Out of Band. Όταν η κρυπτογράφηση ή το data signing είναι μη πιστοποιημένο το κλειδί που χρησιμοποιήθηκε για την διαδικασία παράχθηκε με την μέθοδο του Just works.

- Πέρα από τις καταστάσεις και διαδικασίες σύνδεσης και ανταλλαγής δεδομένων το GAP ορίζει και κάποιες επιπλέον ζεύξεις που σχετίζονται με την ασφάλεια. Παρακάτω αναφέρονται οι αντίστοιχες καταστάσεις και λειτουργίες.
 - Non bondable κατάσταση: μια συσκευή στην παρούσα κατάσταση δεν επιτρέπει την εκτέλεση της διαδικασίας του bonding, παρόλ αυτά επιτρέπει να γίνει η διαδικασία του pairing. Κατά την διάρκεια της διαδικασίας μια συσκευή δεν μπορεί να διανέμει, να δεχθεί και να αποθηκεύσει κλειδιά, περιορίζοντας όλες τις αναβαθμίσεις ασφαλείας κατά την διάρκεια μίας σύνδεσης.
 - Bondable κατάσταση: μια συσκευή μπορεί να εκτελέσει την διαδικασία του bonding και να αποθηκεύσει κλειδιά.
 - Bonding διαδικασία: η συσκευή central μπορεί να εκκινήσει την διαδικασία bonding.
 - Διαδικασία πιστοποίησης: κατά την διάρκεια αυτής της διαδικασίας η συσκευή μπορεί να προβεί σε ανώτερα επίπεδα και καταστάσεις ασφαλείας.
 - Διαδικασία εξουσιοδότησης: δίνει την δυνατότητα στον χρήστη ή στην εφαρμογή να απορρίψει μια συνναλλαγή. Αυτό μπορεί να εκτελεστεί από

κάποιο έλεγχο της εφαρμογής αν η εφαρμογή δεν έχει κάποιο user interface, ή αν έχει μπορεί να ζητηθεί άδεια κατευθείαν από τον χρήστη.

- ο Διαδικασία κρυπτογράφησης: εκκίνηση της διαδικασίας κρυπτογράφησης με χρήση των εργαλείων που προσφέρει το Link επίπεδο για την κρυπτογράφηση της εκάστοτε σύνδεσης.

1.4.4 Επιπλέον λειτουργίες του πρωτοκόλλου GAP

- **Τύποι διαφημιστικών πακέτων:** το πρότυπο ορίζει ότι στα πακέτα διαφήμισης μπορεί ο χρήστης να ενθυλακώσει δεδομένα. Το διαφημιστικό πακέτο αποτελείται από μια σειρά από τύπους δεδομένων, κάθε ένας από τους οποίους αποτελείται από το μέγεθος, το AD Type (advertising data type) και τα δεδομένα. Κάθε δομή δεδομένων περιέχει μια ξεχωριστή πληροφορία από δεδομένα χρήστη. Δεν είναι απαραίτητα όλα τα AD Types σε όλες τις περιπτώσεις και το μέγεθος του πακέτου διαφήμισης στα 31 byte περιορίζει το πόσα AD Types μπορούν να χωρέσουν σε ένα πακέτο. Οπότε το ποιο AD Type θα ενθυλακωθεί στο διαφημιστικό πακέτο θα πρέπει να ανταποκρίνεται στην λειτουργία της εφαρμογής. Στον ΠΙΝΑΚΑ 5 αναφέρονται τα AD Types καθώς και σε τι χρησιμεύει το καθένα.

Όνομα AD Type	Μέγεθος σε Bytes	Περιγραφή
Flags	1 (επεκτάσιμο)	Χρησιμοποιείται για τεθεί η συσκευή σε limited ή general discoverable κατάσταση.
Local name	ποικίλει	Όνομα της συσκευής σε UTF-8 μπορεί να είναι ολόκληρο το όνομα ή μέρος του ονόματος
Appearance	2	Χρησιμοποιείται για την περιγραφή της συσκευής που στέλνει το διαφημιστικό πακέτο.
TX level power	1	Η ισχύς που χρησιμοποιείται σε dBm για την μετάδοση του διαφημιστικού πακέτου, χρησιμοποιείται για να υπολογιστεί η απώλεια του σήματος στον δέκτη.
Service UUID	ποικίλει	Λίστα στην οποία αναγράφονται οι GATT υπηρεσίες που υποστηρίζει η συσκευή που διαφημίζεται.
Slave connection interval range	4	Οι βέλτιστες παράμετροι σύνδεσης που θέτει το περιφερειακό.
Service solicitation	ποικίλει	Λίστα με GATT υπηρεσίες που υποστηρίζει η συσκευή που στέλνει το πακέτο (από την πλευρά του client).
Service data	ποικίλει	Ένα UUID που αναπαριστά μια GATT υπηρεσία και τα συσχετιζόμενα δεδομένα.
Manufacturer Specific data	ποικίλει	Ελεύθερο πεδίο που χρησιμοποιείται ανάλογα με την εφαρμογή που υποστηρίζει η συσκευή.

Πίνακας 5: Πεδία διαφημιστικών πακέτων.

- **GAP service:** κάθε συσκευή πρέπει να περιλαμβάνει μια υπηρεσία (service) του GAP (για τις υπηρεσίες GATT και πως αυτές λειτουργούν αναφερόμαστε σε επόμενο κεφάλαιο). Στην υπηρεσία αυτή έχουν πρόσβαση όλες οι συνδεδεμένες συσκευές χωρίς να υπάρχει κάποιο περιορισμός ασφαλείας και η συγκεκριμένη υπηρεσία περιλαμβάνει τα ακόλουθα χαρακτηριστικά.

- Device Name χαρακτηριστικό: το συγκεκριμένο χαρακτηριστικό περιλαμβάνει το όνομα της συσκευής σε μορφή UTF-8.
- Appearance χαρακτηριστικό: η τιμή του συγκεκριμένου χαρακτηριστικού ορίζει αν η συσκευή ανήκει σε κάποια κατηγορία γενικών συσκευών (τηλέφωνο, ηλ. Υπολογιστής, ρολόι, αισθητήριο κ.α.).
- Peripheral Preferred Connection Parameters (PPCP) χαρακτηριστικό: όταν η κεντρική συσκευή έχει ήδη δημιουργήσει μια σύνδεση με κάποιο περιφερειακό μπορεί να διαβάσει την τιμή του χαρακτηριστικού και να ανανεώσει τις παραμέτρους της σύνδεσης σύμφωνα με την ένδειξη που του δίνει το περιφερειακό.

1.5 Generic Attribute Profile Protocol (GATT)

Το Generic Attribute Profile (GATT) ορίζει πως θα γίνεται η ανταλλαγή δεδομένων πάνω από μια BLE σύνδεση. Ακόμη είναι υπεύθυνο για την δομή των προφίλ που βασίζονται σε αυτό. Όλα τα προφίλ που βασίζονται στο GATT πρωτόκολλο πρέπει να συμμορφώνονται σε αυτό προκειμένου να λειτουργούν σωστά. Το GATT χρησιμοποιεί το ATT πρωτόκολλο προκειμένου να μεταφέρει δεδομένα μεταξύ συσκευών. Τα δεδομένα αυτά είναι ιεραρχικά δομημένα σε τμήματα που λέγονται υπηρεσίες (services), στην ουσία, οι υπηρεσίες, συγκεντρώνουν τα διάφορα χαρακτηριστικά (δεδομένα του χρήστη με επιπλέον πληροφορίες για την φύση των δεδομένων).

1.5.1 Ρόλοι (Roles)

Όπως και τα άλλα πρωτόκολλα έτσι και το GATT ορίζει ρόλους που μπορούν οι συσκευές να υιοθετήσουν:

- **Client:** αντιστοιχεί στο ρόλο του client που ορίζεται στο ATT πρωτόκολλο. Στέλνει αιτήματα στην συσκευή που υποστηρίζει τον ρόλο του server και λαμβάνει ανταποκρίσεις από αυτόν. Αρχικά ο client δεν γνωρίζει τίποτα για τις υπηρεσίες που υποστηρίζει ο server έτσι πρώτα πρέπει να «ερευνήσει» για τα διάφορα γνωρίσματα (attributes) που έχει ο server χρησιμοποιώντας την διαδικασία του service discovery. Αφού ολοκληρώσει την διαδικασία μπορεί να αρχίσει να διαβάζει ή να γράφει στα διάφορα γνωρίσματα ή και ακόμα να παίρνει ειδοποιήσεις για αλλαγές που πραγματοποιούνται σε αυτά από διαδικασίες που ενεργοποιούνται από τον server.
- **Server:** αντιστοιχεί στο ρόλο του server που ορίζεται στο ATT πρωτόκολλο. Λαμβάνει εντολές από τον client και στην συνέχεια ανταποκρίνεται. Επίσης μπορεί να εκκινήσει διαδικασίες χωρίς εντολές από τον client για παράδειγμα να στείλει την αλλαγή μιας τιμής ενός αισθητηρίου. Η συσκευή με αυτόν τον ρόλο στην ουσία είναι αυτή που αποθηκεύει και κάνει διαθέσιμα τα δεδομένα στον client υπό την μορφή γνωρισμάτων.

1.5.2 Γνωρίσματα (Attributes)

Τα γνωρίσματα (Attributes) είναι η μικρότερη μονάδα πληροφορίας που ορίζεται από το πρωτόκολλο ATT. Είναι μικρά διευθυνσιοδοτούμενα κομμάτια πληροφορίας που περιέχουν πληροφορία για το ίδιο το γνώρισμα και την ίδια την πληροφορία. Παρακάτω αναφέρονται τα πεδία που περιλαμβάνει ένα γνώρισμα προκειμένου να αποθηκεύει δεδομένα.

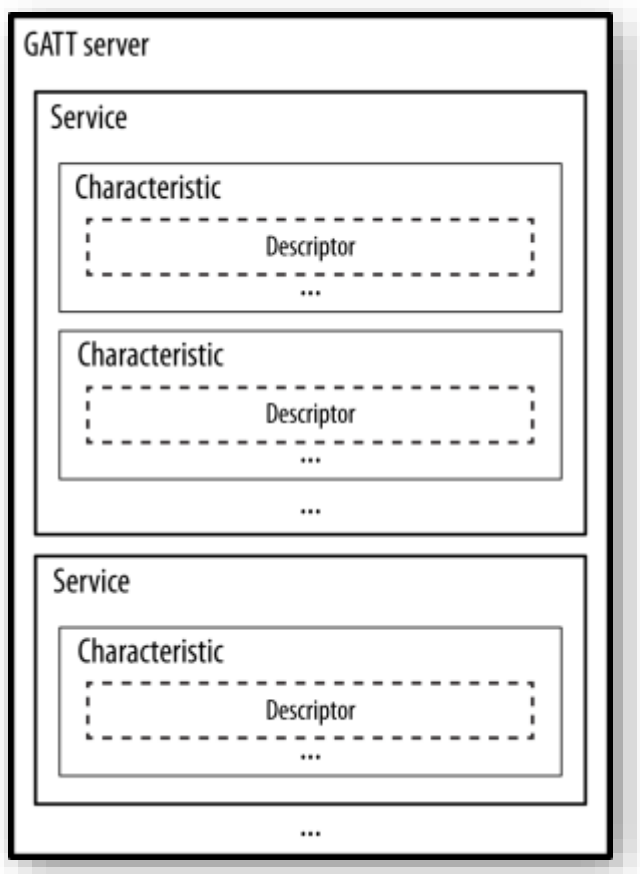
- **Handle:** το handle είναι ένας 16-bit αριθμός, διαφορετικός για κάθε γνώρισμα σε έναν server. Είναι το στοιχείο που κάνει το γνώρισμα διευθυνσιοδοτούμενο και μένει αμετάβλητο κατά την διάρκεια συναλλαγών και κατά την διάρκεια μιας ενεργής σύνδεσης. Σε έναν GATT server οι αυξανόμενες τιμές των handles ορίζουν μια σειρά από γνωρίσματα στα οποία μπορεί να έχει πρόσβαση ο χρήστης.
- **Type (τύπος):** ο τύπος ενός γνωρίσματος είναι ένας 16-, 32-, ή 128-bit UUID αριθμός που ορίζει τον τύπο του δεδομένου που βρίσκεται σε ένα γνώρισμα. Τα διαφορετικά UUID που μπορεί να πάρει το πεδίο type μπορεί να δηλώνουν είτε την έναρξη ενός νέου χαρακτηριστικού, είτε την έναρξη μιας υπηρεσίας, είτε το είδος των δεδομένων που περιέχει το συγκεκριμένο γνώρισμα ανάλογα την εφαρμογή που θα σηκώσει η συσκευή.
- **Permissions (αδειοδότηση):** είναι πεδία που ορίζουν τα δικαιώματα που μπορεί να έχει κάποιος πάνω στο συγκεκριμένο γνώρισμα και συμπεριλαμβάνουν τις ακόλουθες αδειοδοτήσεις.
 - Access Permissions (άδειες πρόσβασης): ορίζουν αν κάποια client συσκευή μπορεί να γράψει ή να διαβάσει την τιμή ενός γνωρίσματος σε έναν server. Κάθε γνώρισμα μπορεί να έχει μία από τις ακόλουθες άδειες πρόσβασης.
 - None (καμία): το γνώρισμα δεν μπορεί να γραφεί ή να αναγνωστεί.
 - Readable (αναγνώσιμη): το γνώρισμα μπορεί μόνο να αναγνωστεί.
 - Writable (εγγράψιμη): το γνώρισμα μπορεί μόνο να εγγραφεί.
 - Readable and Writable (αναγνώσιμη και εγγράψιμη): το γνώρισμα μπορεί να αναγνωστεί και να εγγραφεί.
 - Encryption (κρυπτογράφηση): ορίζει αν το γνώρισμα χρειάζεται κάποιο επίπεδο κρυπτογράφησης προκειμένου να έχει πρόσβαση ο client.
 - No encryption required (χωρίς κρυπτογράφηση): το γνώρισμα είναι προσπελάσιμο χωρίς κάποια κρυπτογράφηση.
 - Unauthenticated encryption required (μη πιστοποιημένη κρυπτογράφηση): η σύνδεση πρέπει να είναι κρυπτογραφημένη αλλά τα κλειδιά που χρησιμοποιήθηκαν δεν χρειάζεται να είναι πιστοποιημένα για να έχει ο client πρόσβαση.
 - Authenticated encryption required (πιστοποιημένη κρυπτογράφηση): η σύνδεση πρέπει να είναι κρυπτογραφημένη με πιστοποιημένα κλειδιά για να υπάρξει πρόσβαση από τον client.
 - Authorization: ορίζει αν χρειάζεται άδεια από τον χρήστη προκειμένου να υπάρξει πρόσβαση στο γνώρισμα.
 - No authorization required (χωρίς άδεια): δεν χρειάζεται αδειοδότηση χρήστη για πρόσβαση.
 - Authorization required (με χρήση άδειας): χρειάζεται η άδεια του χρήστη για πρόσβαση.
- **Value (τιμή):** το συγκεκριμένο πεδίο αποθηκεύει τα δεδομένα ενός γνωρίσματος. Ανάλογα με τον τύπο ενός γνωρίσματος το συγκεκριμένο πεδίο μπορεί να αποθηκεύει πληροφορία που αφορά τα ίδια τα γνωρίσματα ή και δεδομένα του χρήστη ή μετρήσεις από κάποιο αισθητήριο κ.α. Αυτό είναι το πεδίο στο οποίο μπορεί να έχει πρόσβαση ο

χρήστης για εγγραφή ή ανάγνωση, ανάλογα και με την αδειοδότηση που έχει. Όλα τα προηγούμενα πεδία αποτελούν την δομή ενός γνωρίσματος και δεν μπορούν να αλλαχθούν ή να προσπελαθούν απευθείας από τον χρήστη.

1.5.3 Ιεραρχία Δεδομένων

Τα γνωρίσματα ορίζονται από το πρωτόκολλο ATT όπως αναφέρθηκε στην προηγούμενη παράγραφο. Το πρωτόκολλο GATT είναι αυτό που ορίζει την ιεραρχία και την δομή στα διάφορα γνωρίσματα προκειμένου να μπορεί να γίνεται η ανταλλαγή δεδομένων μεταξύ server και client με την χρήση προφίλ που βασίζονται στο πρωτόκολλο GATT.

Τα διάφορα γνωρίσματα σε έναν GATT server οργανώνονται σε υπηρεσίες (services), κάθε μια υπηρεσία περιέχει μηδέν ή περισσότερα χαρακτηριστικά (characteristics). Τα χαρακτηριστικά με την σειρά τους μπορούν να περιλαμβάνουν μηδέν ή περισσότερους περιγραφείς (descriptors). Αυτή είναι η ιεραρχία που ακολουθήται σε όλες τις BLE συσκευές, κάθε γνώρισμα θα πρέπει να βρίσκεται εντός μίας από τις παραπάνω κατηγορίες. Όλα τα δεδομένα σε μία GATT ιεραρχία αποτελούνται από γνωρίσματα τα οποία είτε μπορούν να ορίζουν τα δεδομένα είτε να τα δηλώνουν.



Εικόνα 6: Ιεραρχία δεδομένων GATT.

- **Services (υπηρεσίες):** οι υπηρεσίες του πρωτοκόλλου GATT ομαδοποιούν, συσχετιζόμενα μεταξύ τους, γνωρίσματα σε ένα κοινό κομμάτι της server συσκευής. Με τον όρο υπηρεσία αναφερόμαστε σε όλα τα γνωρίσματα που την αποτελούν. Η έναρξη μιας υπηρεσίας ξεκινά με ένα γνώρισμα που φέρει τα πεδία τύπου και τιμής που αντιστοιχούν στην δήλωση της υπηρεσίας όπως φαίνεται στον ΠΙΝΑΚΑ 6.

Handle	Type	Permissions	Value	Value length
0XNNNN	UUID primary service	Read Only	Service UUID	2,4 or 16 bytes

Πίνακας 6: Δήλωση γνωρίσματος υπηρεσίας.

Στον ΠΙΝΑΚΑ 6 η τιμή στο πεδίο τύπος (type) αναφέρεται στο UUID που έχει οριστεί από το Bluetooth SIG και χρησιμοποιείται για να ορίσει αποκλειστικά την έναρξη μια νέας υπηρεσίας πρωτεύουσα η δευτερεύουσα ανάλογα. Η διαφορά μεταξύ πρωτεύουσας η δευτερεύουσας υπηρεσίας είναι ότι, μια δευτερεύουσα υπηρεσία περιλαμβάνεται σε μία πρωτεύουσα και δεν μπορεί να χρησιμοποιηθεί μόνη της από τον client παραμόνο μέσω της πρωτεύουσας υπηρεσίας. Το πεδίο Value (τιμή) της υπηρεσίας περιέχει και αυτό ένα UUID που δηλώνει την πραγματική υπηρεσία που την έναρξή της δηλώνει το γνώρισμα. Μέσα στην δήλωση της αρχικής υπηρεσίας μπορεί επίσης να γίνουν αναφορες και να συμπεριληφθούν επιπλέον υπηρεσίες χωρίς να χρειαστεί να επαναληφουν τα γνωρίσματα που τις συνθέτουν. Αυτό γίνεται με την χρήση ενός γνωρίσματος που στο πεδίο του τύπου υπάρχει το αντίστοιχο UUID που δηλώνει αναφορά και στο πεδίο της τιμής βρίσκονται η αρχή και το τέλος του handle που ορίζεται η αναφερόμενη υπηρεσία καθώς και το UUID που ορίζει την υπηρεσία καθεαυτή.

- **Characteristics (χαρακτηριστικά):** Τα χαρακτηριστικά αποτελούνται το λιγότερο από δύο γνωρίσματα, την δήλωση του χαρακτηριστικού (παρέχει μεταδεδομένα για τα πραγματικά δεδομένα) και την τιμή του χαρακτηριστικού (ένα γνώρισμα που περιέχει δεδομένα χρήση στο πεδίο της τιμής). Επιπλέον τα χαρακτηριστικά μπορούν να ακολουθούνται από περιγραφείς (descriptors) που προσθέτουν επιπλέον μεταδεδομένα στο υπάρχον χαρακτηριστικό. Η δήλωση, η τιμή και οι επιπλέον περιγραφείς συνθέτουν ένα χαρακτηριστικό.

Handle	Type	Permissions	Value	Value length
0XNNNN	UUID characteristic	Read Only	Properties, value handle (0xMMMM), Characteristic UUID	5,7 or 19 bytes
0xMMMM	Characteristic UUID	Any	Actual Value	Variable

Πίνακας 7: Δήλωση γνωρίσματος χαρακτηριστικού καθώς και γνωρίσματος της τιμής που περιγράφει το χαρακτηριστικό.

Στον ΠΙΝΑΚΑ 7 φαίνεται η δήλωση ενός χαρακτηριστικού με τις αντίστοιχες τιμές στο πεδίο type υπάρχει η δεσμευμένη UUID τιμή που ορίζει αποκλειστικά την έναρξη ενός χαρακτηριστικού. Στον ΠΙΝΑΚΑ 8 αναφέρονται τα περιεχόμενα των διαφόρων πεδίων του ΠΙΝΑΚΑ 7.

Name	Length in bytes	Description
Characteristic Properties	1	Ορίζει τις επιτρεπόμενες ενέργειες που μπορούν να γίνουν στο χαρακτηριστικό.
Characteristic Value Handle	2	Η διεύθυνση του γνωρίσματος που περιέχει την τιμή του χαρακτηριστικού.
Characteristic UUID	2, 4 or 16	Το UUID αυτού του χαρακτηριστικού.

Πίνακας 8: Πεδία δήλωσης του γνωρίσματος του χαρακτηριστικού.

Οι τιμές που μπορεί να πάρει το πεδίο characteristic properties (ιδιότητες χαρακτηριστικού) ορίζουν τις διάφορες ενέργειες και διαδικασίες που μπορούν να γίνουν με το συγκεκριμένο χαρακτηριστικό. Οι ενέργειες αυτές παραθέτονται στον ΠΙΝΑΚΑ 9. Το πεδίο αποτελείται από 1 byte και με έναν επιπλέον descriptor μπορούν να προστεθούν επιπλέον 2 bit για να καλυφθούν όλες οι λειτουργίες που υποστηρίζονται.

Property	Location	Description
Broadcast	Properties	Επιτρέπει στην τιμή του χαρακτηριστικού να μπει στα διαφημιζόμενα πακέτα.
Read	Properties	Επιτρέπει στις client συσκευές να διαβάζουν το χαρακτηριστικό χρησιμοποιώντας την λειτουργία ανάγνωσης του ATT.
Write without response	Properties	Επιτρέπει στις client συσκευές να γράφουν το χαρακτηριστικό χρησιμοποιώντας την λειτουργία εγγραφής του ATT.
Write	Properties	Επιτρέπει στις client συσκευές να γράφουν το χαρακτηριστικό χρησιμοποιώντας την λειτουργία εγγραφής με επιβεβαίωση του ATT.
Notify	Properties	Επιτρέπει στις server συσκευές να στέλνουν notifications αν αλλάξει η τιμή του χαρακτηριστικού με την αντίστοιχη λειτουργία του ATT.
Indicate	Properties	Επιτρέπει στις server συσκευές να στέλνουν indications αν αλλάξει η τιμή του χαρακτηριστικού με την αντίστοιχη λειτουργία του ATT (επιβεβαίωση ότι η αλλαγή της τιμής έχει ληφθεί από τον client).
Signed Write Command	Properties	Επιτρέπει στις server συσκευές να χρησιμοποιήσουν την λειτουργία Signed Write Command του ATT στο χαρακτηριστικό.
Queued Write	Extended properties	Επιτρέπει στις server συσκευές να χρησιμοποιήσουν την λειτουργία Queued Writes του ATT στο χαρακτηριστικό.
Writable Auxiliaries	Extended properties	Επιτρέπει στις server συσκευές να γράψουν σε έναν συγκεκριμένο descriptor που ορίζεται από τον χρήστη.

Πίνακας 99: Ιδιότητες χαρακτηριστικών.

Η συσκευή client μπορεί να διαβάσει αυτές τις ιδιότητες και να ανακαλύψει ποιες λειτουργίες μπορεί να εκτελέσει στα χαρακτηριστικά.

- **Characteristics Descriptors (περιγραφείς):** όπως ειπώθηκε και παραπάνω οι περιγραφείς προσδίδουν μεταδεδομένα στην client συσκευή. Υπάρχουν δύο τύποι descriptor για GATT χαρακτηριστικά:
 - GATT- defined descriptors (περιγραφείς που ορίζονται από το πρωτόκολλο GATT) : αποτελούν τους βασικούς περιγραφείς για τα χαρακτηριστικά και μερικοί από αυτούς είναι:
 - Extended properties descriptor: αποτελεί τον descriptor που προσθέτει 2 επιπλέον bit για να καλυφθούν όλες οι λειτουργίες που υποστηρίζονται από το characteristic properties.
 - Characteristic User Description descriptor: περιέχει μια περιγραφή που είναι αναγνώσιμη από τον χρήστη για το χαρακτηριστικό στο οποίο ανήκει.
 - Client Characteristic Configuration descriptor: αποτελεί ίσως το πιο σημαντικό descriptor εφόσον ενεργεί σαν διακόπτης για όλες τις λειτουργίες που εκκινούνται από τον server για το χαρακτηριστικό που τον περιέχει. Περιλαμβάνει μια τιμή των 2 bit μέσω της οποίας ο χρήστης μπορεί να ενεργοποιήσει τα indications ή τα notifications και στην συνέχεια ο server θα στέλνει ειδοποιήσεις κάθε φορά που η τιμή του συγκεκριμένου χαρακτηριστικού αλλάζει.
 - Characteristic presentation format descriptor: περιέχει την μορφή (format – integer, Boolean κ.α.) της τιμής του χαρακτηριστικού στο οποίο περιλαμβάνεται.

- Profile or vendor – defined descriptors: αποτελούν περιγραφείς που μπορούν να περιέχουν πολλών τύπων δεδομένα, και επιπλέον πληροφορία που αφορά την τιμή των χαρακτηριστικών.

1.5.4 Επιπλέον λειτουργίες γνωρισμάτων

Στην παρούσα παράγραφο αναφέρονται κάποιες επιπλέον λειτουργίες που επιτρέπουν την ευκολότερη χρήση των διαφόρων γνωρισμάτων σε έναν GATT server.

- **Caching γνωρισμάτων:** Ένας client δεν έχει καμμία γνώση για το τι χαρακτηριστικά και υπηρεσίες έχει μια GATT server συσκευή. Όπως είπαμε θα πρέπει να «ανακαλύψει» τα διάφορα handles και τα αντίστοιχα χαρακτηριστικά και υπηρεσίες. Η παραπάνω είναι μια διαδικασία που κοστίζει χρονικά και ενεργειακά. Ένας GATT server συνήθως δεν αλλάζει την δομή των χαρακτηριστικών του κατά την διάρκεια λειτουργίας του, στις περισσότερες περιπτώσεις, αλλά αυτό δεν ορίζεται πουθενά από το πρότυπο. Ένας server είναι σε θέση να αλλάξει την δομή των χαρακτηριστικών του ακόμα και να τα αντικαταστήσει με άλλα εντελώς καινούρια χαρακτηριστικά.

Σαν γενικό κανόνα το πρότυπο ορίζει ότι ο client θα πρέπει να αποθηκεύσει τα handles των γνωρισμάτων, για τα οποία ενδιαφέρεται, και έχει ανακαλύψει κατά την διάρκεια μιας σύνδεσης προκειμένου να κερδίσει χρόνο και ενέργεια σε επικείμενες επανασυνδέσεις. Σε περίπτωση όμως που αυτά τα γνωρίσματα αλλάξουν κατά την διάρκεια λειτουργίας του server το πρότυπο ορίζει ένα επιπλέον γνώρισμα, το Service Changed characteristic προκειμένου ο server να ειδοποιήσει τον client για μια επικείμενη αλλαγή γνωρισμάτων στην συσκευή.

- **Δεδομένα Πρωτοκόλλου GATT σε πακέτα διαφήμισης:** δίνει την δυνατότητα σε δεδομένα γνωρισμάτων που υπάρχουν στον server να συμπεριληφθούν σε πακέτα διαφήμισης προκειμένου να γίνουν γνωστά σε συσκευές που βρίσκονται εντός εμβέλειας.

