



ΕΘΝΙΚΟ ΚΑΙ ΚΑΠΟΔΙΣΤΡΙΑΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

**ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ**

**ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ
"ΣΥΣΤΗΜΑΤΑ ΕΠΙΚΟΙΝΩΝΙΩΝ ΚΑΙ ΔΙΚΤΥΑ"**

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Υλοποίηση Ευφώνων Αισθητήρων με το Πρότυπο IEEE 1451

Νικόλαος Κ. Σιλβέστρος

Επιβλέποντες: **Ευστάθιος Χατζηευθυμιάδης, Επίκουρος Καθηγητής**
Βασίλειος Τσέτσος, Διδάκτωρ

ΑΘΗΝΑ

ΟΚΤΩΒΡΙΟΣ 2010

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Υλοποίηση Ευφών Αισθητήρων με το Πρότυπο IEEE 1451

Νικόλαος Κ. Σιλβέστρος

A.M.: M887

ΕΠΙΒΛΕΠΟΝΤΕΣ: Ευστάθιος Χατζηευθυμιάδης, Επίκουρος Καθηγητής
Βασίλειος Τσέτσος, Διδάκτωρ

ΕΞΕΤΑΣΤΙΚΗ ΕΠΙΤΡΟΠΗ: Αλωνιστιώτη Αθανασία, Λέκτορας

ΟΚΤΩΒΡΙΟΣ 2010

ΠΕΡΙΛΗΨΗ

Όπως όταν πρωτοεμφανίστηκε στη γη ο άνθρωπος, χρησιμοποίησε διάφορα εργαλεία, όπως το τσεκούρι σαν προέκταση του χεριού του, έτσι και σήμερα μπορούμε να θεωρήσουμε ότι οι αισθητήρες χρησιμοποιούνται σαν επέκταση των αισθήσεών μας. Δεν αυξάνουν μόνο τις δυνατότητές μας στην αίσθηση και την αντίληψη του εξωτερικού μας περιβάλλοντος, αλλά με τη χρήση της υπάρχουσας υποδομής της παγκόσμιας δικτύωσης, μπορούμε με μεγάλη ευκολία να παρατηρούμε τις συνθήκες και τα γεγονότα που συμβαίνουν σε οποιοδήποτε μέρος του κόσμου, σαν να βρισκόμασταν εκεί.

Η συγκεκριμένη εργασία ασχολείται κυρίως με τους ευφυείς αισθητήρες και το κυριότερο πρότυπο για αυτούς που είναι το IEEE 1451, για το οποίο γίνεται εκτενής περιγραφή. Επίσης γίνεται αναφορά σε σχετικές τεχνολογίες, όπως είναι ο ιστός αισθητήρων και το βασικότερο πρότυπο που είναι το Sensor Web Enablement (SWE). Τέλος υπάρχει η περιγραφή της υλοποίησης ενός πρωτοτύπου βασισμένο στο IEEE 1451 με τη χρήση των SunSPOTs, τα οποία είναι μικρές συσκευές οι οποίες έχουν ενσωματωμένους αισθητήρες και προσφέρουν εύκολους τρόπους επικοινωνίας αλλά και προγραμματισμού τους.

Το κύριο αποτέλεσμα της συγκεκριμένης εργασίας είναι η ολοκληρωμένη και λειτουργική υλοποίηση του πρωτοτύπου, με το οποίο διαπιστώθηκε η ορθότητα του προτύπου αλλά και έθεσε τις βάσεις για ανάπτυξη μεγαλύτερων και πολυπλοκότερων συστημάτων τα οποία θα μπορέσουν να το χρησιμοποιήσουν.

ΘΕΜΑΤΙΚΗ ΠΕΡΙΟΧΗ: Ευφυείς αισθητήρες

ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ: IEEE 1451, δίκτυα αισθητήρων, ιστός αισθητήρων, SWE, SunSPOT

ABSTRACT

Just like when human first appeared on earth, he used various tools (e.g. axe) as an extension of his hand, we can assume that the sensors can be used nowadays as an extension of our senses. Not only do they increase our capability of sensing and our perception of the external environment, but we can also readily observe conditions and events that occur anywhere in the world, as if we were physically present by using the existing infrastructure of the global network.