1.5.5 Λειτουργίες πρωτοκόλλου GATT

Οι λειτουργίες του GATT είναι αυστηρά καθορισμένες διαδικασίες που επιτρέπουν την ανταλλαγή δεδομένων μεταξύ συσκευών. Είναι όλες βασισμένες σε διαφορετικές λειτουργίες του πρωτοκόλλου ATT. Από τις πιο βασικές λειτουργίες του ATT πρωτοκόλλου είναι:

- **Ανταλλαγή μονάδων μεταφοράς (Exchange MTU):** η συγκεκριμένη λειτουργία επιτρέπει στις συσκευές να ανταλλάσουν πληροφορίες σχετικά με το μέγεθος των πακέτων που θα χρησιμοποιήσουν για την ανταλλαγή των δεδομένων. Χρησιμοποιείται σε περίπτωση που μία από τις δύο συσκευές μπορεί να χειριστεί πακέτα δεδομένων μεγαλύτερα από τα κανονικά και θέλει να ενημερώσει την άλλη προκειμένου να λάβει τις ανάλογες ενέργειες.
- **Service and Characteristic Discovery:** Όπως ήδη αναφέραμε ο client δεν γνωρίζει πριν συνδεθεί με τον server τις υπηρεσίες που υποστηρίζονται. Αυτό προϋποθέτει την ύπαρξη διαδικασιών που ανιχνεύουν και αναφέρουν τα διάφορα ήδη γνωρισμάτων που υπάρχουν σε κάποιον server. Μερικές από αυτές τις διαδικασίες είναι:
 - Discover all primary services: με την συγκεκριμένη διαδικασία ο client ενημερώνεται για όλες τις υπηρεσίες που υποστηρίζει ο server. Αυτό γίνεται σε

περίπτωση που ο client μπορεί να υποστηρίξει περισσότερες από μια υπηρεσίες και θέλει να μάθει ποιες από αυτές τις υπηρεσίες υποστηρίζονται από τον server.

- Discover primary service by service UUID: με αυτήν την διαδικασία ο client ψάχνει για μια συγκεκριμένη υπηρεσία που υποστηρίζει, στον server. Έτσι ψάχνει όλα τα handles που έχει ο server προκειμένου να βρει την υπηρεσία που έχει το ίδιο UUID με αυτό που ψάχνει.

Από την στιγμή που ο client βρει την υπηρεσία που ψάχνει κοιτάει να βρει αν η υπηρεσία που τον ενδιαφέρει σχετίζεται με κάποια άλλη υπηρεσία με τις ακόλουθες διαδικασίες.

- Find included services: αυτό επιτρέπει στον server να βρει όλες τις υπηρεσίες που περιλαμβάνονται στις πρωτεύοντες υπηρεσίες.

Στην συνέχεια για τον εντοπισμό των διαφόρων χαρακτηριστικών χρησιμοποιούνται οι ακόλουθες διαδικασίες.

- Discover all characteristics of a service: από την στιγμή που ο client έχει το handle της υπηρεσίας μπορεί να προχωρήσει σε ανάκτηση όλων των χαρακτηριστικών. Ο client δίνει σαν είσοδο το εύρος του handle και στην συνέχεια ο server αποκρίνεται με το handle και τις τιμές όλων των χαρακτηριστικών που περιλαμβάνονται σε αυτήν την υπηρεσία.
- Discover characteristics by UUID: ο client ψάχνει για συγκεκριμένο χαρακτηριστικό μέσα στην υπηρεσία, και απορρίπτει όλα τα χαρακτηριστικά που το UUID τους δεν αντιστοιχεί σε αυτό που ψάχνει.

Στην συνέχεια αφού βρεθούν τα χαρακτηριστικά και το εύρος των handle τους, ο client προχωρά σε αναγνώριση των αντίστοιχων descriptors που αντιστοιχούν στα συγκεκριμένα χαρακτηριστικά.

- Discover all characteristics descriptors: σ' αυτήν την διαδικασία ο client προχωρά στην αναγνώριση των descriptors για το χαρακτηριστικό που τον ενδιαφέρει.
- **Reading Characteristic and Descriptors:** ο client για να διαβάσει τιμές από τα διάφορα χαρακτηριστικά και από τους περιγραφείς έχει τις ακόλουθες επιλογές.
 - Read characteristic value or descriptor: ανάγνωση περιεχομένων τιμής χαρακτηριστικού ή descriptor με την χρήση του handle.
 - Read long characteristic value or descriptor: αν η τιμή που επιθυμεί ο client να διαβάσει είναι πολύ μεγάλη χρησιμοποιείται αυτή η διαδικασία προκειμένου η τιμή να ανακτηθεί σε διαδοχικά κομμάτια.
 - Read characteristic value using characteristic UUID: σε περίπτωση που ο client δεν γνωρίζει το handle του χαρακτηριστικού, μπορεί να ανακτήσει την τιμή του χρησιμοποιώντας το UUID του.
 - Read multiple characteristic values: για ανάκτηση πολλαπλών τιμών μπορεί ο client να στείλει τα handle των χαρακτηριστικών για τα οποία ενδιαφέρεται και να ανακτήσει τις τιμές από τα αντίστοιχα χαρακτηριστικά.
- **Writing Characteristic and Descriptors:** ο client για να αναγνώσει τιμές από τα διάφορα χαρακτηριστικά και από του περιγραφείς έχει τις ακόλουθες επιλογές.

- Write characteristic value or descriptor: ο client παρέχει το handle και την τιμή για να γίνει εγγραφή στο χαρακτηριστικό. Στο τέλος της συνδιαλλαγής ο server θα επιβεβαιώσει την εγγραφή.
- Write long characteristic value or descriptor: παρόμοια διαδικασία με αυτήν της ανάγνωσης αλλά για εγγραφή τιμής μεγαλύτερου μεγέθους.
- Write without response: διαδικασία κατά την οποία ο client κάνει εγγραφή αλλά δεν περιμένει επιβεβαίωση από τον server. Ο server μπορεί και να μην εκτελέσει την συγκεκριμένη εντολή, παρόλα' αυτά ο client δεν μπορεί να το ξέρει εκτός και αν διαβάσει την τιμή.
- Reliable writes: παρόμοια διαδικασία με την ανάκτηση τιμής πολλαπλών χαρακτηριστικών αλλά για εγγραφή. Ο client στέλνει ουρά από δεδομένα και στην συνέχεια στέλνει ένα τελικό πακέτο για να εκτελέσει όλες τις εντολές εγγραφής.
- **Server Initiated Updates:** Είναι ασύγχρονες διαδικασίες που εκκινούνται από τον server, στέλνονται όποτε αλλάξει η τιμή ενός χαρακτηριστικού χωρίς να απαιτείται από τον client να καταγράψει συνέχεια αυτή την τιμή. Οι διαδικασίες αυτές είναι:
 - Characteristic Value Notification: τα Notifications είναι πακέτα που περιλαμβάνουν το handle της τιμής ενός χαρακτηριστικού και την ίδια την τιμή. Ο client όταν λάβει ένα τέτοιο πακέτο κάνει τις απαραίτητες ενέργειες που απαιτεί το application αλλά δεν στέλνει καμία επιβεβαίωση στον server ότι το πακέτο έχει ληφθεί.
 - Characteristic Value Indication: τα Notifications είναι αντίστοιχα πακέτα με τα indications με την διαφορά ότι στέλνεται επιβεβαίωση στον server. Αξίζει να σημειωθεί ότι αν δεν γίνει λήψη επιβεβαίωσης ο server δεν στέλνει άλλα indications αλλά ο μπορεί να ανταποκριθεί σε άλλες εντολές που ενδεχομένως θα του στείλει ο client.

Κεφάλαιο 2 – HID (Human Interface Device)

Το πρότυπο HID αρχικά ορίστηκε σαν μια κλάση συσκευών για το usb πρωτόκολλο. Ο στόχος ήταν να καθοριστεί ένας αντικαταστάτης για το PS/2 και να δημιουργηθεί μια διεπαφή για USB συσκευές που θα επέτρεπε στα λειτουργικά συστήματα να έχουν έναν γενικό οδηγό (driver) για συσκευές όπως πληκτρολόγια ποντίκια κ.α. Πριν την εισαγωγή του HID πρωτοκόλλου όλες οι συσκευές έπρεπε να συμμορφώνονται σε αυστηρά πρωτόκολλα για τις διάφορες συσκευές εισόδου. Όλες οι καινοτόμες συσκευές επέβαλλαν την δημιουργία ενός πρωτοκόλλου πάνω από το οποίο θα μπορούσαν όλες να λειτουργήσουν, αλλιώς η κάθε μια συσκευή θα έτρεχε με τον δικό της ξεχωριστό οδηγό. Ο σκοπός του HID ήταν να υποστηρίξει τις “boot mode” συσκευές που χρησιμοποιούνται σε ένα λειτουργικό σύστημα αλλά και να δώσει στους κατασκευαστές ένα πρότυπο επεκτάσιμο, τυποποιημένο και εύκολο στην χρήση.

Σήμερα το πρωτόκολλο HID περιλαμβάνει αλφαριθμητικές οθόνες, αναγνώστες barcode, ελεγκτές ήχου σε ηχεία/ακουστικά, αισθητήρια κ.α. Ακόμη πολλοί κατασκευαστές βασίζονται στο πρωτόκολλο για τις καινοτομίες τους.

Το HID μπορεί να άρχισε από το USB, αλλά στην ουσία σχεδιάστηκε, χωρίς να προσδιορίζεται κάποιο πρωτόκολλο μεταφοράς, για low latency και low bandwidth συσκευές προκειμένου ο ρυθμός του να καθορίζεται από το μέσο διάδοσης του. Το πρότυπο για το USB εισήχθη στα τέλη του 90 και στην συνέχεια υποστηρίχθηκε και από άλλα μέσα διάδοσης. Σήμερα είναι ένα πρωτόκολλο που υποστηρίζεται από πολλά πρωτόκολλα μεταφοράς όπως Bluetooth, Bluetooth LE, I2C κ.α.

2.1 HID πάνω από USB

Οι πληροφορίες για μια USB συσκευή εισόδου βρίσκονται σε τμήματα της ROM. Αυτά τα τμήματα λέγονται περιγραφείς. Στην ΕΙΚΟΝΑ 7 παραθέτονται όλοι οι περιγραφείς που περιλαμβάνονται σε μια USB συσκευή.

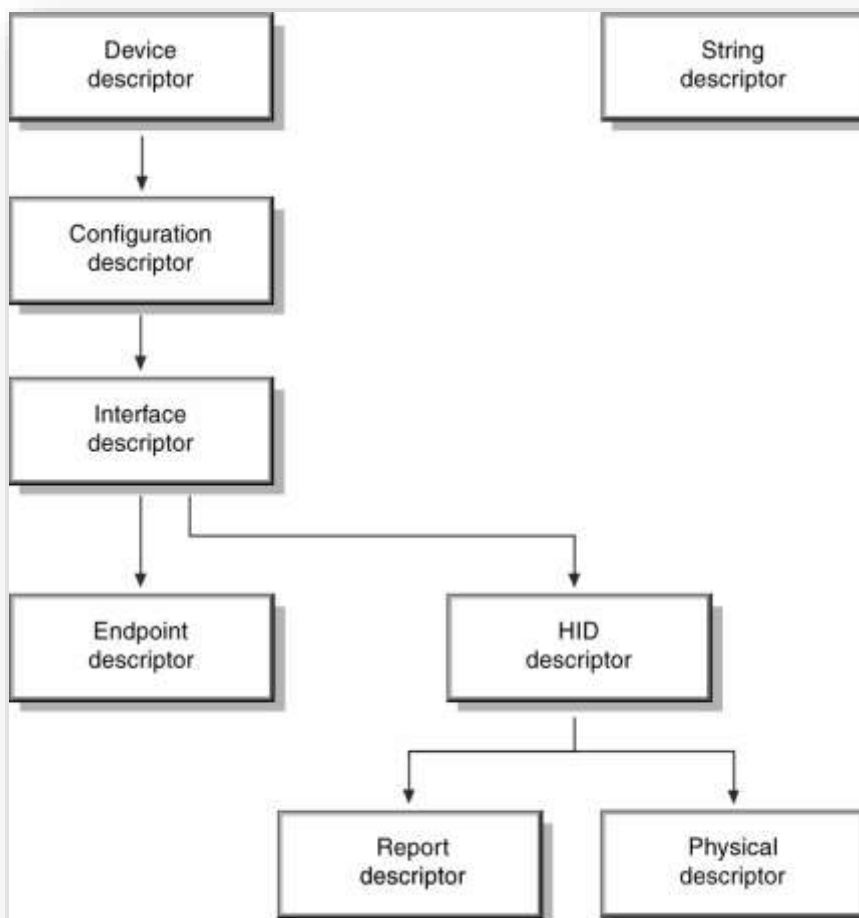
Οι USB συσκευές κατηγοριοποιούνται σε κλάσεις σύμφωνα με τις απαιτήσεις που έχουν για την μεταφορά δεδομένων και με τους οδηγούς που απαιτούν. Μια συσκευή μπορεί να ανήκει σε μία ή και περισσότερες κλάσεις. Για παράδειγμα ένα τηλέφωνο μπορεί να περιλαμβάνει χαρακτηριστικά που ανήκουν στις HID, Audio και Telephony κλάσεις. Οι κλάσεις στις οποίες ανήκει η συσκευή προσδιορίζονται από τον περιγραφέα διεπαφής και όχι από τον περιγραφέα συσκευής.

Ένας περιγραφέας διεπαφής (Interface Descriptor) ορίζει σε ποιο είδος ανήκει η συσκευή μέσα από μία πεπερασμένη λίστα κλάσεων. Με την χρήση αυτού του περιγραφέα διεπαφής η συσκευή (host) που είναι συνδεδεμένη η συσκευή εισόδου μπορεί να χρησιμοποιήσει τον αντίστοιχο οδηγό για να ανακαλέσει, και να δρομολογήσει τα δεδομένα που δέχεται. Αν η συσκευή ανήκει στην HID κλάση τότε ο περιγραφέας διεπαφής ορίζει τον περιγραφέα HID.

Ένας περιγραφέας HID συσκευής προσδιορίζει πόσοι και ποιοί άλλοι περιγραφείς υπάρχουν στην συσκευή καθώς και το μέγεθός τους. Υπάρχουν δύο είδη περιγραφέων HID:

- Περιγραφέας αναφοράς (Report Descriptor): προσδιορίζει τα κομμάτια της πληροφορίας που παράγονται από την συσκευή, και τι ελέγχουν. Για παράδειγμα μπορεί να περιγράψει την θέση ή την κατάσταση στην οποία βρίσκεται ένα κουμπί.

- Σωματικός περιγραφέας (Physical Descriptor): είναι προαιρετική η χρήση του και παρέχει πληροφορίες σχετικά με τα τμήματα του ανθρωπίνου σώματος που χρησιμοποιούνται προκειμένου να ενεργοποιήσουν τις λειτουργίες της συσκευής.



Εικόνα 77: Περιγραφείς που περιλαμβάνονται σε μια USB συσκευή.

2.2 Περιγραφείς αναφοράς (Report Descriptors)

Οι περιγραφείς αναφοράς σε αντίθεση με τους άλλους περιγραφείς δεν είναι απλά πίνακες με τιμές. Το μέγεθος και το περιεχόμενο των περιγραφέων αναφοράς, μεταβάλλεται ανάλογα με τον αριθμό των πεδίων των δεδομένων που απαιτούνται από την συσκευή προκειμένου να περιγραφούν οι λειτουργίες της.

Οι περιγραφείς αναφοράς αποτελούνται από αντικείμενα, τα οποία παρέχουν πληροφορίες για την συσκευή. Το αρχικό κομμάτι κάθε αντικειμένου αποτελείται από τρία πεδία και κάθε πεδίο μπορεί να πάρει αντίστοιχες τιμές. Τα πεδία αυτά καθώς είναι:

- Τύπος αντικειμένου (item type): υπάρχουν τρεις διαφορετικοί τύποι αντικειμένου:
 - Main
 - Global
 - Local
- Σήμανση αντικειμένου (item tag): προσδιορίζει την λειτουργία του αντικειμένου.
- Μέγεθος αντικειμένου (item size): ενδεικτική τιμή μήκους δεδομένων.

Για τον τύπο αντικειμένου Main οι τιμές που μπορεί να πάρει είναι:

- Input: αναφέρεται σε δεδομένα από ένα ή περισσότερα παρόμοια σημεία ελέγχου σε μια συσκευή. Μία εφαρμογή μπορεί να χρησιμοποιήσει αυτήν την πληροφορία για να ερμηνεύσει τα δεδομένα που προέρχονται από την συσκευή.
- Output: χρησιμοποιείται για να ορίσει δεδομένα εξόδου σε έναν περιγραφέα αναφοράς. Είναι παρόμοιο με το αντικείμενο εισόδου (input) με την διαφορά ότι περιγράφει δεδομένα που στέλνονται προς την συσκευή εισόδου για παράδειγμα ενεργοποίηση ενδεικτικής λυχνίας LED.
- Feature: χρησιμοποιείται για την περιγραφή τυχόν ιδιοτήτων που μπορεί να έχει η συσκευή, για παράδειγμα ενεργοποίηση κάποιων επιπλέον ρυθμίσεων που μπορεί να έχει.
- Collection: χρησιμοποιείται για να ομαδοποιήσει τα input, output και feature αντικείμενα.
- End Collection: χρησιμοποιείται σαν σήμανση λήξης μίας ομαδοποίησης αντικειμένων.

Για τους τύπους αντικειμένων Local και Global υπάρχει πληθώρα επιλογών που παρατίθεται στο έγγραφο [1].

Όλα μαζί τα παραπάνω πεδία προσδιορίζουν το είδος της πληροφορίας που παρέχει το αντικείμενο και στην συνέχεια ακολουθούν τα δεδομένα που περιγράφονται από τα πεδία.

Οι περιγραφείς αναφοράς παρέχουν μια περιγραφή των δεδομένων που δημιουργούνται από κάθε σημείο ελέγχου σε μία συσκευή. Κάθε Main τύπος αντικειμένου (input, output ή feature) προσδιορίζει το μέγεθος των δεδομένων που επιστρέφει κάθε σημείο ελέγχου, αν τα δεδομένα που παρέχει είναι απόλυτα ή σχεσιακά καθώς και άλλες σχετικές πληροφορίες. Όλα τα προηγούμενα, στην σειρά, Local ή global αντικείμενα προσδιορίζουν τις μέγιστες και τις ελάχιστες τιμές που μπορούν να πάρουν τα διάφορα σημεία ελέγχου και ου το καθ' εξής. Από τον περιγραφέα αναφοράς και μόνο, μια εφαρμογή μπορεί να χειριστεί όλα τα εισερχόμενα δεδομένα.

Ένα ή περισσότερα πεδία δεδομένων από σημεία ελέγχου προσδιορίζονται από ένα Main αντικείμενο και περιγράφονται επιπλέον από προηγούμενα Global και Local αντικείμενα. Τα Local αντικείμενα περιγράφουν μόνο τα πεδία δεδομένων που ορίζονται από το επόμενο

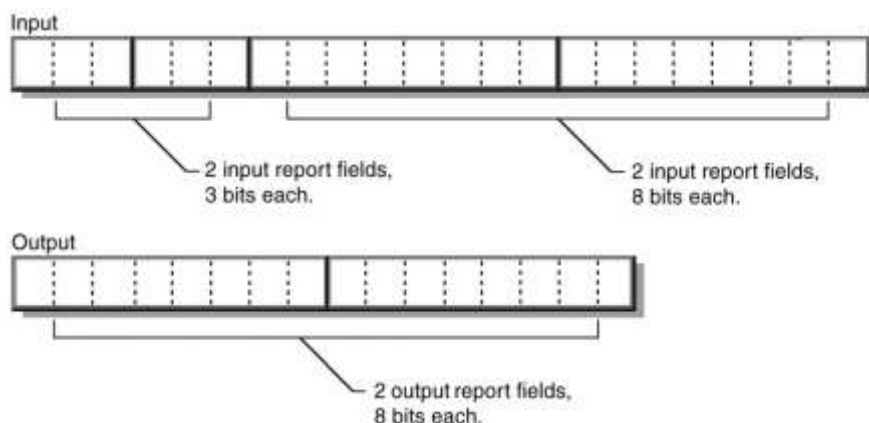
Main αντικείμενο. Τα Global αντικείμενα αποτελούν τις αρχικές τιμές για όλα τα επόμενα πεδία δεδομένων στον παρόντα περιγραφέα.

Για παράδειγμα αν θεωρήσουμε τον παρακάτω περιγραφέα στην ΕΙΚΟΝΑ 8 (οι λεπτομέρειες παραλείπονται για συντομία).

```
Report Size (3)
Report Count (2)
Input
Report Size (8)
Input
Output|
```

Εικόνα 88: Παράδειγμα περιγραφέα (descriptor).

Ο Item Parser θα ερμηνεύσει τον παραπάνω περιγραφέα και θα δημιουργήσει τις ακόλουθες αναφορές (ΕΙΚΟΝΑ 9).



Εικόνα 99: Ερμηνείας περιγραφέα εικόνας 8 από τον item parser της συσκευής Host.

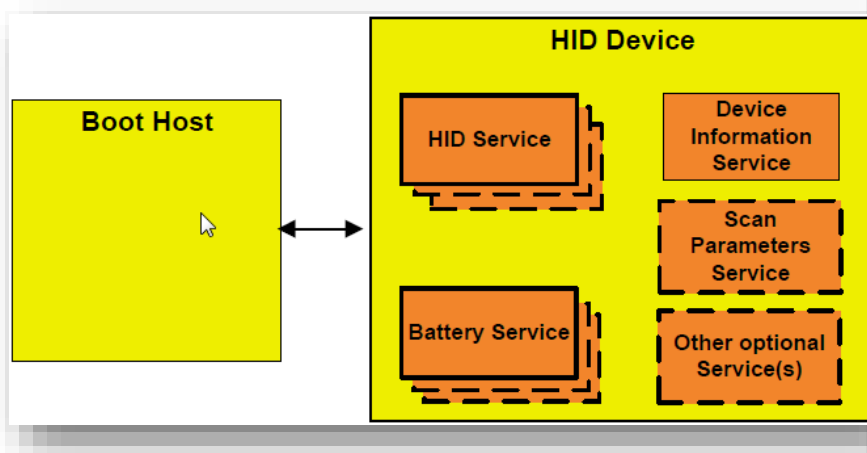
2.3 HID με χρήση σύνδεσης Bluetooth

Στην παράγραφο 2.1 αναλύθηκε το πρωτόκολλο HID πάνω από USB και στην συνέχεια, στην παράγραφο 2.2, έγινε μια αναφορά στο πως γίνεται η μεταφορά των δεδομένων από την συσκευή εισόδου. Στην παρούσα παράγραφο θα αναλυθεί η λειτουργία του HID με την χρήση πρωτοκόλλου μεταφοράς Bluetooth Low Energy.

Για να λειτουργήσει μία συσκευή BLE σαν συσκευή εισόδου θα πρέπει να υποστηρίζει το αντίστοιχο προφίλ (HID over GATT profile). Το προφίλ ορίζει όλες τις διαδικασίες που θα χρησιμοποιηθούν από τις BLE HID συσκευές καθώς και από τις συσκευές στις οποίες θα συνδεθούν. Είναι μια μορφοποίηση του προτύπου HID over USB προκειμένου να λειτουργεί μέσω μίας σύνδεσης Bluetooth.

Το συγκεκριμένο προφίλ απαιτεί από την συσκευή εισόδου να υποστηρίζει κάποια επιπλέον προφίλ προκειμένου να λειτουργήσει. Τα προφίλ αυτά είναι:

- Generic Attribute Profile (GATT): απαραίτητο για όλες τις BLE συσκευές.
- Battery Service: υπηρεσία που αφορά την μπαταρία του client.
- Device Information Service: υπηρεσία που αφορά πληροφορίες του client
- Scan Parameters Profile (optional): υπηρεσία προκειμένου ο client να γνωρίζει τις παραμέτρους ανίχνευσης του host.



Εικόνα 1039: Σχέση μεταξύ υπηρεσιών και προφίλ.

Το προφίλ υποστηρίζει τρεις ρόλους:

- HID Device: είναι η συσκευή εισόδου (GATT Server).
- Boot Host: είναι η host συσκευή (GATT Client).
- Report Host: είναι η host συσκευή (GATT Client). Η συσκευή με τον συγκεκριμένο ρόλο είναι απαραίτητο να υποστηρίζει HID Parser προκειμένου να μπορεί να ερμηνεύει αυθαίρετους περιγραφείς αναφοράς (Report Descriptors) που χρησιμοποιούνται για την μεταφορά δεδομένων. Σε αντίθεση με τον ρόλο Boot Host που οι περιγραφείς αναφοράς είναι προκαθορισμένοι όσον αφορά το μέγεθος και την δομή.

Όπως ειπώθηκε προηγουμένως το προφίλ HID για λειτουργία μέσω Bluetooth ορίζει κάποιες συγκεκριμένες διαδικασίες, προκειμένου ο Host να «ανακαλύψει» τις υπηρεσίες που υποστηρίζει η συσκευή που θα συνδεθεί, καθώς και κάποια απαραίτητα χαρακτηριστικά που θα πρέπει να υπάρχουν στην HID συσκευή για την ομαλή λειτουργία τους.

2.3.1 Διαδικασίες HID με χρήση σύνδεσης Bluetooth

Το πρότυπο BLE ορίζει κάποιες διαδικασίες που κάθε συσκευή (HID Host) θα πρέπει να υποστηρίζει. Αυτές οι διαδικασίες χρησιμοποιούνται από τον Host προκειμένου να μπορέσει να ανακαλύψει τις διαθέσιμες υπηρεσίες, τα χαρακτηριστικά και τις λειτουργίες σε κάποιον Client

και να μπορέσει να ανακτήσει δεδομένα. Στους παρακάτω πίνακες παρατίθενται όλες οι απαραίτητες και μη, διαδικασίες που ορίζει το HID Profile καθώς και τα χαρακτηριστικά που απαιτούνται από κάθε HID Device. Επίσης το ότι το συγκεκριμένο προφίλ υποστηρίζει διαφορετικούς ρόλους από την πλευρά του Host (Boot Host, Report Host) αυτό αλλάζει και τις απαιτήσεις διαδικασιών που ορίζονται από το ίδιο το profile. Στον ΠΙΝΑΚΑ 10 παρατίθενται όλες οι διαδικασίες και τα χαρακτηριστικά που πρέπει να υποστηρίζονται από τις HID συσκευές και στον ΠΙΝΑΚΑ 11 παρατίθενται όλες οι GATT υπό-διαδικασίες με τις οποίες χτίζονται οι ολοκληρωμένες διαδικασίες που απαιτούνται από το προφίλ. Στις μετέπειτα παραγράφους δίνονται αναλυτικά οι λειτουργίες μερικών διαδικασιών για Boot Host αλλά και για Report Host καθώς και μια περιγραφή των χαρακτηριστικών που βρίσκονται στις HID συσκευές.

Διαδικασίες	Boot Host απαιτήσεις	Report Host απαιτήσεις
Ανίχνευση Υπηρεσιών	Υποχρεωτικό	Υποχρεωτικό
• HID Service Discovery	Υποχρεωτικό	Υποχρεωτικό
• Device Information Service Discovery	Προαιρετικό	Υποχρεωτικό
• Battery Service Discovery	Προαιρετικό	Υποχρεωτικό
Ανίχνευση Χαρακτηριστικών	Προαιρετικό	Υποχρεωτικό
• HID Service Characteristic Discovery	Προαιρετικό	Υποχρεωτικό
• Device Information Service Characteristic Discovery	Προαιρετικό	Υποχρεωτικό
• Battery Service Characteristic Discovery	Προαιρετικό	Υποχρεωτικό
Report Map	X	Υποχρεωτικό
Report	X	Υποχρεωτικό
Boot Keyboard Input Report	C.2/ C.3	X
Boot Keyboard Output Report	C.2/ C.3	X
Boot Mouse Input Report	C.2	X
HID Information	X	Υποχρεωτικό
HID Control Point	X	C.1
Protocol Mode	Υποχρεωτικό	Προαιρετικό
Non-HID Service characteristic defined within Report Map	X	Υποχρεωτικό
C.1: Mandatory if the Host supports Suspend Mode, otherwise optional.		
C.2: Mandatory to support at least one of these features.		
C.3: If one of these features is supported, both features shall be supported.		

Πίνακας 1049: Απαιτήσεις χαρακτηριστικών από HID συσκευές για Boot Host και Report Host.