This thesis mainly deals with smart sensors and the main standard for them which is the IEEE 1451. Related technologies are also mentioned, such as sensor web and its main standard which is the Sensor Web Enablement (SWE). Finally there is the description of a prototype implementation based on the IEEE 1451 using SunSPOTs, which are small devices that have built-in sensors and provide easy ways of communication and application programming.

The main contribution of this thesis is the complete and operational implementation of the prototype. Thus, we assessed the correctness of the standard and laid the foundations for the development of larger and more complex systems which can be based on it.

SUBJECT AREA: Smart sensors

KEYWORDS: IEEE 1451, sensor networks, sensor web, SWE, SunSPOT

ΠΕΡΙΕΧΟΜΕΝΑ

ΠΡΟΛΟΓΟΣ	11
1. ΕΥΦΥΕΙΣ ΑΙΣΘΗΤΗΡΕΣ (SMART SENSORS)	12
1.1 Εφαρμογές	13
1.2 Πλεονεκτήματα	14
1.3 Μειονεκτήματα	15
1.4 Κόμβοι αισθητήρων	16
2. ΙΣΤΟΣ ΑΙΣΘΗΤΗΡΩΝ (SENSOR WEB)	17
2.1 SWE (Sensor Web Enablement)	19
2.1.1 Πρότυπα.....	20
2.1.2 Αρχιτεκτονική.....	21
2.1.3 Sensor Model Language (SensorML).....	22
2.1.4 Transducer Markup Language (TML)	23
2.1.5 Sensor Observation Service (SOS).....	23
2.1.6 Sensor Planning Service (SPS)	24
2.1.7 Sensor Alert Service (SAS)	26
2.1.8 Web Notification Services (WNS).....	27
2.1.9 Πλεονεκτήματα	27
2.1.10 Ανακάλυψη αισθητήρων	28
2.2 OSWA (Open Sensor Web Architecture)	29
2.2.1 Αρχιτεκτονική.....	29
3. Η ΟΙΚΟΓΕΝΕΙΑ ΠΡΟΤΥΠΩΝ IEEE 1451	32
3.1 Ιστορική αναδρομή	33
3.2 Πρότυπα.....	34
3.2.1 IEEE 1451.0.....	34
3.2.2 IEEE 1451.1.....	36
3.2.3 IEEE 1451.2.....	36
3.2.4 IEEE 1451.3.....	37
3.2.5 IEEE 1451.4.....	38

3.2.6	IEEE 1451.5.....	39
3.2.7	IEEE 1451.6.....	39
3.2.8	IEEE 1451.7.....	39
3.3	Λειτουργία	40
3.4	TEDS	41
3.4.1	Πλεονεκτήματα	41
3.4.2	Δομή.....	42
3.4.3	Κατηγορίες	42
3.5	Είδη μορφοτροπέων	46
3.6	Αρχιτεκτονική.....	46
3.7	Τρόποι πρόσβασης.....	50
3.8	Ενδιαφέροντα στοιχεία.....	52
3.8.1	Μονάδες μέτρησης.....	52
3.8.2	Εντολές/Απαντήσεις	53
3.9	Πλεονεκτήματα	54
3.10	Μειονεκτήματα	55
3.11	Εφαρμογές	56
4.	ΥΛΟΠΟΙΗΣΗ ΠΡΩΤΟΤΥΠΟΥ.....	57
4.1	SunSPOT	57
4.2	Αρχιτεκτονική.....	60
4.3	Υλοποίηση.....	62
4.3.1	Μέρη	62
4.4	Λεπτομέρειες για Ειδικά Μέρη της Υλοποίησης.....	63
4.4.1	Ανακάλυψη των TIM	63
4.4.2	Χαμηλού επιπέδου επικοινωνία.....	64
4.4.3	Υλοποίηση TEDS.....	65
4.4.4	Αποθήκευση TEDS	66
4.4.5	Κωδικοποίηση/Αποκωδικοποίηση	66
4.4.6	Διάβασμα τιμών αισθητήρων.....	67
4.4.7	Αποσφαλμάτωση	67

4.5	Εγκατάσταση.....	68
4.6	Προσθήκη αισθητήρα	68
5.	ΣΥΜΠΕΡΑΣΜΑΤΑ ΚΑΙ ΑΝΟΙΧΤΑ ΘΕΜΑΤΑ.....	69
5.1	Συμπεράσματα.....	69
5.2	Ανοιχτά θέματα.....	69
	ΠΙΝΑΚΑΣ ΟΡΟΛΟΓΙΑΣ	71
	ΣΥΝΤΜΗΣΕΙΣ – ΑΡΚΤΙΚΟΛΕΞΑ – ΑΚΡΩΝΥΜΙΑ	72
	ΑΝΑΦΟΡΕΣ	74

ΚΑΤΑΛΟΓΟΣ ΣΧΗΜΑΤΩΝ

Σχήμα 1: Δομή ευφυούς αισθητήρα.....	13
Σχήμα 2: Δομή SWE.....	21
Σχήμα 3: Αρχιτεκτονική SWE.....	22
Σχήμα 4: Αρχιτεκτονική του SOS.....	24
Σχήμα 5: Αρχιτεκτονική SPS	25
Σχήμα 6: Αρχιτεκτονική SAS	26
Σχήμα 7: Αρχιτεκτονική WNS	27
Σχήμα 8: Αρχιτεκτονική OSWA.....	30
Σχήμα 9: Αρχιτεκτονική του IEEE 1451	34
Σχήμα 10: Αρχιτεκτονική του IEEE 1451.1	36
Σχήμα 11: Αρχιτεκτονική του IEEE 1451.2	37
Σχήμα 12: Αρχιτεκτονική του IEEE 1451.3	38
Σχήμα 13: Αρχιτεκτονική του IEEE 1451.4	39
Σχήμα 14: Σχηματικό διάγραμμα του IEEE 1451.....	40
Σχήμα 15: Βασική δομή TEDS.....	42
Σχήμα 16: Δομή IEEE 1451 ευφυούς αισθητήρα.....	47
Σχήμα 17: NCAP-TIM (Πελάτης-Εξυπηρετητής).....	49
Σχήμα 18: NCAP-TIM (Δημοσιοποίηση-Εγγραφή)	49
Σχήμα 19: Τρόποι πρόσβασης στο IEEE 1451	52
Σχήμα 20: Στοιβά λογισμικού του SunSPOT	60
Σχήμα 21: Στοιβά λογισμικού εφαρμογής ξενιστή.....	60
Σχήμα 22: Αρχιτεκτονική #1 της υλοποίησής μας.....	60
Σχήμα 23: Αρχιτεκτονική #2 της υλοποίησής μας.....	61
Σχήμα 24: Πρωτόκολλο επικοινωνίας χαμηλού επιπέδου	65

ΚΑΤΑΛΟΓΟΣ ΕΙΚΟΝΩΝ

Εικόνα 1: Sensor Web	18
Εικόνα 2: SunSPOT.....	57
Εικόνα 3: Σύγκριση πρότυπης και Squawk εικονικής συσκευής	58
Εικόνα 4: Υλισμικό του SunSPOT	59

ΚΑΤΑΛΟΓΟΣ ΠΙΝΑΚΩΝ

Πίνακας 1: Δομή MetaTEDS.....	43
Πίνακας 2: Δομή Transducer Channel TEDS	44
Πίνακας 3: IEEE 1451 HTTP API.....	51
Πίνακας 4: Αναπαράσταση μονάδων μέτρησης	53
Πίνακας 5: Παράδειγμα απεικόνισης μονάδων (Volts).....	53
Πίνακας 6: Δομή εντολής IEEE 1451	54
Πίνακας 7: Δομή απάντησης IEEE 1451	54