GATT υπό-Διαδικασίες	Boot Host απαιτήσεις	Report Host απαιτήσεις
Discover All Primary Services	C.1	C.1
Discover Primary Services by Service UUID	C.1	C.1
Discover All Characteristics of a Service	Προαιρετικό	C.2

Discover Characteristics by UUID	Προαιρετικό	C.2
Discover All Characteristics Descriptors	Υποχρεωτικό	Υποχρεωτικό
Find Included Services	X	Υποχρεωτικό
Write Without Response	Υποχρεωτικό	Υποχρεωτικό
Write Characteristic Value	Υποχρεωτικό	Υποχρεωτικό
Notifications	Υποχρεωτικό	Υποχρεωτικό
Read using Characteristic UUID	Υποχρεωτικό	Υποχρεωτικό
Read Characteristic Value	Υποχρεωτικό	Υποχρεωτικό
Read Long Characteristic Value	X	Υποχρεωτικό
Read Characteristic Descriptors	Υποχρεωτικό	Υποχρεωτικό
Write Characteristic Descriptors	Υποχρεωτικό	Υποχρεωτικό
C.1: Mandatory to support at least one of these sub-procedures.		
C.2: Mandatory to support at least one of these sub-procedures.		

Πίνακας 1144: Επιπλέον διαδικασίες GATT που απαιτούνται για την λειτουργία του HID profile. [1] [2] [3] [4] [5]

2.3.2 Ανίχνευση υπηρεσιών (Service Discovery)

Η διαδικασία για την ανίχνευση των υπηρεσιών επιτρέπει στον HID Host να αναγνωρίσει όλες τις διαθέσιμες υπηρεσίες που υπάρχουν σε έναν Client. Ο Host μπορεί να χρησιμοποιήσει τις υπό - διαδικασίες GATT, Discover All Primary Services καθώς και Discover Primary Services by Service UUID προκειμένου να ανακαλύψει τις πρωτεύοντες υπηρεσίες που απαιτούνται από μια HID συσκευή. Ο Boot Host δεν απαιτεί τις υπηρεσίες Device Information Service και Battery Service αλλά μόνο την HID Service. Σε αντίθεση με τον Report Host όπου απαιτούνται όλες οι υπηρεσίες. Το ότι ένας Report Host απαιτεί κάποιες επιπλέον υπηρεσίες για την λειτουργία του οφείλεται στο ενδεχόμενο ότι ο HID Descriptor που έχει η συσκευή να περιλαμβάνει αναφορές σε αυτές τις υπηρεσίες (για παράδειγμα η ένδειξη για την μπαταρία να περιλαμβάνεται στον HID Descriptor). Σε αντίθεση με τον Boot Host όπου η δομή και το μέγεθος του HID Descriptor είναι προτυποποιημένα. Ακόμη σε περίπτωση που ο Report Host υποστηρίζει μεγαλύτερα πακέτα μεταφοράς, μπορεί να επικοινωνήσει με τον Client προκειμένου να συμφωνήσουν σε ανταλλαγή πακέτων μεγαλύτερου μεγέθους.

2.3.3 Ανίχνευση χαρακτηριστικών (Characteristic Discovery)

Στην συνέχεια αφού οι υπηρεσίες έχουν ανακτηθεί ο Host συνεχίζει με εύρεση των χαρακτηριστικών. Τα χαρακτηριστικά που ανιχνεύονται από τον Host αντιστοιχούν στις υπηρεσίες που υποστηρίζονται από την συσκευή. Στο παρόν έγγραφο γίνεται αναφορά μόνο των χαρακτηριστικών που αντιστοιχούν στην HID υπηρεσία. Οι δύο ρόλοι που ορίζονται για τον Host αναζητούν διαφορετικά χαρακτηριστικά στον Client προκειμένου να λειτουργήσουν:

- Ο Boot Host μπορεί να χρησιμοποιήσει τις GATT υπό – διαδικασίες Discover All Characteristics of a Service ή Discover Characteristics by UUID, αν αυτές υποστηρίζονται, προκειμένου να ανακαλύψει τα ακόλουθα χαρακτηριστικά για κάθε HID υπηρεσία σε έναν GATT Server:
 - Protocol Mode characteristic: εφόσον αφορά χαρακτηριστικό σε συσκευή που είναι Boot Host, το χαρακτηριστικό αυτό περιέχει την τιμή Boot Protocol Mode.

Περισσότερες πληροφορίες σχετικά δίνονται στο Protocol Mode Characteristic του Report Host.

- Boot Keyboard Input Report characteristic: είναι το χαρακτηριστικό που χρησιμοποιείται για την μεταφορά των HID Report σε περίπτωση που η HID συσκευή είναι κάποιο πληκτρολόγιο και υποστηρίζει Boot Protocol Mode.
 - Client Characteristic Configuration descriptor: ο Client ενεργοποιεί τα notifications μέσω αυτού του περιγραφέα.
- Boot Keyboard Output Report characteristic: το χαρακτηριστικό αυτό χρησιμοποιείται για την μεταφορά HID δεδομένων τα οποία αναπαριστούν την κατάσταση των LED πάνω σε ένα πληκτρολόγιο.
- Boot Mouse Input Report characteristic: είναι το χαρακτηριστικό που χρησιμοποιείται για την μεταφορά των HID Report σε περίπτωση που η HID συσκευή είναι κάποιο ποντίκι και υποστηρίζει Boot Protocol Mode.
 - Client Characteristic Configuration descriptor: ο Client ενεργοποιεί τα notifications μέσω αυτού του περιγραφέα.

Επιπλέον με την υπό-διαδικασία Discover All Characteristic Descriptors γίνεται η ανίχνευση των διαφόρων περιγραφέν χαρακτηριστικών, αν αυτοί υπάρχουν. Αν όμως δεν υποστηρίζονται οι διαδικασίες για την ανίχνευση χαρακτηριστικών ο Boot Host θα πρέπει να χρησιμοποιήσει την υπό – διαδικασία GATT, Read Using Characteristic UUID προκειμένου να αναγνώσει τα παραπάνω HID χαρακτηριστικά για Boot Mode λειτουργία αντικαθιστώντας την κανονική διαδικασία για εύρεση χαρακτηριστικών.

- Ο **Report Host** προκειμένου να ανακαλύψει τα χαρακτηριστικά του Client μπορεί να χρησιμοποιήσει τις υπό – διαδικασίες GATT Discover all Characteristics of a Service ή την Discover Characteristics by UUID. Στην συνέχεια χρησιμοποιεί την GATT υπό – διαδικασία Discover All Characteristic Descriptors προκειμένου να ανακαλύψει τους περιγραφείς των διαφόρων χαρακτηριστικών αν υπάρχουν. Τα χαρακτηριστικά που πρέπει να ανακαλύψει ο Report Host είναι:
 - **Report Map Characteristic:** όταν αναγνωστεί το παρόν χαρακτηριστικό, επιστρέφει τον HID Descriptor της συσκευής. Ο Report Host θα πρέπει να ανακτήσει το χαρακτηριστικό προκειμένου να μπορέσει να ερμηνεύσει την πληροφορία που πρόκειται να ανακτήσει από το Report χαρακτηριστικό.
 - **Report Characteristics:** είναι το χαρακτηριστικό που μεταφέρει τα HID δεδομένα μεταξύ του Report Host και της HID συσκευής.
 - **Client Characteristic Configuration descriptor:** ο περιγραφέας που αντιστοιχεί στο συγκεκριμένο χαρακτηριστικό ενεργοποιεί τις ειδοποιήσεις (notifications). Σε περίπτωση που αλλάξει η τιμή του χαρακτηριστικού Report, ο Report Host θα ειδοποιηθεί για την αλλαγή, εφόσον τα notifications για το συγκεκριμένο χαρακτηριστικό είναι ενεργοποιημένα.
 - **Report Reference Characteristic descriptor:** Ο τύπος του Report (input, output, feature) που υπάρχει στην τιμή του χαρακτηριστικού Report Characteristic, καθώς και το ID (διαφορετικό ID για κάθε HID Report) του δίνεται από τον συγκεκριμένο περιγραφέα.

- **HID Control Point Characteristic:** το χαρακτηριστικό επιτρέπει στον Report Host να σηματοδοτεί την HID συσκευή ότι εισέρχεται ή εξέρχεται από λειτουργία χαμηλής κατανάλωσης.
- **HID Information Characteristic:** το παρόν χαρακτηριστικό περιέχει τις τιμές bcdHID και bcountryCode όπως αυτές ορίζονται από το USB/HID πρότυπο [1]. Επιπλέον στο χαρακτηριστικό περιέχονται δύο flag (Remote Wakeup flag και Normally Connectable flag). Το Remote Wakeup flag χρησιμοποιείται από τον Report Host και ορίζει αν η HID συσκευή είναι σε θέση να «ξυπνήσει» τον Host σε περίπτωση που αυτός είναι σε Suspend Mode. Το Normally Connectable flag χρησιμοποιείται προκειμένου η HID συσκευή να παραμένει διαθέσιμη για σύνδεση προκειμένου ο Host να εκκινήσει την διαδικασία σύνδεσης.
- **Protocol Mode Characteristic:** το παρόν χαρακτηριστικό περιέχει την λειτουργία πρωτοκόλλου (Report ή Host Protocol). Το HID πρότυπο ορίζει δύο πρωτόκολλα για την δημιουργία και την ανάλυση (parsing) των reports, το Report Protocol και το Boot Protocol. Το Report Protocol είναι σχεδιασμένο προκειμένου να υποστηρίζει κάθε HID συσκευή με όλες τις δυνατότητες της, ενώ το Boot Protocol υποστηρίζει περιορισμένο αριθμό συσκευών με συγκεκριμένη μορφοποίηση στα report. Η λειτουργία Report Protocol είναι η αρχική για όλες τις HID συσκευές. Προκειμένου να εκτελεστεί, ο Host θα πρέπει να περιέχει HID Report Descriptor Parser προκειμένου να μπορεί να ερμηνεύει τον περιγραφέα της συσκευής, σε αντίθεση με το Boot Protocol Mode.

Μια συσκευή HID βέβαια μπορεί να λειτουργεί και σαν Boot αλλά και σαν Report Host. Αυτό σημαίνει πως τα χαρακτηριστικά που περιλαμβάνονται στην συσκευή είναι όλα τα παραπάνω (για παράδειγμα ένα πληκτρολόγιο που μπορεί να λειτουργήσει ενώ ο Host έχει φορτώσει το λειτουργικό του σύστημα - Report Host - αλλά και ενώ ο Host είναι στην διαδικασία εκκίνησης ή σε κάποια λειτουργία που δεν υποστηρίζεται η λειτουργία του HID Parser - BIOS).

Τέλος μια συσκευή Report Host μπορεί να έχει περισσότερα από ένα Report Characteristics, για παράδειγμα μια συσκευή που έχει πληκτρολόγιο και κάποια περιοχή αφής μπορεί να αναφέρει στον Report Host ξεχωριστά τα δύο διαφορετικά τμήματα της συσκευής. Ο Report Host μπορεί και αντιλαμβάνεται ποιο mapping πρέπει να χρησιμοποιήσει για κάθε διαφορετικό report που παίρνει από την συσκευή με την βοήθεια του report ID εφόσον είναι διαφορετικό για κάθε τύπο report.

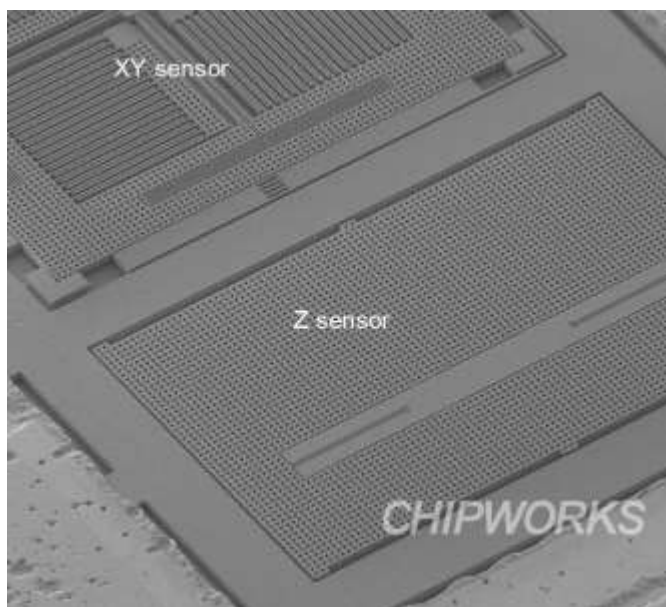
Κεφάλαιο 3 – Αισθητήρες Αδράνειας (Inertia Measurement Units)

Στην παρόν κεφάλαιο αναφέρονται γενικές πληροφορίες σχετικά με τα είδη των αισθητηρίων που χρησιμοποιήθηκαν στην εφαρμογή και στην συνέχεια γίνεται αναφορά στο ολοκληρωμένο κύκλωμα που χρησιμοποιήθηκε για την υλοποίηση της εργασίας.

Οι αισθητήρες αδράνειας (IMU) είναι ηλεκτρονικά εξαρτήματα που μετράνε και αναφέρουν την ταχύτητα, τον προσανατολισμό και τις δυνάμεις της βαρύτητας που ασκούνται πάνω σε κάποιο αντικείμενο. Η μέτρηση των διαφόρων μεγεθών γίνεται με την χρήση επιταχυνσιόμετρων, γυροσκοπίων και μαγνητικών πυξίδων. Οι αισθητήρες αδράνειας βρίσκουν εφαρμογή στα συστήματα πλοήγησης που χρησιμοποιούνται στην ναυσιπλοΐα, στην αεροπλοΐα και όπου είναι απαραίτητη η γνωστοποίηση της θέσης και του προσανατολισμού ενός αντικειμένου.

Πλέον όλα τα αισθητήρια που απαρτίζουν ένα IMU στηρίζονται στην τεχνολογία MEMS (Microelectromechanical systems). Με τον όρο MEMS ορίζουμε την τεχνολογία η οποία δίνει την δυνατότητα να κατασκευάσουμε πολύ μικρές σε μέγεθος μηχανικές κατασκευές όπως αισθητήρια, βαλβίδες, ενεργοποιητές και να τα ενσωματώσουμε σε ολοκληρωμένα chip. Στην ουσία μια MEMS συσκευή περιέχει ένα μικρό κύκλωμα σε ένα chip μέσα στο οποίο υπάρχει επίσης και κάποια μηχανική κατασκευή όπως για παράδειγμα ένα αισθητήριο. Στην ΕΙΚΟΝΑ 11 παρατίθεται φωτογραφία ενός MEMS επιταχυνσιόμετρου τριών αξόνων.

Τα ολοκληρωμένα που στηρίζονται σε MEMS τεχνολογία αποτελούνται από εξαρτήματα μεγέθους από 1 έως και 100 micrometers, και γενικότερα οι συσκευές που είναι βασισμένες στην ίδια τεχνολογία κυμαίνονται σε μέγεθος από 20 micrometers έως και 1 millimeter.



Εικόνα 1144: επιταχυνσιόμετρο 3 αξόνων της ST-Microelectronics

3.1 Εξαρτήματα Αισθητηρίων Αδράνειας

Στην εργασία το IMU που χρησιμοποιήθηκε περιέχει ένα επιταχυνσιόμετρο τριών αξόνων και ένα γυροσκόπιο τριών αξόνων. Επίσης στην πλακέτα που βρίσκεται το IMU, διατίθεται επιπλέον ένα αισθητήριο μαγνητικού πεδίου και ένα αισθητήριο ατμοσφαιρικής πίεσης. Στην παρούσα παράγραφο θα αναφερθεί η λειτουργία των τριών αισθητηρίων (επιταχυνσιόμετρο, γυροσκόπιο και μαγνητόμετρο) εφόσον είναι τα τρία πιο συνηθισμένα αισθητήρια που μπορεί κάποιος να συναντήσει σε ένα IMU.

3.1.1 Επιταχυνσιόμετρο (Accelerometer)

Το επιταχυνσιόμετρο είναι μια ηλεκτρομηχανική κατασκευή που είναι ικανή να μετράει δυνάμεις επιτάχυνσης. Αυτές οι δυνάμεις μπορεί να είναι στατικές, όπως για παράδειγμα, η συνεχής επιτάχυνση της βαρύτητας που ασκεί η γή, ή, μπορεί να είναι και δυναμικές όπως για παράδειγμα όταν το αισθητήριο κουνηθεί. Η πιο συνηθισμένη εφαρμογή των επιταχυνσιόμετρων είναι συνήθως ο προσδιορισμός της γωνία κλίσης μιας συσκευής (Smartphone) σε σχέση με την γη. Παρόλα αυτά, τα συγκεκριμένα αισθητήρια χρησιμοποιούνται προκειμένου να γίνει ανάλυση κίνησης ενός αντικειμένου, στην αυτοκινητοβιομηχανία για συστήματα προστασίας κατά την διάρκεια πρόσκρουσης και καταγραφή δονήσεων στην μηχανή του αυτοκινήτου και επίσης μεγάλη εφαρμογή βρίσκουν στα συστήματα πλοήγησης για UAV (Unmanned Aerial Vehicle). Ακόμη πολλές εταιρίες τα έχουν ενσωματώσει στους σκληρούς δίσκους προκειμένου να προσφέρουν προστασία δεδομένων σε περίπτωση πτώσης του

υπολογιστή και γενικά οι τομείς εφαρμογής τους περιορίζονται από την εφευρετικότητα των χρηστών τους.

Όσον αφορά την αρχή λειτουργία τους, οι δύο πιο συνηθισμένοι τύποι επιταχυνσιόμετρων είναι οι εξής:

- **Πιεζοηλεκτρικά (Piezoelectric):** τέτοια επιταχυνσιόμετρα χρησιμοποιούν το πιεζοηλεκτρικό φαινόμενο, περιέχουν ένα μικροσκοπικό κρύσταλλο, στον οποίο όταν εφαρμόζονται δυνάμεις επιτάχυνσης παράγεται τάση.
- **Χωρητικά (Capacitive):** αυτού του είδους τα αισθητήρια μετράνε αλλαγές στην χωρητικότητα ενός μηχανικού συστήματος που μοιάζει με πυκνωτή. Δύο μικρο-μηχανικά συστήματα το ένα δίπλα στο άλλο έχουν συγκεκριμένη χωρητικότητα. Αν σε κάποιο από αυτά εφαρμοστεί κάποια δύναμη επιτάχυνσης αυτό θα μετακινηθεί και κατά συνέπεια θα αλλάξει η τιμή της χωρητικότητας του συστήματος. Ένα επιπρόσθετο κύκλωμα θα μετατρέψει την χωρητικότητα σε τάση, ανάλογη της επιτάχυνσης που δέχθηκε το σύστημα.

Τα επιταχυνσιόμετρα προσδιορίζονται από κάποια χαρακτηριστικά που ορίζουν το εύρος των μετρήσεων που μπορεί να αναφέρει η συσκευή, η διεπαφή που έχει για να συνδεθεί με κάποιο άλλο ηλεκτρονικό κύκλωμα κ.α. Παρακάτω αναφέρουμε τα πιο σημαντικά από αυτά τα χαρακτηριστικά.

- **Αναλογικό/Ψηφιακό:** ορίζει το είδος της διεπαφής που έχει το αισθητήριο προκειμένου να συνδεθεί με κάποιο μικροεπεξεργαστή. Τα αναλογικά επιταχυνσιόμετρα έχουν σαν έξοδο συνεχόμενη τάση η οποία είναι ανάλογη με την επιτάχυνση που δέχεται το αισθητήριο. Τα αναλογικά αισθητήρια απαιτούν την χρήση μετατροπέα από αναλογικό σε ψηφιακό σήμα (A2D converter) προκειμένου να συνδεθούν με κάποιο επεξεργαστή. Τα ψηφιακά επιταχυνσιόμετρα χρησιμοποιούν διεπαφές όπως I2C ή SPI, και η πληροφορία έρχεται έτοιμη από το αισθητήριο στον επεξεργαστή χωρίς να χρειαστεί οποιαδήποτε μετατροπή.
- **Αριθμός αξόνων:** είναι ο αριθμός των διαστάσεων που μπορεί να μετρήσει την επιτάχυνση το αισθητήριο. Τα περισσότερα αισθητήρια μπορούν να μετρήσουν επιτάχυνση και στις τρεις διαστάσεις (x, y, z).
- **Εύρος μετρήσεων:** είναι η μέγιστη τιμή μέτρησης που μπορεί να επιστρέψει το αισθητήριο. Για παράδειγμα αν η εφαρμογή απαιτεί την μέτρηση επιταχύνσεων κοντά στην επιτάχυνση της βαρύτητας $\pm 1g$ τότε ένα αισθητήριο με εύρος $\pm 1.5g$ αρκεί. Αν όμως οι επιταχύνσεις σχετίζονται με απότομες κινήσεις και δονήσεις τότε θα πρέπει και το εύρος του αισθητηρίου να αυξηθεί.
- **Ευαισθησία:** αφορά την ακρίβεια των μετρήσεων που μπορεί να παρέχει το αισθητήριο.
- **Συχνότητα μέτρησης:** αφορά το πόσο συχνά η συσκευή είναι σε θέση να πάρει αξιόπιστες μετρήσεις. Όσο πιο γρήγορα κινείται το αντικείμενο που θέλουμε να μετρήσουμε τόσο θα πρέπει να αυξάνεται και ο ρυθμός μέτρησης.



Εικόνα 12: Αναλογικό επιταχυνσιόμετρο



Εικόνα 13: Ψηφιακό επιταχυνσιόμετρο

3.1.2 Γυροσκόπιο (Gyroscope)

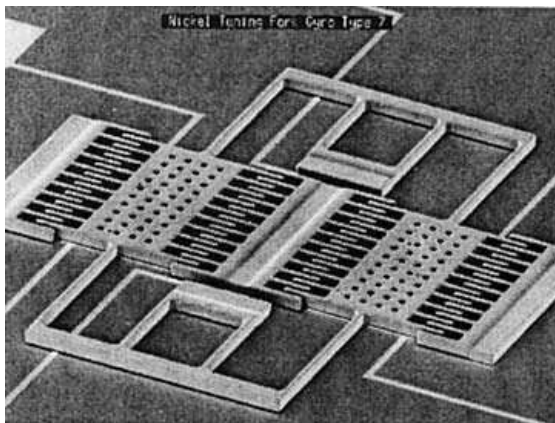
Το γυροσκόπιο είναι μια ηλεκτρομηχανική κατασκευή που έχει την ικανότητα να μετράει την γωνιακή ταχύτητα ενός αντικειμένου ως εκ τούτου μπορεί να αντιλαμβάνεται περιστροφική κίνηση και αλλαγές στον προσανατολισμό. Η πιο συνηθισμένη εφαρμογή που βρίσκει το συγκεκριμένο αισθητήριο είναι για την μέτρηση μετατόπισης της γωνίας ενός αντικειμένου απλά συσσωρεύοντας συνεχώς τις τιμές εξόδου, σε σταθερό χρόνο, από έναν άξονα, γνωρίζοντας όμως την αρχική θέση του αντικειμένου. Για αυτό τον λόγο βρίσκει εφαρμογές στις μεταφορές, στην αεροπολία ακόμα και στην οικιακή τεχνολογία. Το πιο μεγάλο πλεονέκτημα που έχει ο συγκεκριμένος τύπος αισθητηρίου είναι ότι δεν επηρεάζεται από παράγοντες όπως η βαρύτητα ή το μαγνητικό πεδίο της Γης. Παρόλα αυτά όμως εκεί που υστερεί είναι στα σφάλματα που υπεισέρχονται στις μετρήσεις κατά την διάρκεια της συσσώρευσης (τα σφάλματα από τις μετρήσεις συσσωρεύονται έως ότου το σφάλμα γίνει αισθητό). Για το λόγο αυτό συνήθως μαζί με το γυροσκόπιο χρησιμοποιείται και ένα επιταχυνσιόμετρο προκειμένου να διορθώνονται όποια σφάλματα και αποκλίσεις υπάρχουν.

Όσον αφορά την λειτουργία τους υπάρχουν πολλοί τύποι γυροσκοπίων, όλοι όμως χρησιμοποιούν το φαινόμενο κοριόλις σαν αρχή λειτουργίας τους, οι πιο γνωστοί είναι:

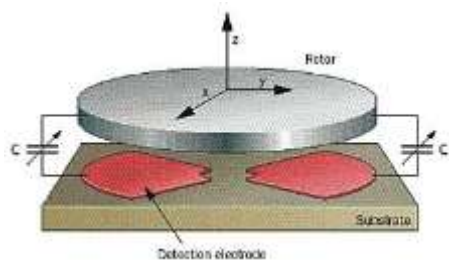
- Tuning Fork Gyroscopes (EIKONA 14): περιέχουν ένα ζευγάρι μαζών τα οποία ταλαντώνονται με το ίδιο πλάτος αλλά με διαφορετικές κατευθύνσεις. Όταν υπάρξει περιστροφή το φαινόμενο κοριόλις δημιουργεί μια κάθετη δόνηση η οποία μπορεί να μετρηθεί με την χρήση μιας ποικιλίας μηχανισμών.
- Vibrating Wheel Gyroscopes (EIKONA 15, 16): περιέχουν έναν τροχό ο οποίος περιστρέφεται γύρω από το άξονα συμμετρίας του, οποιαδήποτε περιστροφή αναγκάζει τον τροχό να πάρει κλίση. Αυτή η αλλαγή της κλίσης μπορεί να εντοπιστεί με την χρήση χωρητικών ηλεκτροδίων που βρίσκονται κάτω από τον τροχό.

Όπως τα επιταχυνσιόμετρα έτσι και τα γυροσκόπια προσδιορίζονται από κάποια λειτουργικά χαρακτηριστικά που ως επί το πλείστον είναι ίδια.

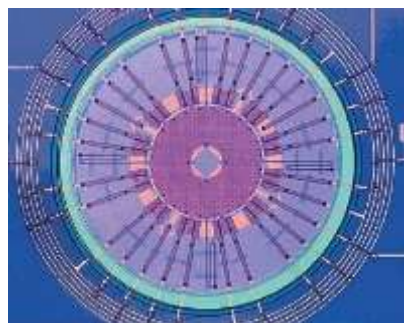
- **Αναλογικό/Ψηφιακό:** ορίζει το είδος της διεπαφής που έχει το αισθητήριο για σύνδεση με κάποιο επεξεργαστή.
- **Αριθμός αξόνων:** είναι ο αριθμός των διαστάσεων που μπορεί να μετρήσει την γωνιακή ταχύτητα το αισθητήριο.
- **Εύρος μετρήσεων:** είναι η μέγιστη τιμή μέτρησης που μπορεί να επιστρέψει το αισθητήριο. Στα συγκεκριμένα αισθητήρια οι τιμές που επιστρέφονται είναι μοίρες ανά δευτερόλεπτο για παράδειγμα $\pm 100^\circ/s$, $\pm 500^\circ/s$.
- **Ευαισθησία:** αφορά την ακρίβεια των μετρήσεων που μπορεί να παρέχει το αισθητήριο.
- **Παροχή ρεύματος:** απαραίτητη ένταση ρεύματος για την λειτουργία του αισθητηρίου.
- **Θερμοκρασία λειτουργίας:** όρια θερμοκρασίας για την σωστή λειτουργία του αισθητηρίου.



Εικόνα 144: Φωτογραφία από μικροσκόπιο γυροσκοπίου MEMS τύπου Tuning Fork



Εικόνα 1545: Δομή γυροσκοπίου τύπου Vibrating Wheel



Εικόνα 1646: Φωτογραφία από μικροσκόπιο γυροσκοπίου MEMS τύπου Vibrating Wheel

3.1.3 Μαγνητόμετρο (Magnetic Field Sensor)

Το μαγνητόμετρο, όπως και τα προηγούμενα αισθητήρια που αναφέρθηκαν, είναι μια ηλεκτρομηχανική συσκευή (MEMS) που έχει την ικανότητα να ανιχνεύεται μαγνητικά πεδία. Υπάρχουν δύο βασικοί τύποι μαγνητομέτρων:

- Scalar (βαθμωτά): χρησιμοποιούνται για να μετρήσουν την συνολική ισχύ του μαγνητικού πεδίου.
- Vector (διανυσματικά): χρησιμοποιούνται για την μέτρηση της πυκνότητας ροής του μαγνητικού πεδίου σε έναν συγκεκριμένο άξονα (x, y, z).

Τα μαγνητικά πεδία μπορεί να είναι είτε το μαγνητικό πεδίο της γης είτε το μαγνητικό πεδίο που παράγεται από μία συσκευή (για παράδειγμα έναν κινητήρα) ή ακόμα και από ρεύμα που διαρρέει έναν αγωγό. Οι τομείς που έχει εφαρμογή μια τέτοια συσκευή είναι, σε οποιοδήποτε είδους σύστημα πλοήγησης, στην ανακάλυψη κοιτασμάτων μεταλλευμάτων και με την χρήση της τεχνολογίας MEMS πλέον τέτοιες συσκευές χρησιμοποιούνται και σε οικιακές συσκευές. Συνήθως χρησιμοποιούνται σε συνδυασμό με τα προαναφερθέντα αισθητήρια προκειμένου να απαλειφθούν τυχόν σφάλματα που υπεισέρχονται στις μετρήσεις λόγω χαμηλής ακρίβειας του συγκεκριμένου αισθητηρίου σε γρήγορες κινήσεις.

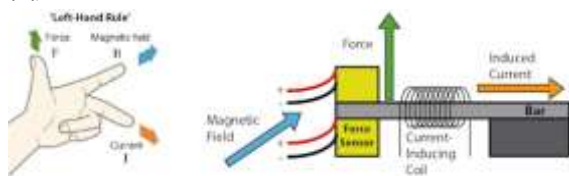
Για παράδειγμα ένας συνδυασμός του συγκεκριμένου αισθητηρίου με ένα γυροσκόπιο μπορεί να επιστρέψει την θέση ενός αντικείμενου με μεγάλη ακρίβεια. Όπως αναφέρθηκε και πιο πριν με την χρήση γυροσκοπίου και συσσώρευση των μετρήσεων σε σταθερό χρόνο μπορούμε να υπολογίσουμε θέση εφόσον γνωρίζουμε την αρχική θέση ενός αντικείμενου. Το πρόβλημα σε ένα τέτοιο σενάριο είναι η συσσώρευση των σφαλμάτων του αισθητηρίου (σε σχετικά μεγάλο χρονικό διάστημα το άθροισμα των σφαλμάτων γίνεται αισθητό από τον χρήστη). Με την χρήση ενός μαγνητόμετρου στο παρόν σενάριο η συσκευή θα μπορούσε να επαναυπολογίζει την θέση της λαμβάνοντας υπ' όψιν μόνο μετρήσεις από το μαγνητόμετρο. Έπειτα θα συνέχιζε τον υπολογισμό θέσης με μετρήσεις από γυροσκόπιο έως ότου θεωρήσει ότι το σφάλμα είναι αρκετά μεγάλο και χρειάζεται επαναπροσδιορισμός της θέσης.

Σε ένα ακόμα παράδειγμα το αισθητήριο θα μπορούσε να επιστρέψει μεγάλη ακρίβεια θέσης σε συνδυασμό με κάποιο επιταχυνσιόμετρο. Στο σενάριο αυτό το επιταχυνσιόμετρο αναλαμβάνει την εξομάλυνση της αρχικής γωνίας που μπορεί να έχει το αντικείμενο, εφόσον κατά πάσα πιθανότητα δεν θα είναι παράλληλα με το έδαφος.

Ανάλογα με το είδος του μαγνητομέτρου (scalar, vector) υπάρχουν πολλοί τύποι τέτοιων αισθητηρίων βασισμένοι σε διαφορετικές αρχές λειτουργίας. Ένας από τους πιο κοινούς τύπους μαγνητόμετρου βασίζονται στην αρχή της δύναμης Lorentz (EIKONA 17). Στο εσωτερικό του ολοκληρωμένου υπάρχουν δοκοί, ένας για κάθε άξονα, γύρω από τους δοκούς υπάρχει τυλιγμένο ένα πηνίο το οποίο διαρρέεται από ρεύμα και οι άκρες των δοκών εφάπτονται σε αισθητήρια πίεσης. Λόγω της δύναμης Lorentz όσο πιο δυνατό είναι το μαγνητικό πεδίο μέσα στο οποίο βρίσκεται η συσκευή τόσο παραμορφώνεται και η δοκός. Τα αισθητήρια πίεσης «μεταφράζουν» αυτήν την παραμόρφωση σε μέτρηση ισχύος μαγνητικού πεδίου.

Όπως και τα παραπάνω αισθητήρια έτσι και τα μαγνητόμετρα προσδιορίζονται από παρόμοια λειτουργικά χαρακτηριστικά.

- **Ακρίβεια:** ορίζει την ακρίβεια μέτρησης που μπορεί να επιστρέψει το αισθητήριο.
- **Ανάλυση:** ορίζει την μικρότερη μονάδα που μπορεί να μετρήσει το αισθητήριο.
- **Συχνότητα μέτρησης:** ορίζει το πόσο συχνά μπορεί το αισθητήριο να επιστρέφει αξιόπιστες μετρήσεις.
- **Μέτρηση πυκνότητας ροής:** ορίζει το εύρος των μετρήσεων που είναι ικανή να μετρήσει η συσκευή.
- **Θερμοκρασία λειτουργίας:** ορίζει το εύρος θερμοκρασίας για την κανονική λειτουργία του αισθητηρίου.



Εικόνα 1747: αρχή λειτουργίας μαγνητόμετρου βασισμένο στην αρχή της δύναμης Lorentz

3.2 Αισθητήριο Αδράνειας GY-88

Για την υλοποίηση της εφαρμογής θεωρήθηκε σωστό να γίνει χρήση ενός επιταχυνσιόμετρου προκειμένου να παίρνουμε μετρήσεις από το χέρι του χρήστη κατά την διάρκεια μίας γρήγορης χειρονομίας. Όσον αφορά την κίνηση του κέρσορα χρησιμοποιήθηκε ένα γυροσκόπιο γνωρίζοντας ότι είναι ένα αισθητήριο που δεν επηρεάζεται από άλλους εξωτερικούς παράγοντες, όπως η βαρύτητα και το μαγνητικό πεδίο της γής, παρά μόνο από την κίνηση του χρήστη.

Η πλακέτα IMU GY-88 (ΕΙΚΟΝΑ 18) θεωρήθηκε ιδανική για την εργασία εφόσον περιέχει, πέρα από το σύνολο των αισθητηρίων που αρχικά θεωρήσαμε ως απαραίτητα, δύο επί πλέον αισθητήρια που ενδεχομένως να βελτιώσουν μετέπειτα εκδόσεις της συγκεκριμένης εργασίας. Πέρα από τα επιπλέον αισθητήρια το ολοκληρωμένο που περιέχει το επιταχυνσιόμετρο και το γυροσκόπιο προσφέρει και κάποια επιπλέον χαρακτηριστικά που βοήθησαν στην ευκολότερη υλοποίηση του έργου, καθώς με την χρήση τους περιορίστηκε η συγγραφή επιπλέον κώδικα αλλά και η κατανάλωση ενέργειας. Τα χαρακτηριστικά αυτά αναφέρονται σε επόμενη παράγραφο κατά την διάρκεια ανάλυσης του επιμέρους τμήματος της πλακέτας που αφορά την ισχύουσα υλοποίηση.

Το IMU GY-88 περιλαμβάνει τα ακόλουθα ολοκληρωμένα:

- **MPU6050:** επιταχυνσιόμετρο και γυροσκόπιο τριών αξόνων καθώς και ένας επεξεργαστής κίνησης.
- **HMC5883L:** μαγνητόμετρο τριών αξόνων.
- **BMP085:** αισθητήριο μέτρησης ατμοσφαιρικής πίεσης.

Όλα τα παραπάνω αισθητήρια είναι ψηφιακά και η πλακέτα παρέχει I²C διεπαφή για την επικοινωνία των αισθητηρίων. Οι ακροδέκτες που είναι διαθέσιμοι είναι:

- **VCC_IN:** είσοδος τροφοδοσίας.
- **3.3V:** έξοδος για τροφοδοσία άλλου IC.
- **GND:** γείωση.
- **SCL:** ακροδέκτης ρολογιού για I²C διεπαφή.
- **SDA:** ακροδέκτης δεδομένων για I²C διεπαφή.
- **M_DRDY:** ακροδέκτης διακοπής για το μαγνητόμετρο.
- **G_AD0:** ακροδέκτης για αλλαγή της διεύθυνσης του ολοκληρωμένου MPU6050.
- **G_INT:** ακροδέκτης διακοπής για το επιταχυνσιόμετρο.

Όλα τα αισθητήρια που ενσωματώνει η πλακέτα είναι συνδεδεμένα με το MPU6050. Το ολοκληρωμένο αυτό διαθέτει έναν βοηθητικό μηχανισμό I²C, που επιτρέπει την σύνδεση τρίτων αισθητηρίων. Αυτό το χαρακτηριστικό καθιστά το αισθητήριο ικανό να λειτουργεί είτε σαν master του διαύλου, είτε σαν «μεσάζοντα», όπου απλά παρέχει πρόσβαση στο συνδεδεμένο αισθητήριο χωρίς να παρεμβαίνει καθόλου μεταξύ του τρίτου αισθητηρίου και του



Εικόνα 18: πλακέτα GY-88.

μικροεπεξεργαστή, ο έλεγχος για το είδος της λειτουργίας του ολοκληρωμένου γίνεται από τους registers. Η τάση λειτουργίας της πλακέτας είναι από 3 έως 5 Volt.

3.3 Αισθητήριο MPU-6050

Όπως αναφέρθηκε και προηγουμένως από τα αισθητήρια που παρέχονται στην πλακέτα GY-88 χρησιμοποιήθηκε μόνο το αισθητήριο MPU6050 που περιλαμβάνει το γυροσκόπιο και το επιταχυνσιόμετρο καθώς και κάποια επιπρόσθετα χαρακτηριστικά που βοήθησαν στην υλοποίηση του συστήματος. Στην παρούσα παράγραφο παραθέτουμε μια επισκόπηση των χαρακτηριστικών του ολοκληρωμένου MPU6050.

3.3.1 MPU-6050 Χαρακτηριστικά Επιταχυνσιόμετρου

Το επιταχυνσιόμετρο είναι τριών αξόνων και χρησιμοποιεί ξεχωριστές μάζες αναφοράς για κάθε άξονα. Η άσκηση επιτάχυνσης πάνω σε έναν από τους τρεις άξονες θα προκαλέσει μετατόπιση της αντίστοιχης μάζας αναφοράς και στην συνέχεια οι χωρητικοί αισθητήρες θα ανιχνεύσουν την μετατόπιση. Όταν το αισθητήριο είναι πάνω σε επίπεδη επιφάνεια οι τιμές που πρέπει να επιστρέψει είναι 0g για τον x και y άξονα και 1g για τον άξονα z. Παρακάτω παρατίθενται τα χαρακτηριστικά του αισθητηρίου:

- Δυνατότητα προγραμματιστικής επιλογής μέγιστου εύρους μετρήσεων, με τις ακόλουθες επιλογές ±2g, ±4g, ±8g και ±16g.
- Διαθέτει τρεις ανεξάρτητους ενσωματωμένους 16-bit A2D μετατροπείς για ταυτόχρονη δειγματοληψία και από τους τρεις άξονες, από το αναλογικό τμήμα του αισθητηρίου.
- Κατανάλωση κανονικής λειτουργίας 500μΑ.
- Δυνατότητα λειτουργίας χαμηλής κατανάλωσης στις αντίστοιχες συχνότητες δειγματοληψίας:
 - 10μΑ στα 1.25Hz
 - 20μΑ στα 5Hz
 - 60μΑ στα 20Hz
 - 110μΑ στα 40Hz
- Λειτουργία αυτό-διάγνωσης.

3.3.2 MPU-6050 Χαρακτηριστικά Γυροσκοπίου

Το γυροσκόπιο είναι τριών αξόνων και αποτελείται από τρία ανεξάρτητα γυροσκόπια τύπου Vibrating Wheel τα οποία μπορούν να ανιχνεύσουν περιστροφή στους άξονες x, y και z. Όταν τα γυροσκόπια τεθούν υπό περιστροφή γύρω από οποιοδήποτε άξονα το φαινόμενο Κορίολις δημιουργεί δονήσεις οι οποίες ανιχνεύονται από χωρητικούς αισθητήρες. Το ανιχνευόμενο σήμα ενισχύεται, αποδιαμορφώνεται και φιλτράρεται προκειμένου να παραχθεί τάση ανάλογη του ρυθμού περιστροφής. Αυτή η τάση ψηφιοποιείται με την χρήση ενσωματωμένων ανεξάρτητων 16-bit A2D μετατροπέων για κάθε άξονα. Το εύρος του αισθητηρίου μπορεί να ρυθμιστεί προγραμματιστικά από τον χρήστη, όπως και η συχνότητα δειγματοληψίας καθώς και το χαμηλοπερατό φίλτρο για αποκοπή υψηλών συχνοτήτων που διατίθεται. Παρακάτω παρατίθενται τα χαρακτηριστικά του αισθητηρίου:

- Δυνατότητα προγραμματιστικής επιλογής μέγιστου εύρους μετρήσεων, με τις ακόλουθες επιλογές ± 250 , ± 500 , ± 1000 , ± 2000 %s.
- Δυνατότητα εισαγωγής εξωτερικού σήματος για συγχρονισμό με άλλα αισθητήρια.
- Δυνατότητα προγραμματιστικής ρύθμισης του χαμηλοπερατού φίλτρου.
- Κατανάλωση κανονικής λειτουργία 3.6mA.
- Κατανάλωση σε κατάσταση αναμονής 5μΑ.
- Λειτουργία αυτό-διάγνωσης.

3.3.3 MPU-6050 Sensor Fusion Engine

Το ολοκληρωμένο διαθέτει ενσωματωμένο έναν ψηφιακό επεξεργαστή κίνησης (Digital Motion Processor). Κύριος σκοπός του επεξεργαστή είναι αποφορτίζει τον κεντρικό επεξεργαστή από κοστοβόρες επεξεργασίες αλγορίθμων δεδομένων κίνησης. Ο DMP ανακτά δεδομένα από τα ενσωματωμένα αισθητήρια αλλά και από τρίτα αισθητήρια (μαγνητόμετρο) και κάνει την επεξεργασία. Τα επεξεργασμένα δεδομένα μπορούν να ανακτηθούν είτε απευθείας από τους registers του DMP είτε από την ενσωματωμένη ουρά του ολοκληρωμένου.

Σε ένα κοινό σενάριο ένας αλγόριθμος κίνησης πρέπει να εκτελείται σε μεγάλη συχνότητα προκειμένου να προσφέρει ακριβή αποτελέσματα με χαμηλή καθυστέρηση. Έτσι ο DMP μπορεί να χρησιμοποιηθεί προκειμένου να χαμηλώσει την κατανάλωση του συστήματος, να απλοποιήσει τις ανάγκες χρονισμού και την αρχιτεκτονική του λογισμικού και βέβαια να μειώσει την επεξεργαστική απαίτηση που έχει το σύστημα από τον κεντρικό επεξεργαστή. Παρακάτω παρατίθενται κάποια χαρακτηριστικά του DMP:

- Υποστηρίζει μια μεγάλη γκάμα αλγορίθμων επεξεργασίας κίνησης και αναγνώρισης χειρονομιών.
- Υποστηρίζει την συλλογή δεδομένων από γυροσκόπιο, επιταχυνσιόμετρο και θερμομέτρο καθώς και από τρίτα αισθητήρια όπως μαγνητόμετρο.
- Διαθέτει προγραμματιζόμενες διακοπές συσχετιζόμενες με τους αλγορίθμους κίνησης.

3.3.4 MPU-6050 Επιπλέον χαρακτηριστικά

- **Διεπαφές (Interfaces):** η συγκεκριμένη έκδοση του ολοκληρωμένου που χρησιμοποιήθηκε υποστηρίζει μόνο I²C διεπαφή. Το chip λειτουργεί σαν slave του bus όταν επικοινωνεί με τον κεντρικό επεξεργαστή. Υπάρχει η δυνατότητα σύνδεσης μέχρι και δύο ίδιων chip στο ίδιο bus εφόσον το LSB της διεύθυνσης του slave chip ελέγχεται από την κατάσταση του pin ADO. Πέρα από την κύρια I²C διεπαφή το ολοκληρωμένο διαθέτει μια επιπλέον βοηθητική είσοδο για επικοινωνία με άλλα αισθητήρια όπως μαγνητόμετρα κ.α. Αυτή η δευτερεύουσα διεπαφή έχει δύο διαφορετικές καταστάσεις, μπορεί να λειτουργήσει είτε σε **Master mode** όπου και αναλαμβάνει τον αντίστοιχο ρόλο σε αισθητήρια που είναι συνδεδεμένα επάνω της, είτε σε **Pass-Through mode** όπου επιτρέπει στον κύριο επεξεργαστή, σαν master του bus να επικοινωνήσει απευθείας με το τρίτο αισθητήριο.
- **Clocking:** Το ολοκληρωμένο προσφέρει διάφορες επιλογές όσον αφορά τον χρονισμό του με μια πλειάδα από εσωτερικές εναλλακτικές πηγές αλλά και δυνατότητα σύνδεσης εξωτερικών πηγών χρονισμού που οδηγούν τα εσωτερικά κυκλώματα του

ολοκληρωμένου. Τα εσωτερικά κυκλώματα που αναφέρουμε περιλαμβάνουν τους A2D μετατροπείς, την DMP μονάδα και διάφορα κυκλώματα και καταχωρητές.

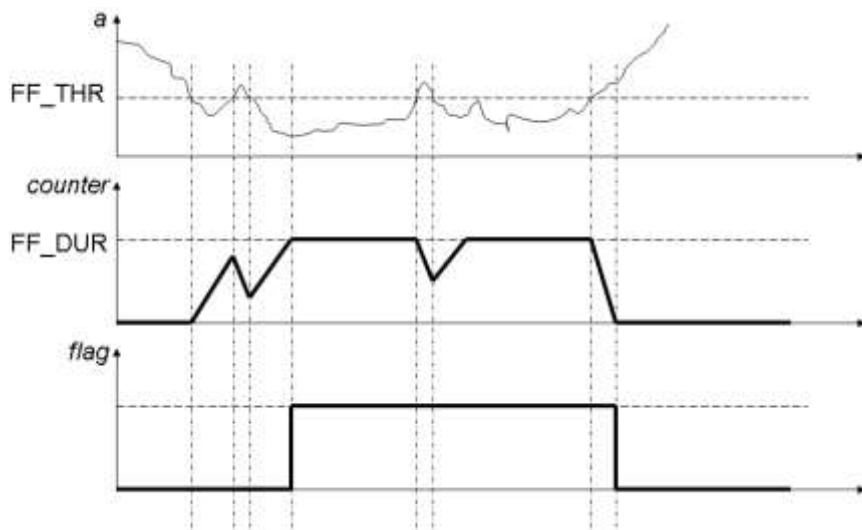
- Τα εσωτερικά κυκλώματα χρονισμού που προσφέρονται είναι:
 - Ένας εσωτερικός ταλαντωτής συχνότητας 8MHz.
 - Οποιοσδήποτε από τους τρεις ταλαντωτές (x, y, z) που χρησιμοποιούνται από το γυροσκόπιο με συχνότητα 8kHz. Με την ρύθμιση ενός 8-bit register (SMPLRT_DIV) ο χρήστης μπορεί να ελέγξει την συχνότητα δειγματοληψίας των αισθητηρίων. Επίσης αν το χαμηλοπερατό φίτρο είναι ενεργοποιημένο τότε η παραγόμενη συχνότητα από τους τρεις ταλαντωτές είναι 1kHz.
- Οι εξωτερικές επιτρεπόμενες πηγές είναι:
 - Ένας ακροδέκτης που δέχεται τετραγωνικό παλμό συχνότητας 32.768kHz.
 - Ένας ακροδέκτης που δέχεται τετραγωνικό παλμό συχνότητας 19.2MHz.

Το ρολόι που θα επιλεγεί για το σύστημα εξαρτάται αποκλειστικά από τις απαιτήσεις της εφαρμογής σε ακρίβεια ή σε κατανάλωση ενέργειας. Για παράδειγμα αν η εφαρμογή απαιτεί μικρή κατανάλωση, χωρίς την χρήση του γυροσκοπίου, αλλά ενεργοποιεί μόνο το DMP για την επεξεργασία δεδομένων από το επιταχυνσιόμετρο, τότε καλή επιλογή χρονισμού είναι, είτε οι εξωτερικές πηγές, είτε ο εσωτερικός ταλαντωτής. Σε διαφορετική περίπτωση που απαιτείται η χρήση γυροσκοπίου τότε κατάλληλη επιλογή είναι οι ταλαντωτές του γυροσκοπίου οι οποίοι προσφέρουν και πιο ακριβή χρονισμό.

- **FIFO:** Το MPU6050 περιλαμβάνει μια FIFO ουρά μεγέθους 1024 bytes που είναι προσβάσιμη μέσω της διεπαφής I²C. Ο χρήστης μπορεί να επιλέξει τα δεδομένα που εισέρχονται στην ουρά (δεδομένα γυροσκοπίου, θερμοκρασίας, επιτάχυνσης, ακόμα και δεδομένα από αισθητήρια συνδεδεμένα στον βοηθητικό δίαυλο I²C). Ένας μετρητής αναφέρει πόσα δεδομένα βρίσκονται κάθε στιγμή στην ουρά και ακόμα ο χρήστης μπορεί να γνωρίζει πότε έχει εισαχθεί ένα νέο δεδομένο καθώς και αν έχει γίνει υπερχείλιση με χρήση ρυθμιζόμενων διακοπών.
- **Temperature Sensor:** διατίθεται ένας αισθητήρας θερμοκρασίας με ένα επιπλέον A2D μετατροπέα. Οι τιμές θερμοκρασίας μπορούν να ανακτηθούν είτε από τους αντίστοιχους registers είτε από την FIFO.
- **Διακοπές (Interrupts):** Πέρα από τις διακοπές που μπορεί να έχει ένα απλό αισθητήριο (Data Ready ή FIFO Overflow) το chip προσφέρει ένα σύνολο διακοπών οι οποίες σχετίζονται με κίνηση και είναι πλήρως προγραμματιζόμενες από τον χρήστη.
 - **Free-fall interrupt:** η διακοπή ενεργοποιείται όταν και οι τρεις άξονες του γυροσκοπίου είναι κάτω από ένα κατώφλι (κατώφλι επιτάχυνσης), που είναι ρυθμιζόμενο από τον χρήστη. Πιο συγκεκριμένα για κάθε δείγμα που είναι κάτω από το συγκεκριμένο κατώφλι (κατώφλι επιτάχυνσης) ένας μετρητής αυξάνεται, αν για το παρόν δείγμα δεν ισχύει η παραπάνω συνθήκη τότε ο μετρητής μειώνεται. Όταν ο μετρητής φτάσει ένα κατώφλι (χρονικό κατώφλι, ρυθμιζόμενο από τον χρήστη), ενεργοποιείται η διακοπή. Όπως ειπώθηκε και τα δύο κατώφλια χρονικό και επιτάχυνσης μπορούν να ρυθμιστούν από τον χρήστη με

ακρίβεια 1ms και 1mg αντίστοιχα. Επίσης και ο ρυθμός μείωσης του χρονικού κατώφλιου είναι ρυθμιζόμενος.

- **Zero Motion interrupt:** η διακοπή ενεργοποιείται και όταν ανιχνευθεί πρώτη φορά ακινησία και όταν ανιχνευθεί κίνηση μετά από ακινησία. Η λειτουργία του συστήματος είναι παρόμοια με την free-fall διακοπή και έχει τα αντίστοιχα ρυθμιζόμενα κατώφλια.
- **Motion Interrupt:** με τον ίδιο τρόπο, όπως η free-fall διακοπή, ενεργοποιείται και η ανίχνευση κίνησης με την διαφορά ότι οι τιμές που παρέχονται από το επιταχυνσιόμετρο περνάνε πρώτα από ένα υπεραπλοποιημένο φίλτρο προκειμένου να εξαλειφθεί η τιμή της βαρύτητας από τις μετρήσεις. Στην συνέχεια για κάθε δείγμα άνω του κατώφλιου ο μετρητής αυξάνεται και αντίστοιχα μειώνεται για κάθε δείγμα που δεν ικανοποιεί την συνθήκη. Η διακοπή ενεργοποιείται όταν η τιμή του μετρητή ξεπεράσει το ορισμένο κατώφλι. Επιπλέον ο άξονας και η πολικότητα που ενεργοποίησαν την διακοπή μπορούν να ανακτηθούν μέσα από συγκεκριμένο register.



Εικόνα 1919: Απλοποιημένο παράδειγμα σε έναν άξονα για την ενεργοποίηση free-fall interrupt (FF_THR: κατώφλι επιτάχυνσης, FF_DUR: χρονικό κατώφλι, flag: ενεργοποίηση διακοπής).

Κεφάλαιο 4 – Ολοκληρωμένο DA14580 και Development Kit

Το chip DA14580 είναι η ολοκληρωμένη λύση της εταιρίας Dialog Semiconductor για εφαρμογές που βασίζονται στην Bluetooth Smart τεχνολογία. Πρόκειται για ένα ολοκληρωμένο κύκλωμα (ράδιο-πομποδέκτης και επεξεργαστής) με την χαμηλότερη κατανάλωση ενέργειας, σε σχέση με τις υπόλοιπες λύσεις που κυκλοφορούν αυτήν την στιγμή στην αγορά. Η πολύ χαμηλή κατανάλωση ενέργειας (λιγότερο από το μισό σε σχέση με λύσεις άλλων εταιριών) οφείλεται στην προσέγγιση της εταιρίας Dialog, να μην συμπεριλάβει μνήμη flash στο ολοκληρωμένο. Με αυτόν τον τρόπο μπόρεσαν και μείωσαν το κόστος του ολοκληρωμένου αλλά και την κατανάλωση, εφόσον η τοποθέτηση μνήμης flash απαιτεί εξειδικευμένο κύκλωμα καθώς και υψηλότερη τάση λειτουργίας. Στον ΠΙΝΑΚΑ 12 παρατίθενται οι λύσεις των πιο γνωστών εταιριών για Bluetooth Smart τεχνολογία με μερικά χαρακτηριστικά τους.

	Mode	Integrated Processor	Flash	RAM	Current Consumption (RX/TX)
TI CC2540/CC2541	Single Mode v4.0	8051	128kB/256kB	8kB	17.9mA / 18.2mA to 14.7mA / 14.3mA
Texas Instruments CC256x	Dual Mode Classic + BLE/ANT	No - External	None	None	-
Texas Instruments CC26xx	Single Mode BLE v4.1	Cortex-M4F	128kB	20kB	5.9mA
Nordic Semiconductor nRF51822	Single Mode v4.1 / ANT	Cortex-M0	128kB/256kB	16kB / 32kB	9.7mA / 8mA
Nordic Semiconductor nRF8001	Single Mode v4.0	None	None	None	14.6mA/12.7mA
Dialog Semiconductor DA14580	Single Mode BLE v4.1	Cortex-M0	32kB OTP	42kB + 8kB	4.9mA / 4.9mA
Cypress Semiconductor PSoC 4 BLE / PSoC BLE	Single Mode BLE v4.1	Cortex-M0	128kB	16kB	15.6mA / 16.4mA
CSR CSR101x	Single Mode BLE v4.1	16-bit RISC	64kB	64kB	16mA

Πίνακας 12-2: Περίληψη χαρακτηριστικών των πιο γνωστών εταιριών για λύσεις Bluetooth Smart [5].

4.1 DA14580 chip

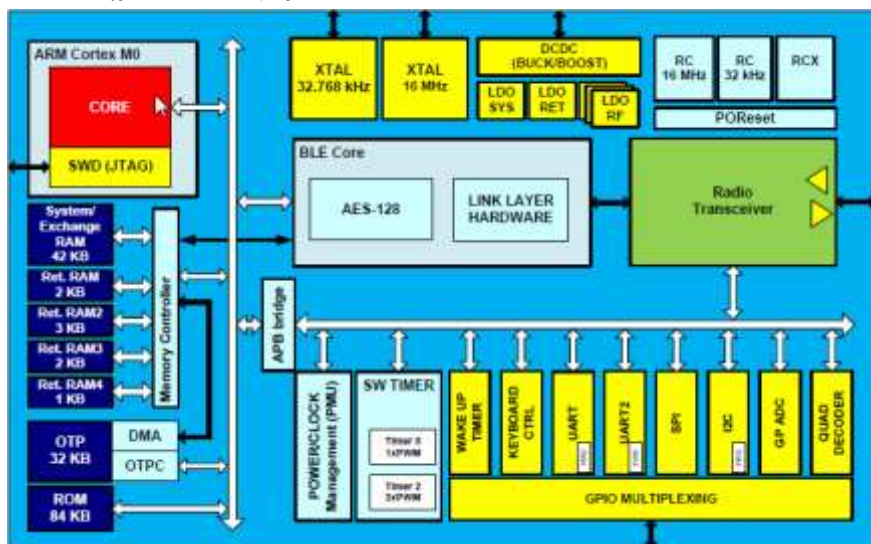
4.1.1 Γενική Περιγραφή

Το ολοκληρωμένο κύκλωμα DA14580 διαθέτει ενσωματωμένο έναν ράδιο – πομποδέκτη καθώς και έναν επεξεργαστή που χρησιμοποιείται για την διαχείριση όλων των διαδικασιών που περιλαμβάνουν την ασύρματη επικοινωνία Bluetooth. Ο επεξεργαστής, επίσης, μπορεί να χρησιμοποιηθεί και για την εκτέλεση εφαρμογών αλλά και για να παρέχει δεδομένα σε fully Hosted συστήματα (συστήματα όπου η εφαρμογή του χρήστη εκτελείται σε άλλον εξωτερικό επεξεργαστή). Το DA14580 διαθέτει μια ενσωματωμένη OTP μνήμη για την αποθήκευση των

Ανάπτυξη συσκευής εισόδου με την χρήση τεχνολογίας Bluetooth Low Energy και αισθητηρίων αδράνειας

Bluetooth προφίλ αλλά και για κώδικα εφαρμογής από τον χρήστη. Η στοίβα του Bluetooth πρωτοκόλλου βρίσκεται σε μια μνήμη ROM και μαζί με την εφαρμογή του χρήστη (αν υπάρχει) χρησιμοποιούν έναν επεξεργαστή ARM Cortex M0 για την εκτέλεση τους. Επίσης διατίθενται χαμηλής διαροής retention RAM για την αποθήκευση σημαντικών δεδομένων καθώς και πληροφοριών σύνδεσης όσο το σύστημα βρίσκεται σε κατάσταση Deep Sleep.