ΠΡΟΛΟΓΟΣ

Η παρούσα διπλωματική εργασία εκπονήθηκε στο πλαίσιο του Μεταπτυχιακού Προγράμματος Σπουδών (ειδίκευση «Συστήματα Επικοινωνιών και Δίκτυα») του Τμήματος Πληροφορικής και Τηλεπικοινωνιών του Πανεπιστημίου Αθηνών. Το αντικείμενο μελέτης της είναι η ανάπτυξη και υλοποίηση ενός συστήματος αισθητήρων βασισμένο στο πρότυπο IEEE 1451. Μελετήθηκαν θέματα σχετικά με ευφυείς αισθητήρες και τον ιστό αισθητήρων (Sensor Web) τα οποία βοήθησαν στην κατανόηση του θέματος και την ανάπτυξη του συστήματος.

Θα ήθελα να ευχαριστήσω θερμά τον επίκουρο καθηγητή Ευστάθιο Χατζηευθυμιάδη ο οποίος μου πρότεινε τη συγκεκριμένη διπλωματική και με καθοδήγησε καθ' όλη τη διάρκεια της εκπόνησής της. Θα ήθελα επίσης να ευχαριστήσω θερμά το Βασίλειο Τσέτσο, διδάκτορα του Τμήματος Πληροφορικής και Τηλεπικοινωνιών ΕΚΠΑ, για τις πολύτιμες συμβουλές και ιδέες του κατά τη διάρκεια ανάπτυξης του συστήματος.

Αθήνα, Οκτώβριος 2010

Νικόλαος Σιλβέστρος

1. ΕΥΦΥΕΙΣ ΑΙΣΘΗΤΗΡΕΣ (SMART SENSORS)

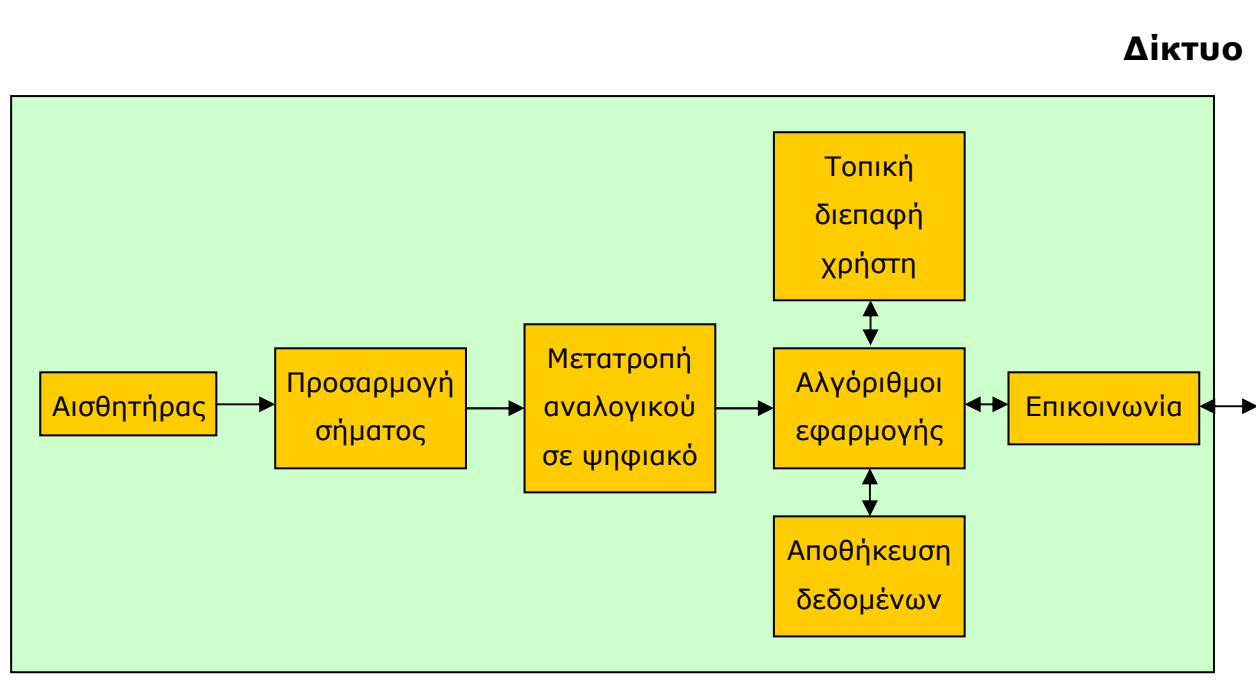
Ο αισθητήρας (sensor) είναι μία συσκευή η οποία μετατρέπει φυσικές, βιολογικές ή χημικές εισόδους σε ηλεκτρικό ή οπτικό σήμα. Για να είναι χρήσιμη η μέτρηση, το σήμα αυτό πρέπει να μετρηθεί και να μετατραπεί σε ψηφιακή μορφή, την οποία μπορούν να αναλύσουν και να επεξεργαστούν αποτελεσματικά οι ηλεκτρονικοί υπολογιστές. Οι εφαρμογές των αισθητήρων καλύπτουν ένα τεράστιο εύρος εφαρμογών οι οποίες περιλαμβάνουν μετρήσεις θερμοκρασίας, θέσης και μετατόπισης αντικειμένου, στάθμης υγρών, ταχύτητας και επιτάχυνσης κινούμενου αντικειμένου, δύναμης, ροής αερίων και ρευστών, τάσης και έντασης ρεύματος, υγρασίας, ακτινοβολίας και πολλών ακόμα.