Το firmware που υλοποιεί την στοίβα του Bluetooth περιλαμβάνει όλα τα απαραίτητα τμήματα που περιγράφηκαν στο κεφάλαιο 1 για την λειτουργία του πρωτοκόλλου (L2CAP – Service Layer Protocols, SM – Security Manager, ATT – Attribute Protocol, GATT – Generic Attribute Profile, GAP – Generic Access Profile) καθώς και τα διάφορα προφίλ όπως αυτό της εγγύτητας, της θερμοκρασίας κ.α. Τέλος διαθέτει αποκλειστικό υλικό για υλοποίηση του Link Layer επιπέδου της Bluetooth στοίβας.



Εικόνα 2020: Block διάγραμμα των τμημάτων που αποτελούν το DA14580..

4.1.2 Επεξεργαστής ARM CORTEX M0

Ο επεξεργαστής ARM CORTEX M0 είναι ένας 32-bit RISC επεξεργαστής με Von Neumann αρχιτεκτονική. Χρησιμοποιεί ένα σύνολο εντολών που καλείται Thumb, στο οποίο έχουν προστεθεί κάποιες επιπλέον εντολές από αρχιτεκτονικές ARMv6 καθώς και μερικές ακόμα από τεχνολογία Thumb-2. Η τεχνολογία Thumb-2 επεκτείνει το σύνολο εντολών thumb προκειμένου να εκτελούνται όλες εντολές σε μία και μόνο κατάσταση του επεξεργαστή. Το σύνολο εντολών Thumb-2 περιλαμβάνει 16-bit και 32-bit εντολές. Ως επί το πλείστον, οι εντολές που παράγονται από έναν C compiler χρησιμοποιούν τις 16-bit εντολές, ενώ οι 32-bit εντολές χρησιμοποιούνται όταν δεν μπορεί να εκτελεστεί την απαραίτητη διαδικασία από τις 16-bit. Αυτό προσφέρει ομοιομορφία στον παραγόμενο κώδικα και συμβάλει στην αποφυγή της επιβάρυνσης που εισάγει η αλλαγή από, και προς, δύο διαφορετικών συνόλων εντολών. Συνολικά ο επεξεργαστής υποστηρίζει 56 βασικές εντολές. Παρόλο που το σύνολο των εντολών είναι μικρό ο

επεξεργαστής είναι εξαιρετικά ικανός λόγω του ότι το σύνολο εντολών thumb είναι πλήρως βελτιστοποιημένο. Μερικά από τα χαρακτηριστικά του επεξεργαστή είναι:

- Ενσωματωμένος ελεγκτής εμφωλευμένων διακοπών (Nested Vector Interrupt Controller).
- Προγραμματιζόμενη προτεραιότητα διακοπών σε τέσσερα διαφορετικά επίπεδα.
- Αρχιτεκτονικά ορισμένες καταστάσεις «ύπνου» (sleep modes) και εντολές για εισαγωγή του επεξεργαστή σε κατάσταση χαμηλής κατανάλωσης.
- Υποστήριξη 24 συνολικά διακοπών.
- Ελεγκτής Wake-up, προκειμένου όταν ο επεξεργαστής είναι σε κατάσταση χαμηλής κατανάλωσης να μπορεί να μπαίνει σε κατάσταση κανονικής λειτουργίας όταν δέχεται κάποια διακοπή.

4.1.3 Bluetooth Smart Τμήματα

- **Πυρίνας BLE (BLE Core):** Bluetooth 4.0 ελεγκτής ραδιοσυχνότητας πλήρως συμβατός με το πρότυπο Bluetooth Smart και επιφορτισμένος με διαδικασίες δρομολόγησης, διαμόρφωσης και αποδιαμόρφωσης πακέτων.
- **Ράδιο – Πομποδέκτης:** περιλαμβάνει το RF τμήμα του πρωτοκόλλου Bluetooth Smart.

4.1.4 Μνήμες

Οι ακόλουθες μνήμες είναι τμήματα του ολοκληρωμένου DA14580.

- **ROM:** είναι μια 84 kB μνήμη που περιέχει την στοίβα του Bluetooth πρωτοκόλλου καθώς και την διαδικασία εκκίνησης.
- **OTP:** είναι μια 32 kB One Time Programmable μνήμη που χρησιμοποιείται για την αποθήκευση του κώδικα της εφαρμογής καθώς και για τα διάφορα profile που υποστηρίζει η συσκευή.
- **System Ram:** μια 42 kB system SRAM (SYS RAM). Κάθε φορά που το σύστημα ξεκινά, σε αυτήν την μνήμη αντιγράφονται τα περιεχόμενα της OTP, ακόμη χρησιμοποιείται και σαν Data RAM για τυχόν μεταβλητές ή δεδομένα που απαιτεί το πρωτόκολλο.
- **Retention Ram:** είναι τέσσερις ξεχωριστές, χαμηλής διαρροής, SRAM μνήμες (2kB, 2kB, 3kB, 1kB) που χρησιμοποιούνται για την αποθήκευση διαφόρων δεδομένων όταν το σύστημα μπαίνει σε κατάσταση Deep Sleep. Κάθε μνήμη μπορεί να ενεργοποιηθεί και να απενεργοποιηθεί ξεχωριστά σύμφωνα με τις ανάγκες της εφαρμογής.

4.1.5 Λειτουργικές Καταστάσεις

Το σύστημα έχει τρεις διαφορετικές καταστάσεις λειτουργίας όσον αφορά την ανάπτυξη και την κανονική του λειτουργία σαν προϊόν.

- **Development Mode:** Σε αυτήν την κατάσταση ο κώδικας της εφαρμογής εγγράφεται στην System RAM του συστήματος ή και στις Retention Ram's, αν χρησιμοποιούνται, με την χρήση είτε του SWD (JTAG) είτε οποιασδήποτε σειριακής διεπαφής. Στην

συνέχεια η CPU κάνει Reset και εκτελεί τον κώδικα. Σε αυτήν την κατάσταση το σύστημα βρίσκεται κατά την διάρκεια ανάπτυξης, εκφαλμάτωσης και ελέγχου.

- **Normal Mode:** Σε αυτήν την κατάσταση ο κώδικας έχει ελεγχθεί και μπορεί να «περάσει» στην OTP μνήμη. Όταν το σύστημα ξεκινήσει ο DMA, αυτόματα, θα αντιγράψει τα περιεχόμενα της OTP στην System RAM και η εκτέλεση του κώδικα θα ξεκινήσει. Μετά από αυτό το στάδιο το σύστημα είναι έτοιμο να ενσωματωθεί στο τελικό προϊόν.
- **Calibration Mode:** Μεταξύ των δύο παραπάνων καταστάσεων υπάρχει μια ενδιάμεση, επιπλέον, κατάσταση. Το ολοκληρωμένο χρειάζεται να κάνει μια βαθμονόμηση βάσει δύο παραμέτρων, της ρύθμιση της διεύθυνσης Bluetooth της συσκευής και βαθμονόμηση του εξωτερικού κρυστάλλου χρονισμού. Αυτή η κατάσταση αντιστοιχεί στην τελική έκδοση του προϊόντος.

4.1.6 Ενεργειακές Καταστάσεις

Το ολοκληρωμένο υποστηρίζει τρεις διαφορετικές καταστάσεις ενέργειας.

- **Active Mode:** το σύστημα είναι ενεργό και λειτουργεί με την μέγιστη κατανάλωση.
- **Sleep Mode:** το σύστημα είναι ενεργό και ο επεξεργαστής βρίσκεται σε κατάσταση αναμονής μέχρι να συμβεί κάποια διακοπή. Ο ραδιοπομπός και τα περιφερειακά είναι σε κατάσταση που ορίζεται από την εκάστοτε τιμή των καταχωριτών ελέγχου τους.
- **Extended Sleep Mode:** όλα τα περιφερειακά του συστήματος και το ράδιο είναι απενεργοποιημένα. Μόνο η Sys RAM και το τμήμα που είναι συνέχεια ενεργό είναι σε κατάσταση λειτουργίας προκειμένου να διατηρεί τα δεδομένα στις retention RAM's και να παρέχει ενέργεια στους χρονιστές που μπορούν να ζυτνήσουν το σύστημα.
- **Deep Sleep Mode:** Είναι παρόμοια με την προηγούμενη κατάσταση με την διαφορά ότι και η Sys RAM είναι απενεργοποιημένη. Σε κατάσταση Deep Sleep επειδή η Sys RAM απενεργοποιείται κάθε φορά του το σύστημα «ξυπνά» απαιτείται η αντιγραφή της OTP στην Sys RAM.

4.1.7 Διεπαφές (Interfaces)

- **Σειριακές διεπαφές:** Το ολοκληρωμένο διαθέτει ένα σύνολο από σειριακές διεπαφές για την επικοινωνία με άλλες συσκευές. Οι διαθέσιμες διεπαφές είναι UART, I²C και SPI.
- **General Purpose ADC:** διατίθεται ένας υψηλής ταχύτητας 10-bit, γενικού σκοπού, A2D μετατροπέα. Το ADC έχει δικό του ρυθμιστή τάσης (LDO) στα 1.2V που αντιπροσωπεύει και την μέγιστη τάση αναφοράς.
- **Quadrature Decoders:** το παρόν τμήμα επιτρέπει στο ολοκληρωμένο να αποκωδικοποιεί παλμοσειρές από έναν περιστροφικό αποκωδικοποιητή (rotary encoder – μετατρέπει περιστροφική κίνηση σε παλμοσειρά) προκειμένου να τις μετατρέψει σε αριθμό βημάτων και κατεύθυνση μιας κίνησης από κάποια εξωτερική συσκευή. Τρεις άξονες υποστηρίζονται (x, y, z). Το παρόν τμήμα μπορεί να θέσει το σύστημα σε κατάσταση λειτουργίας αν αυτό, πριν, έχει πέσει σε κατάσταση χαμηλής λειτουργίας (sleep mode).

- **Keyboard Controller:** αυτό το τμήμα μπορεί να χρησιμοποιηθεί για τον αποκλιδωνισμό (debounce) εισερχομένων σημάτων από τις πόρτες εισόδου/εξόδου. Επίσης για κάθε πάτημα παράγεται διακοπή.

4.1.8 Χρονιστές (Timers)

- **Software Timer:** Το παρόν τμήμα περιέχει 2 χρονιστές (Timer0 και Timer2) που ελέγχονται από λογισμικό, είναι προγραμματιζόμενοι και μπορούν να χρησιμοποιηθούν για διάφορες διεργασίες.
- **Wake-up Timer:** Ο συγκεκριμένος Timer είναι επιφορτισμένος με την λειτουργία της αφύπνισης του ολοκληρωμένου από κατάσταση ανεσταλμένης λειτουργίας (sleep mode).
- **Watchdog Timer:** είναι ένας 8-bit timer που χρησιμοποιείται για την ανίχνευση εκτέλεσης απροσδόκητης σειράς κώδικα, προκειμένου να προκαλέσει reset του συστήματος ή κάποια διακοπή.

4.1.9 Ρολόι

Το σύστημα είναι εξοπλισμένο με δύο ταλαντωτές σχεδιασμένους για χαμηλή κατανάλωση και υψηλή σταθερότητα. Υπάρχουν δύο κρυσταλλικοί ταλαντωτές στο σύστημα, ένας στα 16MHz και ένας ακόμα στα 32.768kHz. Ο δεύτερος ταλαντωτής χρησιμοποιείται σαν ρολόι όταν το σύστημα βρίσκεται σε κατάσταση ανεσταλμένης λειτουργίας.

4.1.10 Διαχείριση Ενέργειας

Το DA14580 διαθέτει ένα ολοκληρωμένο σύστημα διαχείρισης ενέργειας με έναν πλήρως ενσωματωμένο υποβιβασμού ή ενίσχυσης DCDC μετατροπέα και ξεχωριστά LDO's για τους διάφορους ενεργειακούς τομείς του συστήματος.

4.2 DA14580 Development Kit και Software Development Kit

Για την υλοποίηση της συσκευής χρησιμοποιήθηκαν τα εργαλεία που παρέχει η Dialog για την ανάπτυξη εφαρμογών. Τα εργαλεία αυτά, περιλαμβάνουν το υλικό, DA14580 Expert Development Kit (EIKONA 21), καθώς και το λογισμικό, που είναι ένα σύνολο από παραδείγματα κώδικα με τα αντίστοιχα συνοδευτικά έγγραφα, τα διάφορα profile καθώς και τα template για την δημιουργία ενός καινούριου project.

Στην παρούσα παράγραφο θα γίνει μια περιγραφή για τα διαθέσιμα εργαλεία που χρησιμοποιήθηκαν (υλικό και λογισμικό) προκειμένου να γίνει η υλοποίηση της συσκευής.

4.2.1 Hardware Development Kit

Όσον αφορά το υλικό τα τρία βασικά κομμάτια που το αποτελούν είναι (EIKONA 21) η μητρική πλακέτα (motherboard), (EIKONA 22, 23, 24) η κεντρική πλακέτα (daughter board), και (EIKONA 25) το usb Dongle.

Η μητρική πλακέτα παρέχει όλους τους απαραίτητους ακροδέκτες και κυκλώματα για μέτρηση ενέργειας, σύνδεση αισθητηρίων και συσκευής εκσφαλμάτωσης. Επίσης δίνει την

δυνατότητα στον χρήστη να επιλέξει την πηγή ενέργειας που επιθυμεί, διαθέτει υποδοχές για μπαταρίες Coin Shell 3v3 ή Alkaline 1v5 καθώς και ρύθμιση του ολοκληρωμένου στις διαφορετικές καταστάσεις λειτουργίας (Buck ή Boost Mode) με χρήση jumper's. Επίσης δίνεται και η επιλογή σαν πηγή ενέργειας να χρησιμοποιηθεί και τάση από την USB σύνδεση (σε αυτή την επιλογή το ολοκληρωμένο θα πρέπει να τεθεί σε κατάσταση λειτουργίας Buck). Τέλος η μητρική παρέχει δύο κουμπιά (K1 και K2) καθώς και LED για λειτουργίες εκφαλμάτωσης και βέβαια την απαραίτητη διεπαφή για την σύνδεση της κεντρικής πλακέτας.



Εικόνα 21: Μητρική πλακέτα Expert Development Kit.

Η κεντρική πλακέτα περιέχει το ολοκληρωμένο DA14580 καθώς και το κύκλωμα της κεραίας. Βγαίνει σε τρεις εκδόσεις όπου η κάθε μια περιλαμβάνει το κύκλωμα DA14580 σε διαφορετική συσκευασία (WLCSF, QFN40, QFN48 package, εικόνες 22, 23, 24 αντίστοιχα). Η εφαρμογή αναπτύχθηκε σε κεντρική πλακέτα με συσκευασία WL-CSP34.



Εικόνα 22: WLCSF συσκευασία.



Εικόνα 23: QFN40 συσκευασία.



Εικόνα 24: QFN48 συσκευασία.

Το USB Dongle (EIKONA 25) αποτελεί ένα βοηθητικό εργαλείο που παρέχει η Dialog προκειμένου ο χρήστης να μπορεί να συνδιαλεγει με μία συσκευή, που ενδεχομένως είναι το Development Kit σε ρόλο Client. Το Dongle παρέχεται προκειμένου ο χρήστης να μπορέσει να αναπτύξει κατά κύριο λόγο Host εφαρμογές με την χρήση εξωτερικού επεξεργαστή. Στο συγκεκριμένο σενάριο όλη η στοίβα του Bluetooth πρωτοκόλλου εκτελείται στο Dongle και το application εκτελείται από τον κύριο επεξεργαστή, που είναι συνδεδεμένο. Το Dongle επικοινωνεί με την Host εφαρμογή μέσω μίας διεπαφής (Generic Transport Layer) με την χρήση UART.



Εικόνα 25: USB Dongle

4.2.2 Software Development Kit

Το Software Development Kit περιλαμβάνει την υλοποίηση του πρωτοκόλλου Bluetooth Low Energy όπως ορίζεται από το πρότυπο της έκδοσης 4.0. Επίσης προκειμένου ο προγραμματιστής να εξοικειωθεί με τον εξοπλισμό και με τους μηχανισμούς του DA14580 διατίθεται πληθώρα παραδειγμάτων, εφαρμογών και οδηγιών για τα διάφορα περιφερειακά του επεξεργαστή. Μαζί με τα παραπάνω δίνεται ένα επιπλέον, πλήρως παραμετροποιήσιμο αρχείο που μπορεί να χρησιμοποιηθεί σαν template προκειμένου ο προγραμματιστής να αρχίσει την ανάπτυξη της εφαρμογής από μηδενική βάση. Στην παρούσα παράγραφο θα κάνουμε μια μικρή περιγραφή του template project και θα αναφέρουμε σημαντικά κομμάτια για το πώς δομήθηκε η ανάπτυξη της εφαρμογής. Εφόσον η εφαρμογή χρησιμοποιεί τον επεξεργαστή του ολοκληρωμένου για την λειτουργία της γίνεται περιγραφή μόνο του fully embedded συστήματος.

Όλες οι ρυθμίσεις της εφαρμογής περιλαμβάνονται ως επί το πλείστον σε ένα αρχείο (da14580_config.h). Από εκεί ο προγραμματιστής μπορεί ορίσει μια σειρά από επιλογές, οι πιο σημαντικές παρατίθενται στον παρακάτω ΠΙΝΑΚΑ:

Directive	Ορισμένη (Defined)	Μη – ορισμένη (Undefined)
CFG_APP	Η εφαρμογή τρέχει στον επεξεργαστή του	Η εφαρμογή τρέχει σε

	ολοκληρωμένου.	εξωτερικό επεξεργαστή.
CFG_PRF_<profile>	Το αντίστοιχο προφίλ χρησιμοποιείται από την εφαρμογή.	Το προφίλ δεν χρησιμοποιείται από την εφαρμογή.
CFG_APP_<application>	Όνομα της εφαρμογής, πρέπει να οριστεί σε περίπτωση που η εφαρμογή τρέχει αποκλειστικά στο ολοκληρωμένο.	
CFG_NVDS	Η εφαρμογή χρησιμοποιεί τις μη-πτητικές δομές αποθήκευσης.	Η εφαρμογή δεν χρησιμοποιεί μη-πτητικές δομές αποθήκευσης.
CFG_APP_SEC	Χρησιμοποιείται η ασφάλεια BLE.	Δεν χρησιμοποιείται ασφάλεια BLE.
CFG_WDOG	Ο χρονιστής watchdog είναι ενεργοποιημένος.	Ο χρονιστής watchdog δεν είναι ενεργοποιημένος.
CFG_EXT_SLEEP CFG_DEEP_SLEEP	Το αρχικό sleep mode που χρησιμοποιεί η εφαρμογή.	Η εφαρμογή δεν χρησιμοποιεί sleep mode ή ενεργοποιείται αργότερα από την εφαρμογή.
BLE_CONNECTION_MAX_USER	Μέγιστος αριθμός συνδέσεων που μπορεί να υποστηρίξει η συσκευή (1-6).	
DEVELOPMENT_DEBUG	Η λειτουργία εκσφαλμάτωσης είναι ενεργοποιημένη.	Η λειτουργία εκσφαλμάτωσης δεν είναι ενεργοποιημένη.
APP_BOOT_FROM_OTP	Η εκκίνηση της συσκευής γίνεται από την OTP.	Η εκκίνηση της συσκευής δεν γίνεται από την OTP.

Πίνακας 1313: Επιλογές εφαρμογής από το αρχείο da14580_config.h

Το ολοκληρωμένο για την εξυπηρέτηση των απαιτήσεων του συστήματος στηρίζεται σε ένα μικρό λειτουργικό σύστημα μέσω του οποίου ανταλλάσσονται μηνύματα μεταξύ των διαφόρων διεργασιών, αποθηκεύονται και εκτελεί επίσης λειτουργίες χρονιστή. Όλες οι διαδικασίες που σχετίζονται με το BLE και τα αντίστοιχα profile όταν εκτελούνται από το λειτουργικό σύστημα ανταλλάσσουν μηνύματα μέσω αυτού. Όταν μια εφαρμογή χρησιμοποιεί τον επεξεργαστή του ολοκληρωμένου θα πρέπει και αυτή να δηλωθεί σαν διεργασία στο λειτουργικό σύστημα προκειμένου να μπορέσει το λειτουργικό να την συμπεριλάβει στο χρονοπρογραμματισμό.

Κατά την διαδικασία εκκίνησης το ολοκληρωμένο ξεκινά από το function *int main_func(void)*, όπου και γίνονται οι απαραίτητες αρχικοποιήσεις της πλατφόρμας DA14580 και ακολουθεί η αρχικοποίηση της BLE στοίβας. Στην συνέχεια, αν απαιτείται η χρήση του εσωτερικού επεξεργαστή, τότε η εφαρμογή αρχικοποιείται (*app_init()*). Έπειτα το πρόγραμμα μπαίνει σε μια ατέρμονη επανάληψη όπου εκεί το λειτουργικό σύστημα εξυπηρετεί όλα τα μηνύματα από διάφορα BLE γεγονότα (με την εντολή *rwip_scedule()*) και κατόπιν ελέγχεται αν το σύστημα θα αλλάξει κατάσταση από κανονική σε ανεσταλαμμένη λειτουργία (sleep mode). Στο

τέλος η εντολή *WFI()* καλείται, προκειμένου να ανασταλεί η λειτουργία του επεξεργαστή μέχρι το επόμενο BLE γεγονός.

Όπως περιγράφηκε στην προηγούμενη παράγραφο μέσα στην διαδικασία εκκίνησης του συστήματος περιλαμβάνεται και η διαδικασία αρχικοποίησης της ενσωματωμένης εφαρμογής (αν αυτή υπάρχει). Κατά την διάρκεια αυτής της διαδικασίας εκτελούνται διεργασίες όπως η αρχικοποίηση των διαφόρων προφίλ που χρειάζεται η εφαρμογή, η δήλωση και δημιουργία της εφαρμογής σαν διεργασία του λειτουργικού συστήματος, η αρχικοποίηση ασφάλειας (αν περιέχει κάποια ασφάλεια η εφαρμογή) καθώς και αρχικοποίηση υλικού που χρησιμοποιείται.

Πιο πριν επίσης αναφέρθηκε ότι κατά την διάρκεια της αρχικοποίησης εκτελούνται και οι διαδικασίες για την αρχικοποίηση της στοίβας BLE. Όταν το τμήμα GAP είναι έτοιμο προκειμένου να προσφέρει υπηρεσίες στα πιο πάνω επίπεδα προωθεί το μήνυμα *GAPM_DEVICE_READY_IND*. Η εφαρμογή με την λήψη του μηνύματος εκτελεί τον αντίστοιχο χειρισμό καλώντας την function *gapm_device_ready_ind_handler()*. Η προηγούμενη function θα απαντήσει με *GAPM_RESET* μήνυμα, στην *TASK_GAPM* διεργασία. Η ολοκλήρωση του *GAPM_RESET* απαντάται με *GAPM_CMP_EVT* και καλείται ο αντίστοιχος χειριστής. Ο χειριστής θα εκτελέσει ενέργειες ρύθμισης και θα στείλει και το απαραίτητο μήνυμα *GAP_SET_DEV_CONFIG_CMD* στην διεργασία *TASK_GAPM*. Αυτό το μήνυμα θα απαντηθεί με *GAPM_CMP_EVT* ολοκληρώνοντας την προηγούμενη ρύθμιση και την αρχικοποίηση του BLE πρωτοκόλλου. Στην συνέχεια ακολουθούν μια σειρά από μηνύματα και διεργασίες που αφορούν την αρχικοποίηση των διαφόρων προφίλ και των αντίστοιχων βάσεων που χρησιμοποιούν. Τέλος αφού έχουν αρχικοποιηθεί όλα τα προφίλ της εφαρμογής το σύστημα μπαίνει σε διαδικασία διαφήμισης.

Όποια διεργασία εκτελείται στο ολοκληρωμένο πραγματοποιείται όσο το πρόγραμμα βρίσκεται μέσα στην ατέρμονη επανάληψη. Μέσα στην επανάληψη αυτή όπως ειπώθηκε εξυπηρετούνται όλα τα BLE γεγονότα από το λειτουργικό (*rwip_schedule()*) και στην συνέχεια εκτελούνται μια σειρά από διαδικασίες προκειμένου το σύστημα είτε να πέσει σε κατάσταση ανεσταλμένης λειτουργίας είτε να μείνει σε κανονική λειτουργία για έναν ακόμα κύκλο.

Το ότι, το λειτουργικό σύστημα εξυπηρετεί μηνύματα μόνο εφόσον ο πυρήνας BLE και το ράδιο είναι ανοιχτά περιορίζει την εκτέλεση διεργασιών σε συγκεκριμένες χρονικές στιγμές. Για παράδειγμα αν η αποστολή BLE πακέτου στηρίζεται στο συμβάν μιας διακοπής, η αποστολή θα πρέπει να μείνει σε αναμονή, έως ότου όλα τα απαραίτητα τμήματα να είναι ανοιχτά προκειμένου το λειτουργικό να κάνει την εξυπηρέτηση του μηνύματος και να γίνει η αποστολή.

Οι διαδικασίες αυτές έχουν ως σκοπό να συγχρονίσουν ασύγχρονα συμβάντα (για παράδειγμα διακοπές) που ενδεχομένως μπορεί να συμβούν όσο το ολοκληρωμένο έχει τα απαραίτητα τμήματα για αποστολή πακέτων κλειστά καθώς επίσης και να αλλάξουν δυναμικά την είσοδο και έξοδο του ολοκληρωμένου από και προς την κατάσταση ανεσταλμένης λειτουργίας.

#	Function Name	Functionality	Timing Constraints
1	<i>app_asynch_trm()</i>	Χρήση για την αποστολή μηνυμάτων στο λειτουργικό σύστημα από ασύγχρονα συμβάντα που έχουν επεξεργαστεί στην συνάρτηση <i>app_asynch_proc()</i> .	Medium

2	app_asynch_proc()	Χρήση για την επεξεργασία ασύγχρονων συμβάντων.	Medium
3	app_asynch_sleep_proc()	Χρήση για την ενημέρωση της κατάστασης της εφαρμογής σύμφωνα με τα τελευταία γεγονότα.	Medium
4	app_sleep_prepare_proc()	Χρήση για την ακύρωση εισόδου σε ανεσταλμένη λειτουργία βάση της τελευταίας κατάστασης της εφαρμογής.	Hard
5	app_sleep_entry_proc()	Χρήση για εκτέλεση συγκεκριμένων διαδικασιών πριν το ολοκληρωμένο εισέλθει σε κατάσταση ανεσταλμένης λειτουργίας.	Hard
6	app_sleep_exit_proc()	Χρήση για εκτέλεση διεργασιών αμέσως μετά την επαναφορά του συστήματος σε κανονική λειτουργία.	Hard

Πίνακας 1414: Συναρτήσεις απόφασης για ανεσταλμένη λειτουργία του ολοκληρωμένου και συγχρονισμό γεγονότων.

Η παραπάνω αρχιτεκτονική του λογισμικού ακολουθείται μόνο σε περίπτωση συστήματος που βασίζεται σε διακοπές. Σε άλλες περιπτώσεις εφόσον δεν υπάρχουν ασύγχρονα γεγονότα και ο Host με τον Client είναι συγχρονισμένοι, κάθε συμβάν θα εκτελείται ενώ το λειτουργικό σύστημα είναι ενεργό.

Κεφάλαιο 5 – Αναλυτική λειτουργία της συσκευής

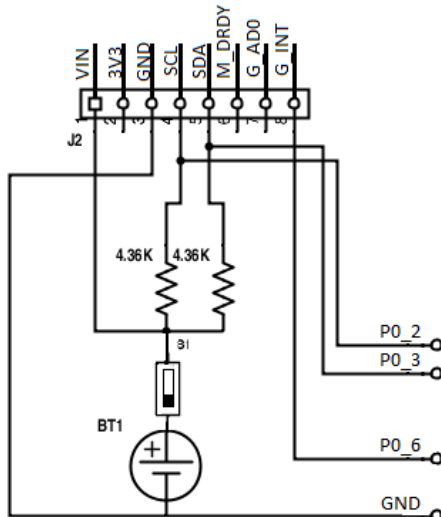
Στο παρόν κεφάλαιο θα αναφερθούμε στην υλοποίηση της συσκευής από πλευράς υλικού, καθώς και αρχιτεκτονικής λογισμικού που ακολουθήθηκε. Στην πρώτη παράγραφο του κεφαλαίου παρατίθεται το κύκλωμα της συσκευής και οι απαραίτητες συνδέσεις με το αναπτυξιακό και έπειτα, σε επόμενες παραγράφους αναλύεται το λογισμικό που συντάχθηκε για την λειτουργία της συσκευής.

5.1 Υλοποίηση Υλικού

Όπως αναφέρθηκε και στο κεφάλαιο 3 το δαχτυλίδι χρησιμοποιεί το IMU GY-88 που είναι ένα αισθητήριο αδράνειας αποτελούμενο από ένα σύνολο επιμέρους αισθητηρίων. Από τα επιμέρους τμήματα του IMU μόνο το αισθητήριο MPU6050, που περιελάμβανε το επιταχυνσιόμετρο και το γυροσκόπιο, χρησιμοποιήθηκε. Το αισθητήριο διαθέτει I²C διεπαφή προκειμένου να συνδεθεί με τον επεξεργαστή του ολοκληρωμένου. Η συσκευή αποτελείται από μια διάτρητη πλακέτα πάνω στην οποία είναι τοποθετημένα:

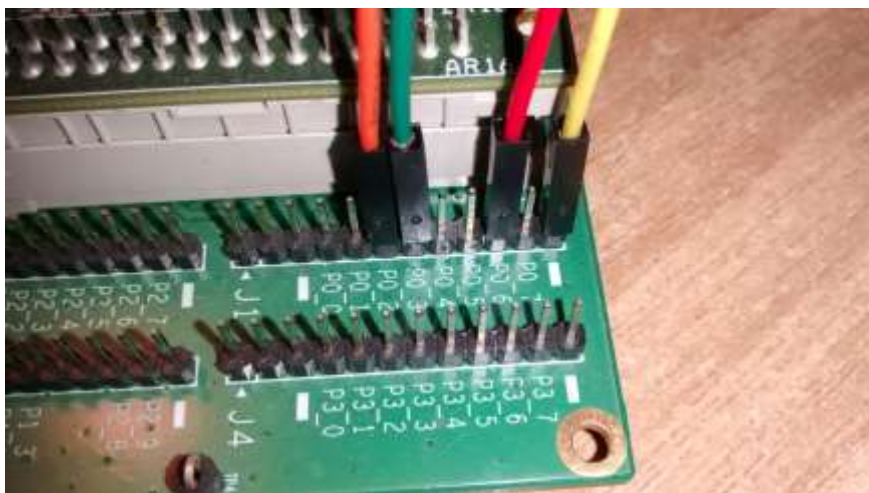
- Ένα σύνολο 8 θηλυκών ακροδεκτών για σύνδεση του αισθητηρίου.
- Δύο αντιστάσεις 4,36kΩ που είναι απαραίτητες για το κύκλωμα επικοινωνίας I²C.
- Μια θέση για μπαταρία coin cell CR2032 3,3V.
- Ένας διακόπτης λειτουργίας on/off

Στις εικόνες 26 και 27 παρατίθεται το σχηματικό του κυκλώματος της εφαρμογής και η σύνδεση με το αναπτυξιακό αντίστοιχα.



Εικόνα 2626: Σχηματικό κυκλώματος εφαρμογής δαχτυλιδιού.

Ανάπτυξη συσκευής εισόδου με την χρήση τεχνολογίας Bluetooth Low Energy και αισθητηρίων αδράνειας



Εικόνα 2727: Σύνδεση κυκλώματος με το αναπτυξιακό του DA14580.

Πάνω στην διάτρητη πλακέτα με το υλοποιημένο κύκλωμα κολλήθηκε ένα δεματικό προκειμένου ο χρήστης να προσαρμόσει την συσκευή στον δείκτη του χεριού. Στην ΕΙΚΟΝΑ 28 δίνεται η ολοκληρωμένη συσκευή.



Εικόνα 2828: Ολοκληρωμένη κατασκευή προσαρμοσμένη στο χέρι χρήστη.

5.2 Εκκίνηση εφαρμογής

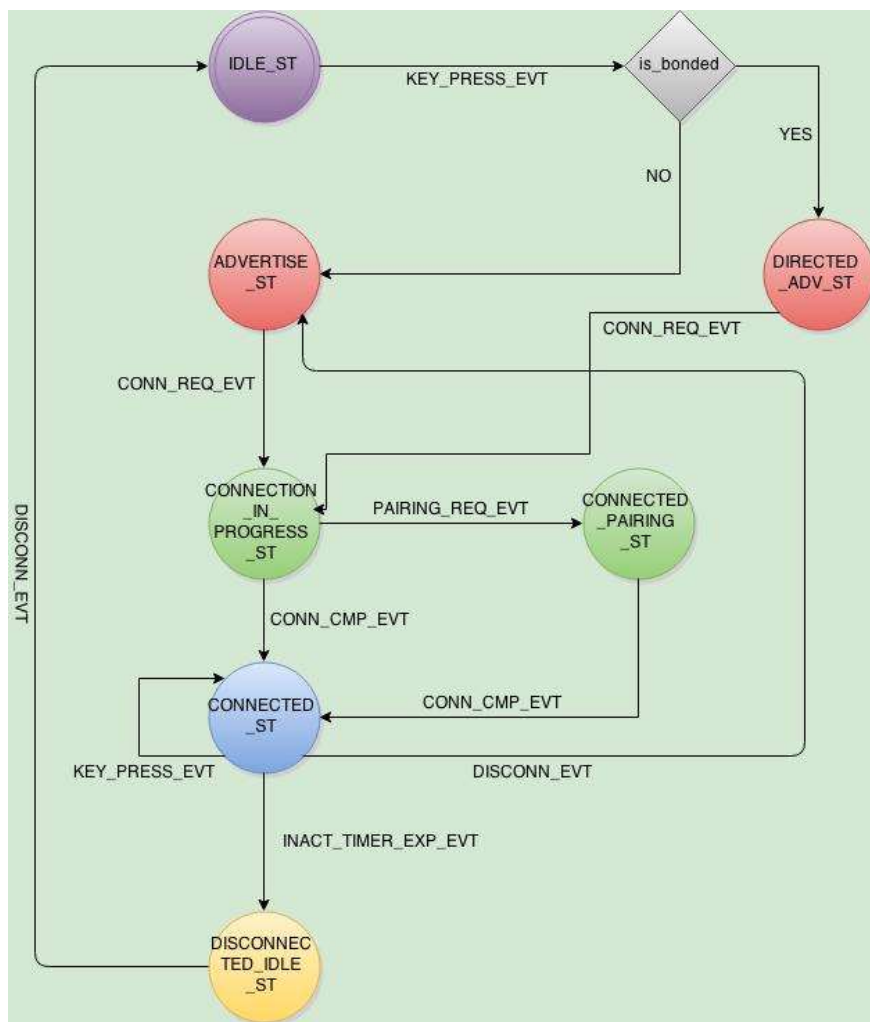
Όπως επώθηκε και στην παράγραφο 4.2.2 κατά την εκκίνηση το ολοκληρωμένο ξεκινά εκτελώντας όλες οι απαραίτητες αρχικοποιήσεις που αφορούν, την στοίβα του BLE πρωτοκόλλου, την εφαρμογή και τα διάφορα περιφερειακά και ακροδέκτες. Έπειτα αφού έχει ολοκληρωθεί η αρχικοποίηση το σύστημα προχωρά στην ενεργοποίηση των profile που υποστηρίζει η εφαρμογή. Μετά το πέρας και της αρχικοποίησης των profile το σύστημα προχωρά σε ενεργοποίηση και ρύθμιση του αισθητηρίου. Η ρύθμιση του αισθητηρίου περιλαμβάνει:

- Επιλογή ρολογιού για το κύκλωμα του ολοκληρωμένου.
- Επιλογή εύρους μετρήσεων για το επιταχυνσίμετρο και το γυροσκόπιο (στην παρούσα έκδοση της εφαρμογής η μέγιστη τιμή των αισθητηρίων είναι 2g και 250 degrees/sec).
- Ορισμός της δειγματοληψίας στα 100Hz.
- Επιλογή του ψηφιακού χαμηλοπερατού φίλτρου σε επιθυμητή συχνότητα αποκοπής (τιμή τρέχουσας έκδοσης κώδικα 44Hz).
- Ρύθμιση των διακοπών κίνησης (κατώφλι επιτάχυνσης – 32mg, χρονικό κατώφλι – 6ms, ρυθμός μείωσης μετρητή – ανά 1 δείγμα).
- Επιλογή του ψηφιακού υψιπερατού φίλτρου σε επιθυμητή συχνότητα αποκοπής (τιμή τρέχουσας έκδοσης κώδικα 0.63Hz).
- Ρύθμιση συμπεριφοράς διακοπών.
- Ενεργοποίηση διακοπών.

Κατόπιν ενεργοποιούνται οι διακοπές στον ARM M0 και στην συνέχεια αναγκάζεται το σύστημα να πέσει για ύπνο (extended sleep) με μόνη επιλογή αφύπνισης κάποια εξωτερική διακοπή. Εφόσον ο χρήστης κουνήσει την συσκευή ξεπερνώντας το κατώφλι που έχουμε θέσει η συσκευή ξυπνά και αρχίζει να διαφημίζεται είτε στοχευμένα είτε γενικά προς κάθε διαθέσιμη συσκευή.

5.3 Κεντρικός Μηχανισμός Λειτουργίας

Το σύστημα για την συμπεριφορά της εφαρμογής είναι βασισμένο στο πρότυπο σχέδιο της Dialog για πληκτρολόγιο [7] και για ποντίκι [8]. Στην ΕΙΚΟΝΑ 29 παρατίθεται το διάγραμμα του κεντρικού μηχανισμού λειτουργίας. Το σύστημα, όταν γίνεται η εκκίνηση είναι σε IDLE_ST κατάσταση, και περιμένει να γίνει μια απότομη κίνηση από τον χρήστη. Κατόπιν το σύστημα διαφημίζεται προκειμένου να γίνει ορατό από τις συσκευές που ανιχνεύουν. Από την στιγμή που κάποια συσκευή στείλει αίτηση για σύνδεση, το σύστημα θα περάσει από τις καταστάσεις CONNECTION_IN_PROGRESS_ST και CONNECTED_PAIRING_ST για ολοκληρώσει τις διαδικασίες σύνδεσης και σύζευξης. Όταν ολοκληρωθούν οι προηγούμενες διαδικασίες το σύστημα παραμένει σε κατάσταση CONNECTED_ST και λειτουργεί κανονικά. Αν ο χρήστης δεν κινήσει τον κέρσορα για συγκεκριμένο χρονικό διάστημα το σύστημα αποσυνδέεται και επιστρέφει σε IDLE_ST. Όταν ο χρήστης κουνήσει τον κέρσορα το σύστημα αυτόματα, διαφημίζεται στοχευμένα και κάνει επανασύνδεση με την συσκευή. Αν ο χρήστης αποφασίσει να αποσυνδέσει την συσκευή από τον Host τότε το σύστημα θα αποσυνδεθεί και θα αρχίσει να διαφημίζεται.



Εικόνα 2929: Κύριος μηχανισμός λειτουργίας εφαρμογής δαχτυλιδιού.

Ο κεντρικός μηχανισμός λειτουργίας ορίζει την συμπεριφορά του συστήματος ανάλογα με την κατάσταση στην οποία βρίσκεται την εκάστοτε στιγμή. Η συνάρτηση που ορίζει την συμπεριφορά του συστήματος είναι η `void app_state_update(enum main_fsm_events)`. Η συνάρτηση είναι μια μηχανή πεπερασμένων καταστάσεων (finite state machine) που τροφοδοτείται από συμβάντα, η συμπεριφορά του συστήματος εξαρτάται από την παρούσα κατάσταση του συστήματος και από το συμβάν που εκτελείται. Παρακάτω παρατίθενται οι καταστάσεις της μηχανής και τα αντίστοιχα συμβάντα που τον τροφοδοτούν:

- **IDLE:** σ' αυτήν την κατάσταση το σύστημα είναι ανενεργό και σε κατάσταση ανεσταλμένης λειτουργίας.
 - **NO_EVENT:** συμβάν καλείται μόνο μια φορά, μετά το πέρας της αρχικοποίησης των διαφόρων profile. Με αυτό το συμβάν στην κατάσταση IDLE, το σύστημα πέφτει σε κατάσταση ανεσταλμένης λειτουργίας και περιμένει διακοπή από το αισθητήριο προκειμένου να λειτουργήσει κανονικά.
 - **KEY_PRESS_EVT:** καλείται όταν υπάρχει διακοπή από τον Wake-up χρονιστή. Το σύστημα σε κατάσταση IDLE, με αυτό το συμβάν, αν είναι συζευγμένο επιχειρεί στοχευόμενη διαφήμιση και επανασύνδεση με την συσκευή που ήταν συνδεδεμένο αν δεν ήταν συζευγμένο με καμιά συσκευή το σύστημα διαφημίζεται προς όλες τις διαθέσιμες συσκευές. Τέλος αλλάζει την κατάσταση του συστήματος σε DIRECTED_ADV_ST ή ADVERTISE_ST.
- **ADVERTISE_ST:** σ' αυτήν την κατάσταση το σύστημα βρίσκεται σε κατάσταση γενικής διαφήμισης.
 - **CONN_REQ_EVT:** συμβάν όταν υπάρξει αίτηση για σύνδεση με άλλη συσκευή, στην συνέχεια αλλάζει η κατάσταση του συστήματος σε CONNECTION_IN_PROGRESS_ST.
- **CONNECTION_IN_PROGRESS_ST:** το σύστημα έχει δεχθεί αίτηση για σύνδεση και αναμένει αίτηση για σύζευξη.
 - **PAIRING_REQ_EVT:** το σύστημα δέχεται αίτηση για σύζευξη από άλλη συσκευή και προχωρά σε κατάσταση CONNECTED_PAIRING_ST.
 - **CONN_CMP_EVT:** το σύστημα ολοκληρώνει μία σύνδεση μετά από στοχευόμενη διαφήμιση. Επαναρχικοποιεί τις απαραίτητες μεταβλητές και το αισθητήριο και ενεργοποιεί τον χρονιστή αδράνειας. Η κατάσταση του συστήματος αλλάζει σε CONNECTED_ST.
- **CONNECTED_ST:** το σύστημα είναι συνδεδεμένο με κάποια συσκευή.
 - **INACT_TIMER_EXP_EVT:** συμβάν που εκδηλώνεται όταν ο χρονιστής αδράνειας λήγει (ο χρήστης έχει ώρα να κουνήσει την συσκευή). Η συσκευή αποσυνδέεται, η ουρά του αισθητηρίου απενεργοποιείται, μόνο η διακοπή κίνησης είναι ενεργή. Το σύστημα αλλάζει κατάσταση σε DISCONNECTED_IDLE_ST.
 - **KEY_PRESS_EVT:** συμβάν που ανανεώνει τον μετρητή αδράνειας σε κάθε κίνηση του χρήστη όσο η συσκευή είναι συνδεδεμένη.
 - **DISCONN_EVT:** συμβάν που εκδηλώνεται όταν ο χρήστης θέλει να αποσυνδεθεί από την συσκευή. Το σύστημα αδειάζει τις ουρές που αποθηκεύουν ενέργειες του χρήστη και τις αναφορές προς τον Host, απενεργοποιεί τους χρονιστές και τον αισθητήρα και στην συνέχεια αρχίζει να διαφημίζεται προς όλες τις διαθέσιμες συσκευές. Η κατάσταση του συστήματος αλλάζει σε ADVERTISE_ST.
- **CONNECTED_PAIRING_ST:** στην παρούσα κατάσταση το σύστημα ολοκληρώνει την διαδικασία σύζευξης και σύνδεσης.
 - **CONN_CMP_EVT:** Συμβάν ολοκλήρωσης σύζευξης. Ενεργοποιείται η ουρά του αισθητηρίου και οι αντίστοιχες διακοπές και αρχικοποιούνται οι ουρές που

αποθηκεύουν τις αναφορές προς τον Host. Τέλος ενεργοποιείται ο χρονιστής που μετρά την αδράνεια του συστήματος και το σύστημα πηγαίνει σε κατάσταση CONNECTED_ST.

- **DISCONNECTED_IDLE_ST:** το σύστημα μπαίνει σ' αυτή την κατάσταση όταν λήγει ο χρονιστής αδράνειας και το σύστημα αποσυνδέεται από τον Host.
 - **DISCONN_EVT:** συμβάν που εκδηλώνεται όταν ολοκληρωθεί η αποσύνδεση. Το σύστημα αδειάζει τις ουρές και απενεργοποιεί τους χρονιστές. Έπειτα μπαίνει σε κατάσταση ανεσταλμένης λειτουργίας (extended sleep) και μπορεί να «ξυπνήσει» μόνο από διακοπή κίνησης. Τέλος η κατάσταση του συστήματος αλλάζει σε IDLE_ST.
- **DIRECTED_ADV_ST:** κατάσταση κατά την οποία το σύστημα διαφημίζεται μόνο προς την συσκευή που ήταν πριν συνδεδεμένο.
 - **CONN_REQ_EVT:** αίτηση σύνδεσης κατά την διάρκεια στοχευόμενης διαφήμισης. Το σύστημα αλλάζει την κατάσταση του σε CONNECTION_IN_PROGRESS_ST.

5.4 Μηχανισμός Ουράς για τις αναφορές προς τον Host

Ο μηχανισμός της ουράς είναι επίσης βασισμένοι στα πρότυπα σχέδια της Dialog για πληκτρολόγιο [7] και για ποντίκι [8]. Το σύστημα χρησιμοποιεί τρεις ουρές για την αποθήκευση και την αποστολή αναφορών. Η πρώτη ουρά (*gst_buffer[]*) αποθηκεύει τις χειρονομίες που κάνει ο χρήστης, όταν αυτές αναγνωρίζονται σαν ενέργειες από τον αλγόριθμο (για παράδειγμα πάτημα ενός κουμπιού από το ποντίκι). Οι δύο άλλες ουρές αποθηκεύουν αναφορές που είναι έτοιμες προς αποστολή (*kbd_trim_list*), και κρατάνε το διαθέσιμο των πεπερασμένων θέσεων που υποστηρίζει η συσκευή για αποθήκευση αναφορών (*kbd_free_list*).

Το πρόγραμμα γεμίζει την ουρά των ενεργειών και κινήσεων που κάνει ο χρήστης από δύο διαφορετικά σημεία.

- Αφού διαβαστούν οι τιμές από το γυροσκόπιο και γίνει η κατάλληλη επεξεργασία συγκρίνονται με την προηγούμενη τιμή που στάλθηκε στον Host. Αν η εκάστοτε τιμή διαφέρει από την προηγούμενη τιμή τότε το σύστημα θεωρεί πως υπάρχει κίνηση από τον χρήστη. Από την στιγμή που το σύστημα αντιληφθεί κίνηση του κέρσορα αφαιρεί μια θέση από την ουρά των ελεύθερων θέσεων (*kbd_free_list*), συμπληρώνει τα απαραίτητα πεδία για την πληροφορία της κίνησης και η έτοιμη πλέον αναφορά της νέας κίνησης εισέρχεται στην ουρά με τις αναφορές που εκκρεμούν (*kbd_trim_list*).
- Το δεύτερο σημείο από όπου γεμίζουν οι λίστες είναι όταν η εφαρμογή έχει αντιληφτεί χειρονομία, έχει κάνει την απαραίτητη επεξεργασία και αφού αναγνωρίσει το είδος της χειρονομίας την τοποθετεί στην ανάλογη ουρά (*gst_buffer[]*). Η ουρά από εκεί και έπειτα ελέγχεται λίγο πριν αρχίσει να γίνεται η αποστολή προς τον Host. Αν η συγκεκριμένη ουρά έχει μέσα χειρονομίες τότε το πρόγραμμα αφαιρεί μια θέση από την ουρά με τις διαθέσιμες αναφορές (*kbd_free_list*), συμπληρώνει τα δεδομένα που αντιστοιχούν στην όποια κίνηση έχει αναγνωριστεί και έπειτα τοποθετεί την αναφορά στην ουρά με τις εκκρεμείς αναφορές (*kbd_trim_list*).

Το σύστημα συνεχίζει να γεμίζει τις λίστες έως ότου το πρόγραμμα φθάσει σε σημείο να στείλει τις κινήσεις στον Host. Κάθε φορά που το πρόγραμμα φτάνει στο σημείο να κάνει αποστολή των αναφορών παραμένει στο σημείο και συνεχίζει την αποστολή έως ότου αδειάσει

η ουρά (*kbd_trim_list*). Αν το σύστημα δεν έχει άλλα δεδομένα για αποστολή τότε συνεχίζει κανονικά την εκτέλεση.

5.5 Μηχανισμός κίνησης του κέρσορα

Το ότι το σύστημα θα έπρεπε να τοποθετηθεί μόνο στο δάχτυλο του χρήστη έθεσε ένα σημαντικό πρόβλημα. Έχοντας το ολοκληρωμένο, που περιελάμβανε και τα δύο αισθητήρια, τοποθετημένο σε ένα σημείο εξέτρεπε τον κέρσορα από την θέση του κάθε φορά που ο χρήστης επιχειρούσε να κάνει μια χειρονομία. Εφόσον το γυροσκόπιο αφορά μόνο την κίνηση του κέρσορα, όταν ο χρήστης επιθυμούσε να κάνει μια χειρονομία, θα πρέπει να σταματήσει να αναφέρει τιμές στον Host προκειμένου η θέση του κέρσορα να παραμένει σταθερή έως ότου ο χρήστης τελειώσει την κίνηση.

Προκειμένου το πρόβλημα να αντιμετωπιστεί χρησιμοποιήθηκε μια ουρά (*gyro_buffer*) στην οποία αποθηκεύονται διαδοχικές αναγνώσεις του γυροσκοπίου. Κάθε φορά που το πρόγραμμα περνάει από το τμήμα ανάγνωσης του γυροσκοπίου η εκάστοτε αναγνωσθείσα τιμή μπαίνει σε αυτήν την ουρά. Όταν η ουρά φτάσει στο μέγιστο των εγγραφών της (*GYRO_WINDOW* η τιμή του τρέχοντος κώδικα είναι 6 θέσεις) το πρόγραμμα βρίσκει την μέση τιμή των εγγραφών. Η παραπάνω μέθοδος, όσον αφορά την μέση τιμή, λειτουργεί σαν φίλτρο στις μικρές διακυμάνσεις τιμών που οφείλονται σε θόρυβο ή πολύ μικρές ακούσιες κινήσεις του χρήστη. Όσον αφορά την ουρά, αυτή εισάγει μια χρονική καθυστέρηση στο σύστημα, σχετικά ανεκτή από τον χρήστη. Αυτή η καθυστέρηση δίνει την δυνατότητα στο σύστημα αναγνώρισης, σε περίπτωση που έχει ανιχνεύσει κάτι, να ακυρώσει τα δεδομένα που είναι ήδη στην ουρά του γυροσκοπίου και ενδεχομένως περιέχουν ανεπιθύμητες τιμές (τιμές που αφορούν χειρονομία και όχι κίνηση κέρσορα). Έτσι λοιπόν κάθε φορά που το σύστημα αντιλαμβάνεται ότι το ο μηχανισμός ανίχνευσης χειρονομιών είναι ενεργός διαγράφει όλες τις τιμές από την ουρά του γυροσκοπίου.

Κάτι επιπλέον που πρέπει να αναφερθεί είναι τρόπος που αναφέρεται η κίνηση στον Host. Το Report Map που έχει η συσκευή δηλώνει στον Host ότι η κίνηση του κέρσορα (x , y) είναι ένας 8-bit αριθμός για κάθε άξονα. Οι τιμές που αναφέρει το γυροσκόπιο είναι 16-bit για κάθε άξονα και κάθε τιμή αποτελείται από ένα High και ένα Low τμήμα σε ξεχωριστούς 8-bit καταχωρητές (για παράδειγμα *GYRO_XOUT_H* και *GYRO_XOUT_LOW* για τον άξονα x). Η ανάγνωση της τιμής για την κίνηση του κέρσορα γίνεται μόνο από τους καταχωρητές *GYRO_XOUT_H* και *GYRO_ZOUT_H* που είναι τα High τμήματα των αξόνων x , z του γυροσκοπίου.

Θεωρήθηκε πως με αυτόν τον τρόπο δεν θα χρειαζόταν να γίνει βαθμονόμηση του γυροσκοπίου προκειμένου να εξαλειφτούν τυχόν αποκλίσεις του αισθητηρίου, εφόσον τυχόν σφάλματα επηρεάζουν τα χαμηλότερα bit του καταχωρητή. Με αυτόν τον τρόπο αποφύγαμε επιπλέον υπολογιστικές εντολές εφόσον η βαθμονόμηση θα απαιτούσε την αφαίρεση του σφάλματος από τις τιμές του γυροσκοπίου. Τέλος η συγκεκριμένη μέθοδος εισήγαγε μια επιπλέον σταθερότητα στην κίνηση του κέρσορα και ανταπόκριση του συστήματος αποκλειστικά σε εκούσιες κινήσεις του χρήστη.

5.6 Χρονισμός συσκευής

Για τον συγχρονισμό του συστήματος χρησιμοποιούνται οι λειτουργίες χρονιστή που διαθέτει το λειτουργικό σύστημα. Η ελάχιστη χρονική διάρκεια που μπορούν να μετρήσουν είναι 10ms και αντίστοιχα πολλαπλάσια του.

Προκειμένου το σύστημα να στέλνει αναφορές και να κάνει αναγνώσεις του γυροσκοπίου ανά τακτά χρονικά διαστήματα θέσαμε έναν χρονιστή συγχρονισμού (*APP_RING_REPORT_SYNCHRONIZATION_TIMER*) με τιμή λήξης τα 10ms. Ο χρονιστής ενεργοποιείται κατευθείαν μετά από την σύνδεση του συστήματος με τον Host και σε κάθε λήξη του, αυξάνεται ένας μετρητής (*inactivity_counter*) και ο χρονιστής επαναφορτώνεται με άλλα 10ms. Όσο ο μετρητής είναι μικρότερος από μια προκαθορισμένη τιμή το σύστημα βρίσκεται σε πλήρη λειτουργία (ο μετρητής μηδενίζεται κάθε φορά που ο χρήστης κουνά τον κέρσορα η κάνει κάποια χειρονομία). Σε πλήρη λειτουργία γίνεται ανάγνωση του γυροσκοπίου, αποθήκευση των τιμών, αποστολή των επεξεργασμένων δεδομένων και το σύστημα είναι διαθέσιμο για όποια διακοπή συμβεί. Όταν ο μετρητής φτάσει τον προκαθορισμένο αριθμό τότε το σύστημα σταματά να κάνει αναγνώσεις αισθητηρίων και αποστολή αναφορών και αλλάζει την κατάσταση του σε ανεσταλμένη λειτουργία (*extended sleep*). Από το σημείο αυτό το σύστημα ξυπνά είτε για να διατηρήσει την σύνδεση είτε όταν συμβεί κάποια διακοπή από το αισθητήριο. Οι ρυθμίσεις του χρονιστή στην παρούσα έκδοση του προγράμματος ορίζουν την τιμή επαναφόρτωσης του χρονιστή στα 10ms (*MOVE_INTERVAL*) και την μέγιστη τιμή του μετρητή στα 100 (*INACTIVITY_COUNTER_MAX*). Αυτό δίνει συνολικό χρόνο 1sec πριν το σύστημα πέσει σε κατάσταση ανεσταλμένη λειτουργίας.