Επίσης μία άλλη ομάδα συσκευών οι οποίες χρησιμοποιούνται σε συνεργασία με τους αισθητήρες είναι οι εκκινήτες (actuators) οι οποίες μετατρέπουν κάποια μορφή ενέργειας σε κάποιου είδους κίνηση ή άλλου είδους ενέργεια. Για παράδειγμα μπορούν θέτοντάς τους κάποια τάση να διαχειριστούν τη λειτουργία διακοπών, αντλιών και συστημάτων ειδοποίησης. Τα παραπάνω δύο είδη συσκευών, οι αισθητήρες και οι εκκινήτες, ονομάζονται μορφοτροπείς (transducers).

Καθώς οι μορφοτροπείς και τα συστήματα [2] που τους πλαισίωναν έδειχναν ότι μπορούσαν να αλλάξουν τη ζωή μας προς το καλύτερο, άρχισαν να έχουν μεγάλη ανάπτυξη. Ενώ βρισκόταν ακόμα σε πρώιμο στάδιο, είχαν αναπτυχθεί ήδη πολλά και διαφορετικά δίκτυα δεδομένων και μορφοτροπέων. Παρόλο που η ανάπτυξη μορφοτροπέων έδειχνε ότι θα έλυσε πολλά προβλήματα, λίγο μετά την έναρξή της είχε ένα μεγάλο πρόβλημα, το οποίο ήταν η κατασκευή τους από διαφορετικές εταιρίες, οπότε και διαφορετικές αρχιτεκτονικές, οι οποίες προφανώς ήταν ασύμβατες μεταξύ τους. Το γεγονός αυτό είχε ως συνέπεια το αυξημένο κόστος των εξαρτημάτων, καθώς σχεδόν κάθε κατασκευαστής χρηματοδοτούσε την ανάπτυξη του δικού του «προτύπου», με βάση το οποίο σχεδίαζε έπειτα τους μορφοτροπείς. Ένα άλλο μεγάλο πρόβλημα ήταν ότι οι χρήστες αυτών των τεχνολογιών ήταν δεσμευμένοι να συνεργάζονται με έναν και μόνο κατασκευαστή, καθώς ήταν αναγκασμένοι να προμηθεύονται όλα τα επιμέρους εξαρτήματα από αυτόν, ώστε να είναι σίγουροι ως προς τη συμβατότητα και την ορθή επικοινωνία μεταξύ τους. Τη λύση στα παραπάνω προβλήματα ήρθαν για να δώσουν οι ευφυείς αισθητήρες (smart sensors).