Με την ίδια φιλοσοφία χρησιμοποιείται ένας ακόμα χρονιστής (*APP_RING_INACTIVITY_TIMER*) που αναλαμβάνει αποκλειστικά την περίπτωση που το σύστημα δεν ανιχνεύει κίνηση για μεγάλο χρονικό διάστημα. Ο χρονιστής ξεκινά να μετρά από την στιγμή που το σύστημα συνδέεται με κάποιον Host, σε κάθε λήξη του χρονιστή ένας μετρητής αυξάνεται (*inactivity_timer_counter*). Όταν ο μετρητής φτάσει ένα προκαθορισμένο αριθμό το σύστημα πέφτει σε κατάσταση μόνιμης αναστολής λειτουργίας και μόνο κάποια διακοπή από το αισθητήριο μπορεί να επαναφέρει το σύστημα σε κανονική λειτουργία. Οι ρυθμίσεις του χρονιστή στην παρούσα έκδοση του προγράμματος ορίζουν την τιμή επαναφόρτωσης του χρονιστή στα 15000ms (*INACTIVITY_TIMER_QUOTA*) και την μέγιστη τιμή του μετρητή στα 4 (*INACTIVITY_TIMER_COUNTER_MAX*). Αυτό δίνει συνολικό χρόνο 1 λεπτό πριν το σύστημα πέσει σε κατάσταση μόνιμης ανεσταλμένη λειτουργίας.

Οι χρονιστές που χρησιμοποιούνται από το σύστημα, πέρα από την λειτουργία του συγχρονισμού, (για τον χρονιστή *APP_RING_REPORT_SYNCHRONIZATION_TIMER*) αναλαμβάνουν τον σημαντικό ρόλο της μείωσης κατανάλωσης ενέργειας. Σχετικά με την κατανάλωση γίνεται αναλυτική περιγραφή στην παράγραφο 5.8.

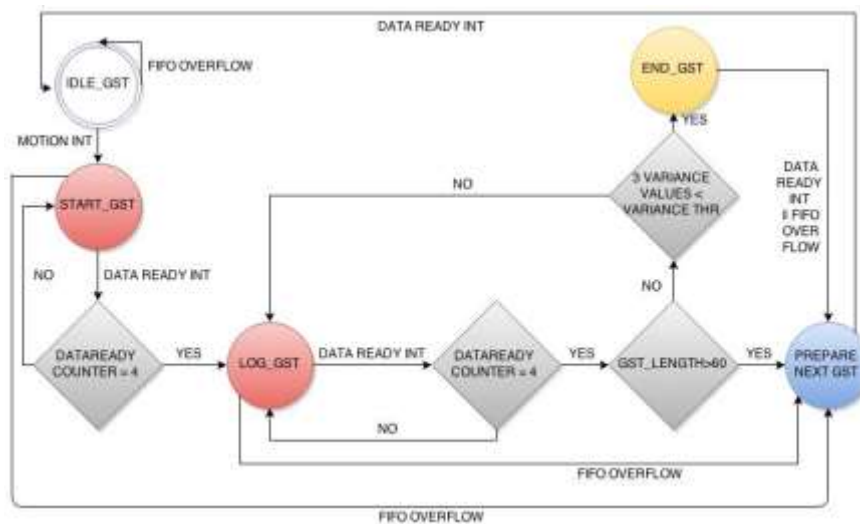
5.7 Μηχανισμός απομόνωσης και ανίχνευσης χειρονομιών

Προκειμένου η εφαρμογή να αντιλαμβάνεται και να εκτελεί ενέργειες οδηγούμενες από τον χρήστη, χωρίς την ύπαρξη διακοπών, έχει την δυνατότητα να καταλαβαίνει ένα προκαθορισμένο σύνολο από συγκεκριμένες χειρονομίες. Αυτές οι χειρονομίες αντιστοιχούν σε συγκεκριμένες ενέργειες, που στην παρούσα έκδοση του κώδικα είναι οι συνηθισμένες ενέργειες που κάνει ένα ποντίκι του εμπορίου.

Το σύστημα αντιλαμβάνεται τις χειρονομίες με την χρήση ενός μηχανισμού βασισμένου σε τεχνικές αναγνώρισης προτύπων. Ο αλγόριθμος δυναμικής χρονικής στρέβλωσης, όπως είναι το όνομα του αλγορίθμου, ή DTW (Dynamic Time Warping), είναι μια διαδικασία κατά την οποία ορίζεται από τον χρήστη ένα πρότυπο αναφοράς και στην συνέχεια ο αλγόριθμος προσπαθεί να ταιριάξει αυτό το πρότυπο με άλλες χρονοσειρές που εισάγει ο χρήστης. Μετά το πέρας της διαδικασίας ταιριάσματος ο αλγόριθμος παράγει ένα κόστος, το κόστος αντιπροσωπεύει το κατά πόσο η εκάστοτε χρονοσειρά μπορεί να ταιριάξει με τα πρότυπα που έχουν οριστεί από τον χρήστη. Όσο πιο μικρό είναι το κόστος, τόσο πιο πολύ οι διαφορετικές χρονοσειρές μοιάζουν. Περισσότερες λεπτομέρειες σχετικά με την επιλογή την χρήση και την βελτιστοποίηση του αλγορίθμου, προκειμένου να λειτουργήσει σε ένα σύστημα περιορισμένων πόρων αναφέρονται στην εργασία [9] η οποία πραγματοποιήθηκε για την συγκεκριμένη εφαρμογή.

Κάτι ακόμα που πρέπει να επισημανθεί είναι ο τρόπος που το σύστημα απομονώνει την χειρονομία προκειμένου ο αλγόριθμος να πραγματοποιήσει την αναγνώριση. Η εφαρμογή χρησιμοποιεί έναν μηχανισμό πεπερασμένων καταστάσεων που στο τέλος καταλήγει στην απόφαση αν η εκάστοτε χρονοσειρά είναι τελικά μια κίνηση αναγνωρίσιμη ή όχι.

Οι χρονοσειρές αποτελούν μετρήσεις επιτάχυνσης και στους τρεις άξονες που διαβάζονται από το επιταχυνσιόμετρο με συχνότητα δειγματοληψίας 100Hz. Το σύστημα μπαίνει σε λειτουργία αναγνώρισης μόλις ο επεξεργαστής δεχθεί διακοπή κίνησης. Η διακοπή κίνησης ενεργοποιείται μόλις οι μετρήσεις από το επιταχυνσιόμετρο ξεπεράσουν τα κατώφλια χρονικά και επιτάχυνσης που έχουν τεθεί στο πρόγραμμα. Από εκεί και έπειτα η εφαρμογή διαβάζει ένα συγκεκριμένο ποσό δεδομένων από την FIFO του αισθητηρίου προκειμένου να συμπεριλάβει στην χρονοσειρά αναγνώρισης δεδομένα πριν δεχθεί την διακοπή κίνησης, ενεργοποιούνται οι διακοπές data ready και απενεργοποιούνται οι διακοπές κίνησης. Με την χρήση των διακοπών ο μηχανισμός πεπερασμένων καταστάσεων συλλέγει και υπολογίζει τα δεδομένα, με την χρήση του αλγορίθμου, κατά την διάρκεια εκτέλεσης της κίνησης από τον χρήστη. Η διαδικασία αναγνώρισης ολοκληρώνεται όταν ο μηχανισμός αντιληφθεί το πέρας της κίνησης. Την λήξη της κίνησης σηματοδοτεί ένα ορισμένο κατώφλι στην τιμή διακύμανσης που υπολογίζεται ανά σταθερό αριθμό δεδομένων. Στην εργασία [9] αναφέρεται πιο αναλυτικά η λειτουργία του μηχανισμού απομόνωσης και ο χρονισμός του μηχανισμού πεπερασμένων καταστάσεων. Στην ΕΙΚΟΝΑ 30 παρατίθεται το διάγραμμα ροής του μηχανισμού αναγνώρισης κίνησης με όλες τις καταστάσεις καθώς και με όλα τα συμβάντα που τον οδηγούν. Οι καταστάσεις με κόκκινο χρώμα υποδεικνύουν την εκτέλεση του αλγορίθμου αναγνώρισης.



Εικόνα 3030: Διάγραμμα ροής μηχανισμού αναγνώρισης και απομόνωσης κίνησης.

5.8 Μηχανισμός μείωσης κατανάλωσης ενέργειας

Στην παράγραφο 5.6 έγινε αναφορά για την λειτουργία των χρονιστών στην εφαρμογή, επισημαίνοντας την χρησιμότητά τους στην μείωση της κατανάλωσης ενέργειας. Στην παρούσα παράγραφο θα αναφερθούμε στις καταναλώσεις που έχουν οι διάφορες καταστάσεις, στις οποίες οδηγείται το σύστημα από τους χρονιστές, αλλά και πως μεταβαίνει το σύστημα από λειτουργία σε λειτουργία.

Στην παράγραφο 4.2.2. αναφερθήκαμε στην αρχιτεκτονική του λογισμικού προκειμένου ένα σύστημα να λειτουργεί βασισμένο σε διακοπές, και έγινε μια περιγραφή των διαδικασιών που συγχρονίζουν και αποφασίζουν για το αν το σύστημα θα πέσει σε κατάσταση ανασταλμένης λειτουργίας ή θα παραμείνει σε κανονική. Θα πρέπει σ' αυτό το σημείο να επισημάνουμε πως οι δύο χρονιστές, για το αν, το σύστημα θα πέσει σε κατάσταση ανασταλμένης λειτουργίας ενεργοποιούνε κάποια flags και η τελική απόφαση λαμβάνεται από τις παρακάτω συναρτήσεις. Στον ΠΙΝΑΚΑ 15 παραθέτουμε πως έγινε η εκμετάλλευση των συναρτήσεων αυτών στην εφαρμογή.

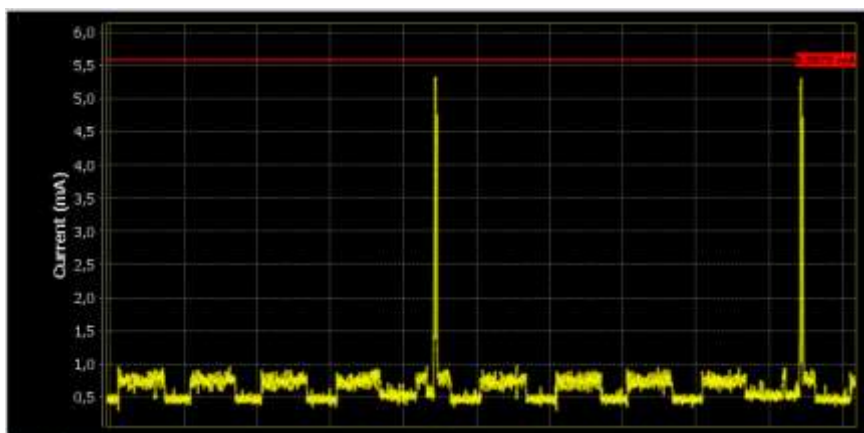
#	Function Name	Functionality
1	app_asynch_trm()	Αν έχουν υπάρχουν χειρονομίες ή κινήσεις του κέρσορα στην ουσά, ετοιμάζονται οι ανάλογες αναφορές, ανανεώνεται ο χρονιστής συγχρονισμού και γίνεται η αποστολή των πακέτων.
2	app_asynch_proc()	Γίνεται η ανάγνωση και η επεξεργασία δεδομένων των τιμών του γυροσκοπίου.
3	app_asynch_sleep_proc()	Η συνάρτηση είναι προσατευμένη από διακοπές οπότε γίνεται έλεγχος αν έχει γίνει κάποια διακοπή με ανάγνωση του ακροδέκτη. Επίσης γίνεται και άδειαση της FIFO του

	αισθητηρίου.
4	app_sleep_prepare_proc()
5	app_sleep_entry_proc()
6	app_sleep_exit_proc()

Πίνακας 1545: Συναρτήσεις απόφασης για ανεσταλμένη λειτουργία του συστήματος.

Σε πλήρη λειτουργία το σύστημα, για εξοικονόμηση ενέργειας απενεργοποιεί τον πυρήνα BLE και το ράδιο και ενεργοποιείται μόλις υπάρξει αίτηση για χειρισμό μηνύματος από το λειτουργικό σύστημα. Προκειμένου να κρατηθεί σε επαγρύπνηση η εφαρμογή, γίνεται χρήση ενός flag (*reporting_phase_active*) το οποίο αλλάζει τιμή (true σε false), όταν ο μετρητής *inactivity_counter* φτάσει την μέγιστη ορισμένη τιμή (*INACTIVITY_COUNTER_MAX*). Όσο το flag είναι σε κατάσταση true το σύστημα παραμένει σε κατάσταση αναμονής πραγματοποιεί αναγνώσεις και επεξεργασία των δεδομένων του γυροσκοπίου, πραγματοποιεί αποστολές πακέτων και είναι διαθέσιμο για κάθε διακοπή που μπορεί συμβεί.

Στην παρούσα κατάσταση (ΕΙΚΟΝΑ 31) το σύστημα καταναλώνει 0.56mA όταν κάνει την αποστολή των πακέτων, όπου είναι και η μέγιστη κατανάλωση, και έχει μια μέση κατανάλωση κοντά στα 59.31uA.



Εικόνα 3131: Γραφική κατανάλωσης ενέργειας ενώ το σύστημα είναι σε πλήρη λειτουργία.

Όταν ο μετρητής *inactivity_counter* φτάσει την προκαθορισμένη τιμή (*INACTIVITY_COUNTER_MAX*), εφόσον ο χρήστης δεν κάνει καμία κίνηση προκειμένου να μηδενίσει τον μετρητή, το flag (*reporting_phase_active*) αλλάζει την τιμή του σε false. Πλέον επιτρέπεται να μπει σε κατάσταση ανεσταλμένης λειτουργίας και έτσι το σύστημα κλείνει όλα τα περιφερειακά του, το ράδιο, και ο επεξεργαστής μπαίνει σε κατάσταση αναμονής. Τα μόνα τμήματα που μένουν ανοιχτά είναι η SYSRAM και ένα τμήμα του ολοκληρωμένου που πρέπει

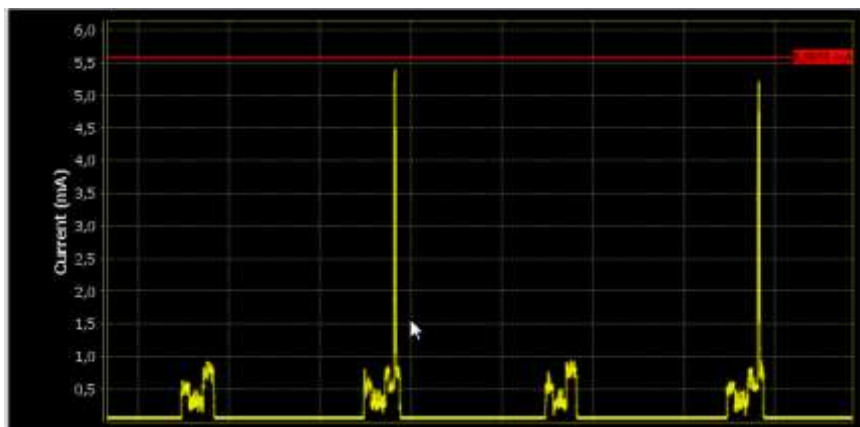
να μένει συνέχεια ανοιχτό, σ' αυτό περιλαμβάνεται ο χρονοστάθης που πρόκειται να ξυπνήσει το σύστημα την επόμενη χρονική στιγμή που θα χρειαστεί να κάνει επιβεβαίωση της σύνδεσης. Όταν έρθει η χρονική στιγμή για την επιβεβαίωση της σύνδεσης το σύστημα ξυπνά, ανοίγει όλα τα περιφερειακά και όσα συστήματα είναι ανενεργά κάνει την αποστολή και ξαναπέφτει σε κατάσταση ανεσταλμένης λειτουργίας.

Ένας ακόμα τρόπος να ξυπνήσει το σύστημα είναι να δεχθεί διακοπή από κίνηση. Σ' αυτήν την περίπτωση το σύστημα εξακολουθεί να μένει ενεργό για όσο διάστημα ορίζει το σύνολο του χρονιστή και του μετρητή, στην παρούσα υλοποίηση της εφαρμογής ο χρόνος αυτός είναι 1 sec.

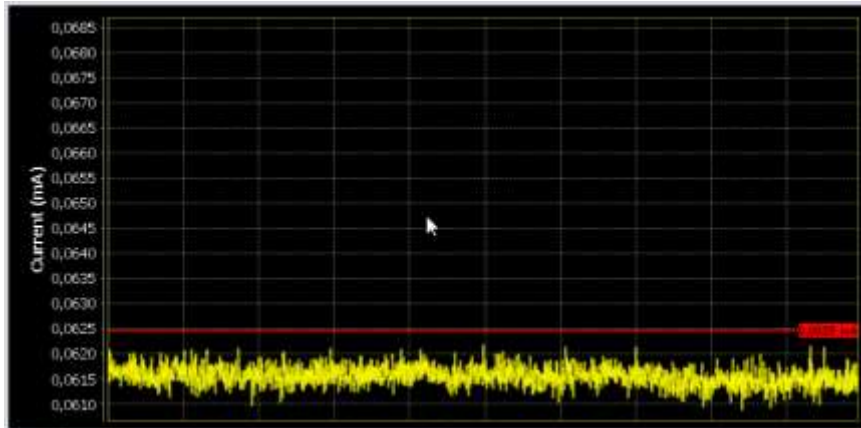
Παρακάτω, (EIKONA 32) παρατίθεται η γραφική παράσταση της έντασης ρεύματος σε συνάρτηση με τον χρόνο όπως μετρήθηκε από το εργαλείο που παρέχει η dialog για την μέτρηση κατανάλωσης ενέργειας (χωρίς την κατανάλωση ρεύματος που απαιτεί το αισθητήριο). Στην παρούσα κατάσταση το σύστημα έχει μια μέγιστη κατανάλωση ενέργειας 0.56 mA (όταν γίνεται η αποστολή του πακέτου επιβεβαίωσης σύνδεσης) και μέση κατανάλωση 12uA.

Τέλος το σύστημα στηρίζεται στον επιπλέον χρονοστάθης που αναφέρθηκε στην παράγραφο 5.6 (*APP_RING_INACTIVITY_TIMER*) προκειμένου να αποσυνδεθεί από τον Host και πέσει σε μόνιμη αναστολή λειτουργίας. Σε αυτήν την κατάσταση η εφαρμογή δεν μπορεί να ξυπνήσει από BLE γεγονότα, παρά μόνο από εξωτερική διακοπή. Όλα τα περιφερειακά, το ράδιο και ο πυρήνας BLE είναι κλειστά και τα μόνα ενεργοποιημένα τμήματα είναι η SYSRAM και ο χρονοστάθης που θα ξυπνήσει το ολοκληρωμένο κατόπιν μίας εξωτερικής διακοπής. Στην EIKONA 33 παρατίθεται η γραφική παράσταση του συστήματος ενώ αυτό βρίσκεται σε κατάσταση μόνιμης ανεσταλμένης λειτουργίας.

Πρέπει να σημειώσουμε πως σε κατάσταση μόνιμης αναστολής λειτουργίας δεν μπορεί να γίνει μέτρηση των τιμών κατανάλωσης με την χρήση του εργαλείου της dialog εφόσον οι τιμές της έντασης ρεύματος είναι εκτός των ορίων που μπορεί να μετρήσει. Οι τιμές που εμφανίζονται στην γραφική οφείλονται σε αποκλίσεις που εισάγει η αναπτυξιακή πλακέτα.



Εικόνα 3232: Γραφική κατανάλωσης ενέργειας ενώ το σύστημα είναι σε κατάσταση ανεσταλμένης λειτουργίας και επανέρχεται για αποστολή πακέτου επιβεβαίωσης σύνδεσης.



Εικόνα 3333: Γραφική κατανάλωσης ενέργειας ενώ το σύστημα είναι σε κατάσταση μόνιμης ανεσταλμένης λειτουργίας.

Κεφάλαιο 6 – Συμπεράσματα και μελλοντική έρευνα

Η παρούσα υλοποίηση της συσκευής δεν βασίζεται σε κάποιο υπάρχον προϊόν, διαθέσιμο στο καταναλωτικό κοινό. Αντίστοιχες συσκευές με την μορφή δαχτυλιδιού και με κύριο στόχο την αναγνώριση χειρονομιών από τον χρήστη, είναι υπό εξέλιξη και ο τελεολογικός σκοπός τους είναι η αναγνώριση σχεδιαστικής κίνησης και ενεργοποίησης της αντίστοιχης συσκευής ή διαδικασίας.

Η παρούσα ανάπτυξη είναι καθαρά πειραματική και η κατασκευή που υλοποιήθηκε είχε σαν πρωταρχικό στόχο την βασική και όχι την βέλτιστη λειτουργία. Τα σημεία όπου η συσκευή, στην παρούσα υλοποίηση, ενδεχομένως υστερεί είναι η σταθερότητα του κέρσορα ενώ ο χρήστης εκτελεί μια χειρονομία. Εξαιτίας του λόγου αυτού απαιτείται από τον χρήστη μεγάλη εξοικείωση με την συσκευή προτού μπορέσει να την λειτουργήσει απρόσκοπτα. Πιθανή λύση θα ήταν η τοποθέτηση του γυροσκοπίου σε διαφορετικό τμήμα του χεριού, είτε σε άλλο δάχτυλο (επιπλέον δαχτυλίδι) είτε σε κάποιο τμήμα της παλάμης (γάντι). Αυτό όμως θα επηρέαζε την εργονομία και την ευκολία της συσκευής στην τοποθέτηση καθώς και το κόστος κατασκευής. Γι' αυτό τον λόγο σε μελλοντική έκδοση της συσκευής θα προτιμηθεί να γίνει κάποια βελτιστοποίηση λογισμικού του μηχανισμού κίνησης του κέρσορα.

Όσον αφορά τον αλγόριθμο ανίχνευσης χειρονομιών, παρότι για τις υλοποιημένες κινήσεις η απόδοσή του ήταν ικανοποιητική υπήρχε μεγάλη συσχέτιση με τον προσανατολισμό του χεριού του χρήστη. Ακόμη θα πρέπει να γίνουν δοκιμές με επιπλέον χειρονομίες και κατά πόσο το σύστημα μπορεί να ξεχωρίσει περισσότερες από τρεις χειρονομίες χωρίς να τις μπερδεύει. Βελτιώσεις του αλγορίθμου χειρονομιών για το παρόν σύστημα αναφέρονται αναλυτικά στην εργασία [9] και ως προς την βελτίωση της ταχύτητας αναγνώρισης αλλά και ως προς τον αριθμό των αναγνωρίσιμων χειρονομιών. Οι προοπτικές για μελλοντική έρευνα μετά την παρούσα εργασία σχετίζονται ως επί το πλείστον με την βελτιστοποίηση του αλγορίθμου αναγνώρισης. Από εκεί και έπειτα ενδιαφέρουσες μετατροπές και προσθήκες θα μπορούσαν να μετατρέψουν την συσκευή σε κάτι άλλο, πέρα από μια απλή συσκευή εισόδου καθώς και να βελτιώσουν την αλληλεπίδραση με τον χρήστη.

Το γεγονός ότι οι χειρονομίες που μπορεί να αναγνωρίσει η εφαρμογή είναι πλήρως εξατομικευμένες, σημαίνει πως ο χρήστης θα πρέπει με κάποιο τρόπο να «διδάξει» στην συσκευή τις αναγνωρίσιμες χειρονομίες. Μια εφαρμογή όπου ο χρήστης θα μπορεί πέρα από το να εισάγει τις δικές του χειρονομίες, αλλά και να προσδίδει σ' αυτές, τις λειτουργίες που επιθυμεί, είναι σχετικά απαραίτητο για μια εμπορική έκδοση της συσκευής. Τέλος αυτό που θα μπορούσε να διευρύνει το δραστικό πεδίο πέρα από αυτό της συσκευής εισόδου είναι να υποστηρίζει κάποιες εναλλακτικές λειτουργίες. Εναλλακτικές λειτουργίες όπως να αναλάβει ρόλο τηλεχειριστηρίου πολλαπλού σκοπού για οποιαδήποτε συσκευή μέσα στο σπίτι ή οπουδήποτε.

Συνοψίζοντας, αναλύσαμε τις τεχνολογίες που χρησιμοποιήθηκαν προκειμένου να υλοποιηθεί μια συσκευή εισόδου βασισμένη σε αισθητήρες αδράνειας. Αναφερθήκαμε στο πρωτόκολλο ασύρματης σύνδεσης BLE καθώς και στην εφαρμογή του πρωτοκόλλου HID με την χρήση τεχνολογίας BLE. Έγινε μια γενική αναφορά στο υλικό που χρησιμοποιήθηκε, σχετικά με τα αισθητήρια καθώς και στο αναπτυξιακό του ολοκληρωμένου που δόθηκε για την υλοποίηση της εφαρμογής. Τέλος αναφερθήκαμε στα κύρια σημεία της αρχιτεκτονικής του συστήματος που αναπτύχθηκε. Με την παρούσα εργασία αποδείχθηκε πως η εκτέλεση πολύπλοκων και κοστοβόρων αλγορίθμων μπορεί να γίνει και να αποδώσει σε επεξεργαστές

Μεταπτυχιακή Διατριβή

Τζανιδάκης Μάριος

χαμηλών προδιαγραφών με σωστή χρήση των πόρων τους χωρίς την απαίτηση μεγάλων ποσών ενέργειας.

Βιβλιογραφία

- [1] "Device Class Definition for Human Interface Devices (HID)," 2001.
- [2] "Future Electronics," [Online]. Available: <http://www.futureelectronics.com/en/sensors/gyro.aspx>.
- [3] "Sensors Online," [Online]. Available: <http://www.sensorsmag.com/sensors/acceleration-vibration/an-overview-mems-inertial-sensing-technology-970>.
- [4] "Digi-Key," [Online]. Available: <http://www.digikey.com/en/articles/techzone/2011/apr/mems-accelerometers-gyroscopes-and-geomagnetic-sensors---propelling-disruptive-consumer-applications>.
- [5] Y. Z. M. BY YONGYAO CAI, "memsic," 2 2012. [Online]. Available: http://www.memsic.com/userfiles/files/publications/Articles/Electronic_Products_Feb_%202012_Magnetometer.pdf.
- [6] "ARGENOX," [Online]. Available: <http://www.argenox.com/bluetooth-low-energy-ble-v4-0-development/library/a-guide-to-selecting-a-bluetooth-chipset/>.
- [7] Dialog Semiconductor, "User Manual DA14580 Keyboard reference application UM-B-009," Dialog Semiconductor, 2014.
- [8] Dialog Semiconductor, "User Manual DA14580 Mouse reference application UM-B-027," Dialog Semiconductor, 2014.
- [9] Ι. Πάνου, "Βελτιστοποίηση αλγόριθμου αναγνώρισης προτύπων για εκτέλεση σε low power cru.," ΠΑ.ΠΕΙ, Πειραιάς, 2015.
- [10] C. C. A. a. R. D. Kevin Townsend, Getting Started with Bluetooth Low Energy, 1005 Gravenstein Highway North, Sebastopol, CA 95472: O'Reilly Media, Inc, 2014.
- [11] Dialog Semiconductor, "DA14580 Low Power Bluetooth Smart SoC," Dialog Semiconductor, 2014.
- [12] Dialog Semiconductor, "DA14580 Sleep mode configuration UM-B-006," Dialog Semiconductor, 2014.
- [13] Dialog Semiconductor, "DA14580 Software architecture UM-B-015," Dialog Semiconductor, 2014.

- [14] Bluetooth SIG, "HID Over GATT Profile Specification," Bluetooth SIG, 2011.
- [15] InveSense Inc, "MPU-6000 and MPU-6050 Product Specification," InveSense Inc, 2011.
- [16] "Quora," [Online]. Available: <http://www.quora.com/How-does-a-MEMS-gyroscope-work>.
- [17] Dialog Semiconductor, "DA14580 Development Kit UM-B-014," Dialog Semiconductor, 2014.

ΠΑΡΑΡΤΗΜΑ Α : User Manual DA14580 Gesture Ring

This appendix describes the Gesture Ring application based on the DA14580. The Gesture Ring application runs on a ring shaped hardware which is equipped with a GY-88 IMU sensor. The schematic and the connections to the Dialogs Expert Development Kit are also provided in chapter 2.

The User Manual is provided with two .rar files `gesture_ring.rar` and `SDK_v_3.0.2.0_Gesture_recon.rar`. The first .rar file contains the source code for the Gesture Ring mouse and the latter contains an auxiliary code for template isolation and testing.

The auxiliary source code is used for isolating and testing the gestures that will be used as templates, and provides the amount of time that the program spends executing the gesture recognition algorithm. The functions and explanation of the Gesture Ring mouse source code is given in chapter 3 and the examples and settings for the auxiliary code are provided in chapter 4.

The source code for the application, explained in chapter 3, is provided in the `gesture_ring.rar` file. Is based on the DA1458x_SDK_3.0.6 Software Development Kit. The developer should extract the .rar file and place the extracted contents (three folders) in the directories explained below:

- The `gesture_ring` folder is the application source code. This folder should be placed in the `"\DA1458x_SDK_3.0.6\DA1458x_SDK_3.0.6\dk_apps\keil_projects\"` directory.
- The `i2c` folder is a modified version `i2c_eeprom` driver. This folder should be placed in the `"\DA1458x_SDK_3.0.6\DA1458x_SDK_3.0.6\dk_apps\src\plf\refip\src\driver"` directory.
- The `mpu_6050` folder includes the driver for the sensor and also should be placed in the `"DA1458x_SDK_3.0.6\DA1458x_SDK_3.0.6\dk_apps\src\plf\refip\src\driver"` directory.