Ο όρος ευφυής ή έξυπνος αισθητήρας [1], [7] άρχισε να εμφανίζεται στα μέσα της δεκαετίας του 1980 και περιέγραφε έναν μορφοτροπέα ο οποίος παρείχε κάποια επιπλέον λειτουργικότητα από αυτή που παρείχε ένας απλός αισθητήρας. Οι ευφυείς αισθητήρες, πέρα από τα συστήματα μέτρησης κάποιους μεγέθους, περιλαμβάνουν

επεξεργαστή σήματος, μονάδα μετατροπής αναλογικού σε ψηφιακό σήμα, κάποιου είδους μικροεπεξεργαστή και κάποιο μικρό αποθηκευτικό χώρο δεδομένων. Οι παραπάνω βαθμίδες τους επιτρέπουν να επιστρέφουν τη μέτρησή τους σε ψηφιακή μορφή και μάλιστα σε κάποιες προκαθορισμένες μονάδες μέτρησης, κάτι που διευκολύνει τη διασύνδεσή τους με δικτυακά περιβάλλοντα. Για παράδειγμα, ένας ευφυής αισθητήρας ο οποίος μετράει θερμοκρασία, μπορεί να μετατρέψει την τάση εξόδου ενός κανονικού αισθητήρα θερμοκρασίας, σε δεδομένα τα οποία αντιπροσωπεύουν βαθμούς Celsius ή Fahrenheit. Επίσης οι ευφυείς αισθητήρες είναι αυτοπροσδιοριζόμενοι και κάνουν από μόνοι τους αυτοδιάγνωση χωρίς εξωτερική παρέμβαση. Λόγω αυτών των χαρακτηριστικών, οι ευφυείς αισθητήρες ξεπέρασαν τα κύρια προβλήματα των απλών αισθητήρων και έδωσαν νέα ώθηση στα συστήματα αισθητήρων.



Σχήμα 1: Δομή ευφυούς αισθητήρα

1.1 Εφαρμογές

Οι ευφυείς αισθητήρες μπορούν να χρησιμοποιηθούν σε μία πληθώρα από εφαρμογές [3]. Θα μπορούσαμε να πούμε ότι οι εφαρμογές αυτές, μπορούν να ομαδοποιηθούν σε τρεις κατηγορίες.

Μία σημαντική κατηγορία είναι η συλλογή δεδομένων όπου συνήθως γίνεται συλλογή και αποθήκευση τιμών από διάφορα σημεία για μεγάλα χρονικά διαστήματα. Η επεξεργασία αυτών των δεδομένων μας επιτρέπει να παρατηρούμε εύκολα εξαρτήσεις

και τάσεις που υπάρχουν μεταξύ τους. Κάποιες εφαρμογές που ανήκουν σε αυτή την κατηγορία σχετίζονται με την παρακολούθηση στοιχείων του φυσικού περιβάλλοντος, για παράδειγμα τις συνθήκες που επικρατούν στα σημεία που επιλέγουν για να φωλιάσουν τα πουλιά [12]. Μία σημαντική υποκατηγορία ασχολείται με την παρακολούθηση της υγείας (πίεση, χτύποι καρδιάς) [13] σε ασθενείς που μπορεί να έχουν κάποιο πρόβλημα.