The user, after placing the files in previously mentioned directories, can open the `\DA1458x_SDK_3.0.6\DA1458x_SDK_3.0.6\dk_apps\keil_projects\ gesture_ring\gesture_ring\gesture_ring.uvproj` file using keil compiler. **Note: The program runs in O2 optimization.**

The `SDK_v_3.0.2.0_Gesture_recon.rar` file, explained in chapter 4, includes the auxiliary code for gesture isolation and testing. Is based on the `SDK_v_3.0.2.0` software development kit. The user should extract the .rar file and open the `SDK_v_3.0.2.0_Gesture_recon\plain_gesture_recon_alg \peripheral_examples\DA14580_peripheral_setup.uvproj` file using keil compiler.

ΠΑΡΑΡΤΗΜΑ Β : Hardware Overview

The application for the acquisition of inertia data uses the GY-88 chip which consists from three different sensors:

- HMC5883L – Digital pressure sensor
- BMP085 – 3Axis Digital Compass
- MPU6050 – Accelerometer and Gyroscope

The application only uses the MPU6050 to acquire acceleration and angular velocity readings. The sensor is attached on a copper board where the I2C circuit and the battery are located. Figure 1 shows the schematic and the connections with the Dialogs Expert Development Kit. The J2 header shows where the GY-88 is placed.

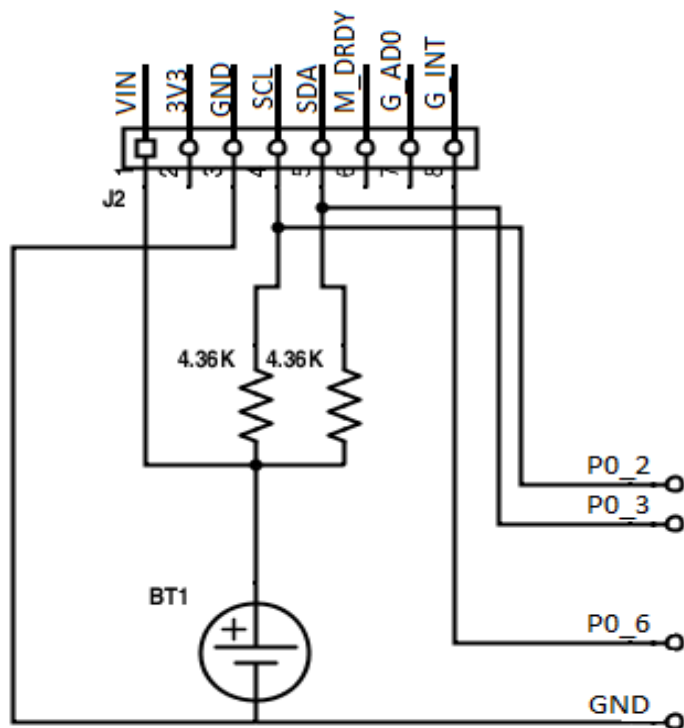


Figure 1

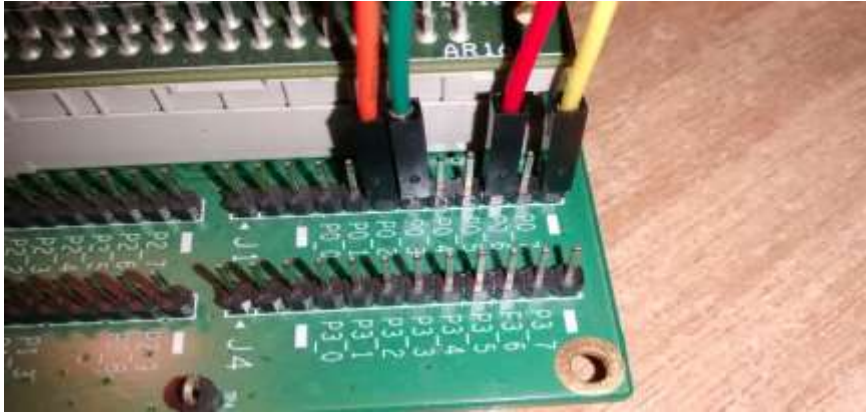


Figure 2

ΠΑΡΑΡΤΗΜΑ Γ : Application Source Code

In this chapter the Gesture Ring mouse source code is explained. All the functions below and the .uvproj file can be found in \DA1458x_SDK_3.0.6\DA1458x_SDK_3.0.6\dk_apps\keil_projects\gesture_ring\gesture_ring directory.

1. File `gst_ring_config.h` contains all the possible configurations that are available to the programmer.

Sensor configurations

ACCEL_FS_SEL	selects the full scale range of the gyroscope outputs										
	<table border="1"> <thead> <tr> <th>AFS SEL</th> <th>Full scale range</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>±2 g</td> </tr> <tr> <td>1</td> <td>±4 g</td> </tr> <tr> <td>2</td> <td>±8 g</td> </tr> <tr> <td>3</td> <td>±16 g</td> </tr> </tbody> </table>	AFS SEL	Full scale range	0	±2 g	1	±4 g	2	±8 g	3	±16 g
AFS SEL	Full scale range										
0	±2 g										
1	±4 g										
2	±8 g										
3	±16 g										
GYRO_FS_SEL	selects the full scale range of the gyroscope outputs										
	<table border="1"> <thead> <tr> <th>FS SEL</th> <th>Full scale range</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>±250 °/sec</td> </tr> <tr> <td>1</td> <td>±500 °/sec</td> </tr> <tr> <td>2</td> <td>±1000 °/sec</td> </tr> <tr> <td>3</td> <td>±2000 °/sec</td> </tr> </tbody> </table>	FS SEL	Full scale range	0	±250 °/sec	1	±500 °/sec	2	±1000 °/sec	3	±2000 °/sec
FS SEL	Full scale range										
0	±250 °/sec										
1	±500 °/sec										
2	±1000 °/sec										
3	±2000 °/sec										
SENSOR_DLPF	configures the digital low pass filter for accelerometer and gyroscope										
ACCEL_DHPF	configures the accelerometer digital high pass filter										
SMRT_DIV	configures the sampling rate Sample Rate = Accelerometer Output Rate / (1 + SMPRT_DIV), Accelerometer Output Rate=1kHz										
MOT_THR_SET	motion interrupt threshold										
MOT_DUR_SET	Number of values that must exceed the motion threshold in order to trigger the interrupt										
MOT_DECR_RATE	Motion detection counter decrement rate set in 1										

Timers configurations

INACTIVITY_TIMER_QUOTA	Inactivity time (in multiples of 10msec) before the device goes into permanent sleep
INACTIVITY_TIMER_COUNTER_MAX	Time in multiples of INACTIVITY_TIMER_QUOTA
INACTIVITY_COUNTER_MAX	The time that we will stay in active mode after the last movement in multiples of 10msec
MOVE_INTERVAL	Sampling time for gyroscope

Mouse movement configurations

MOVE_OFFSET	The difference a current gyroscope reading must have from the previous reading in order to consider that the mouse has moved
GYRO_WINDOW	The length of the buffer that keeps successive values from the gyroscope

FIFO and Gesture Recognition configurations

DUMMY_READS_MAX	The number of data that has to be in fifo in order to empty it
FIFO_GST_BEF	Datasets in fifo before the read burst
DATA_LEFT_IN_FIFO	Datasets that are left in fifo after every read burst *The values that enter the fifo after the motion interrupt has triggered are part of the gesture, but usually the gesture begins 4-6 datasets before the motion interrupt is triggered. That's why we always keep the last 6 elements of the fifo when we perform burst reads in order to empty it (the fifo)
MEAN_WINDOW	Mean window size in datasets *In order to find the end of the gesture we need to check the variance of MEAN_WINDOW_SIZE samples
MEAN_WIN_THR	Threshold used for isolating the gesture
MAX_GST_LEN_INIT_VAL	Initial value of measurements when starting the gesture recognition algorithm
MAX_GST_SIZE	The max length a valid gesture can have
VARIANCE_COUNT_UNDER_THR	Count the variance values under the threshold that indicate the end of a gesture

2. File app_ring_fsm.h:

Function name	start adv directed
inputs	void
Return value	void
description	Starts the process of directed advertising

Function name	app state update
inputs	Evt: an event that will cause a state change
Return value	void
description	Function that handles the main finite state machine of Gesture Ring

Function name	app_rng_init_vars
inputs	void
Return value	void
description	Reinitializes the variables used in the FSM

3. File gesture_recon.c:

Function name	interrupt_events
inputs	void
Return value	gst_recon_events: the type of interrupt that was triggered
description	This function executes every time an interrupt occurs and returns to us the interrupt that was triggered

Function name	gestureRecon
inputs	Evt:the interrupt that was triggered
Return value	void
description	This function is the gesture recognition finite state machine

Function name	variance
inputs	Buffer: array containing the 4 elements whose variance we want to compute
Return value	The computed variance
description	Function that computes the variance for the z axis of 4 values

Function name	DTWSakoeImp
inputs	input: array containing the samples that we want to compare to the templates using DTW input_size: the number of samples we want to compare calc_first_row: boolean expressing whether it's the first row that we give as input or not
Return value	void
description	Function that computes the dynamic time warping cost of the provided test template samples when compared to the 3 templates

Function name	min
inputs	a: the cost generated from DTW while comparing the unknown gesture to click template b: the cost generated from DTW while comparing the unknown gesture to press template c: the cost generated from DTW while comparing the unknown gesture to release template
Return value	void
description	This function decides what template is more similar to our unknown gesture and if it is considered valid gesture (the cost is below a threshold) adds this gesture into the gesture buffer

4. File `gst_template.h`:

This file contains the templates that we use in order to identify the unknown gesture. In case that we change one of these templates, we have to change the size of it so that it corresponds to the new template size(e.g. `RELEASE_SIZE_TEMPL`).

5. File `app_reports.c`

Function name	<code>send_mouse_report</code>
inputs	<code>void</code>
Return value	1 if everything went alright, 0 otherwise
description	Function that will update the attribute <code>db</code> with the new values that we want to send

Function name	<code>ring_process_gesture</code>
inputs	<code>Gst</code> : the gesture that was performed
Return value	1 if everything went alright, 0 otherwise
description	Function that will take the performed gesture, create a report and push it to <code>kbd_trm_list</code>

Function name	<code>ring_process_motion_8</code>
inputs	<code>move</code> : the movement of the cursor
Return value	1 if everything went alright, 0 otherwise
description	Function which takes the movement of the cursor, create a report and push it to <code>kbd_trm_list</code>

Function name	<code>app_gst_prepare_reports</code>
inputs	<code>void</code>
Return value	1 if everything went alright, 0 otherwise
description	Function which takes each item in the <code>gst_buffer</code> and create an appropriate report for each one of the gestures in order to be sent

6. File `mpu6050_func.c`

Function name	<code>app_initialize_sensor</code>
inputs	<code>void</code>
Return value	<code>void</code>
description	Function that initialize the sensor when the app will start

Function name	<code>app_disable_sensor</code>
inputs	<code>void</code>
Return value	<code>void</code>
description	Function that disables the fifo and set only the motion interrupt active

Function name	<code>app_reinit_sensor</code>
inputs	<code>void</code>
Return value	<code>void</code>
description	Function that will re-initialize the sensor after we wake up and re-connect to the host

Function name	severFIFOData
inputs	Buffer_dst: the destination buffer the data will be put in Buffer_src: the source buffer that contains the data Size: the size of the buffer in bytes Data_sel: select whether accelerometer or gyroscope data
Return value	void
description	Function that will take the FIFO data and isolate only accelerometer or only gyroscope data according to data sel value

Function name	mean value
inputs	Buffer: the buffer that contains the values we want to find the mean value of
Return value	The mean value
description	Function that will find the mean value of the elements of the given buffer

Function name	getGyroData
inputs	void
Return value	The gyroscope values
description	Function that will read the gyroscope values from the register and return the higher 8-bits of the gyroscope value *since the gyroscope values are 16-bit and we need to map them to an 8-bit number, we perform this mapping by using only the 8-high bits of the value

7. File mpu6050.c

Function name	setClockSource
inputs	Source: the clock source of the sensor
Return value	void
description	Function that sets the clock source of the sensor

Function name	sensorReset
inputs	void
Return value	void
description	Function that resets the sensor

Function name	setSensorSleepMode
inputs	void
Return value	void
description	Function that configures the power mode

Function name	setDLPFMode
inputs	Mode: the digital low pass filter mode
Return value	void
description	Function that configures the digital low pass filter

Function name	setRate
inputs	Rate: the desired sampling rate
Return value	void
description	Function that configures the rate used to put the data into fifo

Function name	setGyroFullRange
inputs	Mode: selects the full scale range of the Gyroscope
Return value	void
description	Function that configures the full scale range of the gyroscope sensor

Function name	setAccelFullRange
inputs	Mode: selects the full scale range of the Accelerometer
Return value	void
description	Function that configures the full scale range of the accelerometer sensor

Function name	setMotionDetectionThreshold
inputs	Threshold: the desired threshold used to trigger motion interrupt
Return value	void
description	Function that configures the threshold used in order to trigger the motion interrupt

Function name	setMotionDetectionDuration
inputs	duration: the desired duration used to trigger motion interrupt
Return value	void
description	Function that configures the number of values that must exceed the motion threshold in order to trigger the interrupt

Function name	setMotionDetectionCountDecrement
inputs	decrement: the desired motion detection counter decrement rate
Return value	void
description	Function that configures the Motion detection counter decrement rate

Function name	setAccelDHPFMode
inputs	mode: the desired mode for digital high pass filter
Return value	void
description	Function that configures the digital high pass filter of the accelerometer

Function name	setInterruptConfiguration
inputs	Int cfg: the enabled interrupt configuration
Return value	void
description	Function that configures interrupts that are enabled

Function name	setEnabledInterrupts
inputs	Int en: the interrupts we want to enable
Return value	void
description	Function that enables the desired interrupts
Function name	setFIFOData
inputs	Fifo_data: the data that we want stored in the fifo
Return value	void
description	Function that configures what data (accel or gyro) will be stored in the fifo
Function name	enableFIFO
inputs	void
Return value	void
description	Function that enables the fifo in order to store data
Function name	disableFIFO
inputs	void
Return value	void
description	Function that disables the fifo
Function name	countFIFO
inputs	void
Return value	The total bytes of the datasets stored in fifo
description	Function that returns the total bytes of the datasets stored in fifo
Function name	resetFIFO
inputs	void
Return value	void
description	Function that resets the fifo
Function name	readFIFO
inputs	Buffer: the buffer that will hold the datasets we read from FIFO Size: the number of bytes we want to read from fifo
Return value	void
description	Function that reads as many bytes we want from the fifo and stores them in the provided buffer

8. File app_sleep.h

Function name	app_async_trm
inputs	void
Return value	true, to force calling off schedule else false
description	In app_async_trm function, we prepare the reports that we want to send. Also, we set APP_RING_REPORT_SYNCHRONIZATION_TIMER, a timer that is used to synchronize the cursor reports and also to keep the device awake.

Function name	app_async_proc
inputs	void
Return value	true, to force calling of scedule false, else
description	In app_async_proc function, every time we have a mouse_report_synchronization_event_triggered event, we poll the gyroscope and put the values in a buffer that can contain 8 elements. We do this because as we stated earlier, by the moment the motion interrupt triggers, 4-6 samples (the start of the gesture) have already occurred. If these values are polled just before the motion interrupt triggers, then there is a high chance that a mouse cursor move report will be sent moving the cursor away from the actual point where the user wants to perform the gesture. So, every time we poll the gyroscope, the values read will be written in the next empty space of the buffer. At the moment all 8 spots are full, we will take the average of the 4 older values and compare the produced average with the previous cursor movement we sent earlier. If there is a difference between them, we consider that the cursor has moved and we store the cursor movement in trm_list in order to be sent. As it is expected, if a motion interrupt occurs, the contents of this buffer are cleared.

Function name	app_async_sleep_proc
inputs	void
Return value	void
description	In app_async_sleep_proc function, we check the pin to make sure we didn't miss any interrupt. Also, in case we are not in gesture recognition mode, we empty all the data in the FIFO except the last 6, so that it doesn't get overflow.

Function name	app_sleep_prepare_proc
inputs	sleep mode: the sleep mode rwip sleep returned
Return value	void
description	In app_sleep_prepare_proc function, we check the pin to make sure we didn't miss any interrupt. Also, if the device is in active mode, we prevent it from going to sleep. Finally, if going to sleep is inevitable, we set the wakeup interrupt service routine.

Function name	<code>app_sleep_entry_proc</code>
inputs	<code>sleep mode: the sleep mode rwip sleep returned</code>
Return value	<code>void</code>
description	<code>in app_sleep_entry_proc function, we check one last time the pin to make sure we don't miss any interrupt, and use lower clocks.</code>

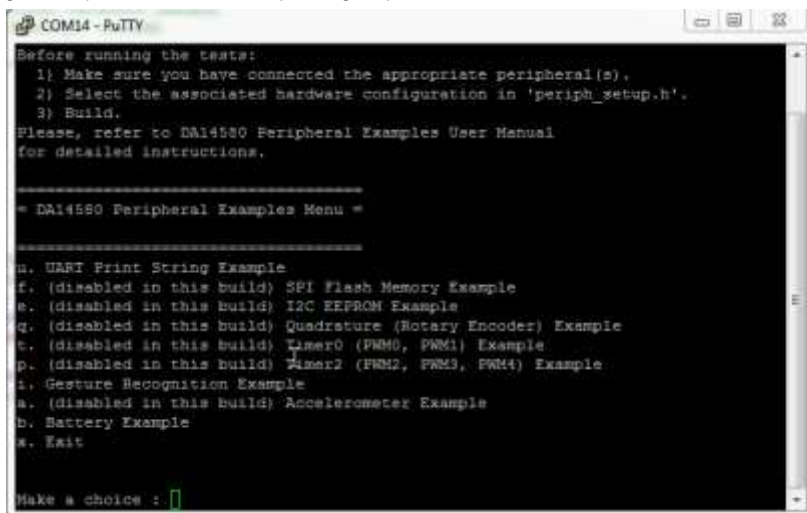
ΠΑΡΑΡΤΗΜΑ Δ : Auxiliary Source

In this appendix the auxiliary source code is explained the .uvproj file should be located in SDK_v_3.0.2.0_Gesture_recon\plain_gesture_recon_alg\peripheral_examples\DA14580_peripheral_setup.uvproj. The specific program is used in order to acquire and test the templates that are used as reference to recognize the gestures that the user performs. The program is based on dialog's peripheral_examples source code. The main section of the program is located in the DA14580_examples.c. The code initializes the IMU sensor, sets and enables the motion interrupt and enters a while loop where it keeps polling an interrupt variable and then empties the FIFO of the sensor in order to avoid overflow. If a motion interrupt occurs then the motion interrupt is deactivated and the program enters the gesture recognition mode. As soon as the gesture recognition finishes, the program reactivates the motion interrupt and waits for the next gesture to occur.

An additional choice is added in the HARDWARE_CONFIGURATION_INDEX definition in periph_setup.h file. By setting the HARDWARE_CONFIGURATION_INDEX to 4 the TRYOUT_MPU6050_EXAMPLE is defined in order to make the appropriate settings for the MPU6050 sensor. The user then can set the APP_MODE definition to one of the three different modes:

- TIME_MEASURE: prints the execution time of the gesture recognition algorithm.
- TEMPLATE_SELECTION: prints accelerometer values after a completed gesture.
- TEMPLATE_TESTING: prints recognized gestures to test the accuracy of the templates.

An extra option is added "Gesture Recognition Example" which is triggered by pressing "j" in the example menu on the console (Figure 3). Depending on the value of APP_MODE definition the program responds with the corresponding outputs.



```
COM14 - PuTTY
Before running the tests:
 1) Make sure you have connected the appropriate peripheral(s).
 2) Select the associated hardware configuration in 'periph_setup.h'.
 3) Build.
Please, refer to DA14580 Peripheral Examples User Manual
for detailed instructions.

=====
= DA14580 Peripheral Examples Menu =
=====
u. UART Print String Example
f. (disabled in this build) SPI Flash Memory Example
e. (disabled in this build) I2C EEPROM Example
q. (disabled in this build) Quadrature (Rotary Encoder) Example
t. (disabled in this build) Timer0 (PWM0, PWM1) Example
p. (disabled in this build) Timer2 (PWM2, PWM3, PWM4) Example
j. Gesture Recognition Example
a. (disabled in this build) Accelerometer Example
b. Battery Example
e. Exit

Make a choice : [j]
```

Figure 3

1. TIME MEASUREMENT

In the **TIME_MEASURE** option the program uses the timer in order to count how long the program executes the gesture recognition algorithm every time an interrupt occurs (dataready or motion interrupt).

The program (`void enableTimer(void)`) enables `TIMER0`, sets the clock of the timer with division factor 1 (16MHz) and loads the value 65000. When the timer elapses the interrupt is triggered and a counter increases (`void user_callback_function_test(void)`). The value of the counter and the remaining value in the timer are printed every time we exit the ISR in order to calculate the time period the program spent in the gesture recognition algorithm. The output of the program is shown in Figure 4.

```
COM14 - PuTTY
Make a choice : 0000
EB1D
0001
E2DA
0000
F2E2
0000
3A0D
0000
F2E9
0000
F2E9
0000
F2E9
0000
259D
0000
F2E9
0000
F2E9
0000
F2E9
0000
2619
```

Figure 4

2. TEMPLATE SELECTION

In the **TEMPLATE_SELECTION** option an additional buffer (print_buffer[60]) is added to the program. All the accelerometer values that the program reads from the sensor (while a gesture is detected) are copied in this buffer. As soon as the program detects the end of a gesture it prints the buffer (Figure 5). The values from the sensor appear in columns in x – y – z order.

```

Make a choice : E902, 0B54, 3D2A
E9D2, 0C0E, 3D4C
EE04, 15FC, 3824
FBA0, 25CE, 3752
1416, 4262, 20A4
1F9A, 5782, F8A4
20A2, 345C, A30A
D7EE, B9D6, D914
8000, 8000, 7FFF
BB48, 8000, 7FFF
E9C6, 992C, 5926
BB82, E0D4, 33DE
FB10, 0546, 1E80
0738, 016C, 2F92
EC24, D39A, 3CEC
F158, BD84, 3F0A
EEC6, DF66, 35EE
E278, E65A, 3088
E034, CC56, 363A
DAB2, CC2A, 3172
E8E6, E87C, 20B2
F0C0, 0D78, 1C8C
E728, 2290, 289E
DEB8, 09F8, 4660
D514, C69A, 534E
D31C, AD62, 4374
DC5A, C0E0, 3CD8
F472, EF7A, 214C
06A8, 1ABA, 25FA
00F8, 22EC, 2BD2
EEB8, 138A, 3384
DD1E, 083C, 3A68
E1BC, 0A00, 3E22
E156, FCDC, 3BC0
EA04, 0D5C, 3E3C
EE92, 0BAA, 3F8C
E8B2, 04D6, 3D7E
E38C, 0ADC, 3C82
E6BA, 0FFC, 39FA
E8E2, 0D62, 395C
E3FA, 09B2, 39C6
E5B0, 0CC8, 3952
EA5E, 0B0C, 39A2
E9A6, 0A1E, 3A66
E830, 0A60, 3A22
EAB0, 0CCC, 3986
0000, 0000, 0000
0000, 0000, 0000
  
```

Figure 5

The user can visualize the data using an excel program. The image below shows 3 clicks, 1 press and 1 release performed.

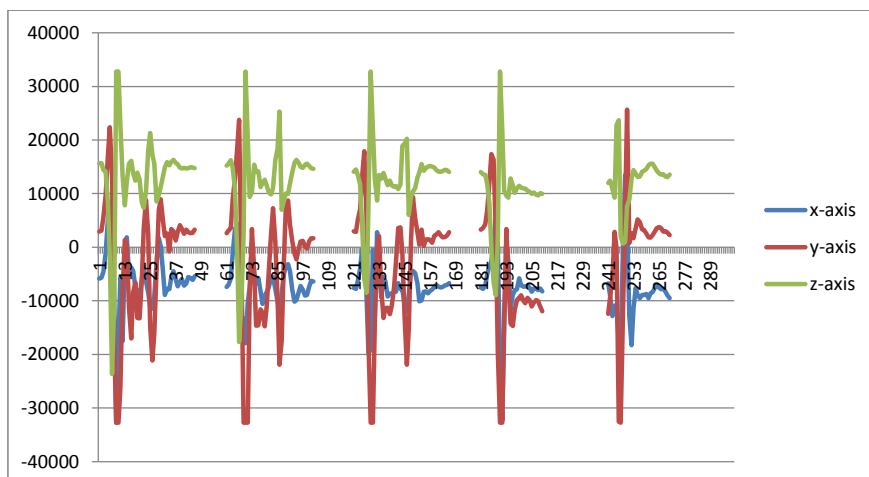


Figure 6

3. TEMPLATE TESTING

In the **TEMPLATE_TESTING** option the user can perform gestures and test if the gesture templates are sufficiently recognized by the program. The user templates are located in the `gst_templates.h` file along with the corresponding array sizes.



Figure 7

Also the user can configure an upper cost sensitivity in the `gesture_recon.h` file. With the upper cost sensitivity the user can define the maximum cost value the program should accept as a valid gesture. In Figure 6, gestures are tested against the current templates, defined in `gst_templates.h` file. The “no” tag indicates that the gesture has greater cost value than any of upper cost values so the algorithm doesn’t recognize it as a valid gesture. The “to large” tag indicates that the gesture is larger than 60 samples.

ΠΑΡΑΡΤΗΜΑ Ε : Auxiliary Source Code Configurations

In periph_setup.h file the user can also set:

- The motion interrupt condition definitions:
 - MOT_THR_SET: acceleration threshold.
 - MOT_DUR_SET: number of values over the MOT_THR_SET threshold.
 - MOT_DECR_RATE: decrement rate of non qualified threshold sample.
- The FIFO's storing data rate:
 - SMRT_DIV
- The accelerometer and gyroscope range settings:
 - ACCEL_FS_SEL: accelerometer range.
 - GYRO_FS_SEL: gyroscope range.
- The Digital Low Pass Filter for gyroscope and accelerometer:
 - SENSOR_DLPF

In the gesture_recon.c file the user can set:

- The conditions met in order to properly isolate a gesture:
 - MEAN_WIN_THR: the variance value under which we consider that a gesture has ended.
 - VARIANCE_COUNT_UNDER_THR: every time the value of variance is under the MEAN_WIN_THR a counter increases. If the counter reaches VARIANCE_COUNT_UNDER_THR (a number of successive variance values under MEAN_WIN_THR) we consider that the gesture has ended.

ΠΑΡΑΡΤΗΜΑ ΣΤ : Auxiliary Source code Files

In the current appendix the most important functions from file `MPU_6050.c` that have been used in the source code are being described. The file contains functions mainly related to the MPU sensor.

Function Name	<code>MPU_Initialize</code>
Inputs	<code>Void</code>
Return Value	<code>Void</code>
Description	The function Initializes the mpu sensor, sets the range of the gyroscope and accelerometer, sets the fifo update rate and the digital low pass filter.
Function Name	<code>enable_MPU_Interrupts</code>
Inputs	<code>Void</code>
Return Value	<code>Void</code>
Description	The function sets and enables the interrupts on the sensor. It configures the motion interrupt, sets the accelerometer high pass filter and enables the motion and the FIFO overflow interrupt.
Function Name	<code>ring_enable_kbd_irq</code>
Inputs	<code>Void</code>
Return Value	<code>Void</code>
Description	Enables the interrupts on the processor and registers the callback function.
Function Name	<code>MPU_FIFO_Initialize</code>
Inputs	<code>Void</code>
Return Value	<code>Void</code>
Description	Configures the FIFO to store accelerometer data and enables the FIFO.
Function Name	<code>MPU_FIFOReset</code>
Inputs	<code>Void</code>
Return Value	<code>Void</code>
Description	Resets the FIFO
Function Name	<code>enableTimer</code>
Inputs	<code>Void</code>
Return Value	<code>Void</code>
Description	Sets, enables and registers the callback function for the timer when the software is in <code>TIME_MEASUREMENT</code> mode.
Function Name	<code>FIFOCount</code>
Inputs	<code>Void</code>
Return Value	<code>Uin16_t</code>
Description	Returns the number of valid data in the FIFO.

The majority of all functions related to the gesture recognition algorithm are identical to the functions used in the application software that were explained in chapter 3.