Άλλη μία κατηγορία είναι σχετική με την ασφάλεια. Η διαφορά με την προηγούμενη κατηγορία είναι ότι σε αυτή την περίπτωση δεν είμαστε αναγκασμένοι να αποθηκεύουμε τα δεδομένα. Το μόνο που μας ενδιαφέρει είναι να ειδοποιηθούμε όταν κάποια μέτρηση είναι εκτός κάποιων ορίων ή όταν διαπιστωθεί κάποια ανωμαλία. Εφαρμογές που εντάσσονται σε αυτή την κατηγορία σχετίζονται με τον εντοπισμό κάποιας επικίνδυνης κατάστασης όπως φωτιά [14], [15] ή στάθμης νερού [16] και την ειδοποίηση σε περίπτωση που έχουμε είτε στοιχεία για ενδεχόμενη φωτιά (ύπαρξη καπνού, αύξηση θερμοκρασίας, μείωση υγρασίας) ή για πλημμύρα (αύξηση στάθμης πάνω από κάποιο όριο).

Άλλη μία ομάδα, περιλαμβάνει εφαρμογές οι οποίες ασχολούνται με τον εντοπισμό αντικειμένων επεξεργαζόμενες τα σημεία από τα οποία έχουν περάσει. Με αυτόν τον τρόπο δεν μπορούμε να είμαστε ακριβείς και σίγουροι για την τωρινή θέση του αντικειμένου αλλά μπορούμε να ξέρουμε τη διαδρομή την οποία έχει ακολουθήσει. Ένα παράδειγμα τέτοιας εφαρμογής είναι η παρακολούθηση κάποιων ζώων [17], [18] όσον αφορά τις μετακινήσεις τους και τις αλληλεπιδράσεις τους με το φυσικό περιβάλλον.

1.2 Πλεονεκτήματα

Όπως έχουμε ήδη αναφέρει, το κυριότερο πλεονέκτημα των ευφυών αισθητήρων, σε σχέση με τους απλούς, είναι ότι παρέχουν τις μετρήσεις τους σε προκαθορισμένες ψηφιακές μορφές οι οποίες μπορούν να επεξεργαστούν με μεγάλη ευκολία. Με τη χρήση τους έχουμε μειωμένο κόστος ανάπτυξης συστημάτων, καθώς παρόλο το ότι οι ίδιοι είναι πιο ακριβοί από τους απλούς αισθητήρες, δεν χρειάζεται να χρησιμοποιούνται διαφορετικών ειδών μετατροπείς από αναλογικό σε ψηφιακό σήμα, οπότε τα κυκλώματα προσαρμογής είναι απλούστερα και φθηνότερα. Επίσης έχουν τη δυνατότητα να κάνουν από μόνοι τους βαθμονόμηση και να αναφέρουν πληροφορίες για τον εαυτό τους και τις δυνατότητές τους. Λόγω αυτών των χαρακτηριστικών είναι πολύ εύκολη η προσθήκη

τους σε ένα σύστημα καθώς χρειάζονται ελάχιστες αρχικές ρυθμίσεις για να μπορούν να επικοινωνούν με τις υπόλοιπες οντότητες του συστήματος.

1.3 Μειονεκτήματα

Οι ευφυείς αισθητήρες όμως μερικές φορές μπορεί να δημιουργούν και νέα προβλήματα [4] τα οποία οι κλασσικοί αισθητήρες δεν είχαν. Το πιο σημαντικό από αυτά είναι το αυξημένο κόστος τους λόγω της ενσωμάτωσης ηλεκτρονικών που προσδίδουν την «εξυπνάδα» στους αισθητήρες. Επίσης η ολοκλήρωση και ενσωμάτωση όλων αυτών των επιπλέον ηλεκτρονικών στους αισθητήρες προκαλεί κι άλλα προβλήματα που έχουν να κάνουν κυρίως με τα περιβάλλοντα και τον τρόπο που λειτουργούν οι αισθητήρες. Για παράδειγμα κάποιοι αισθητήρες πρέπει να τοποθετούνται σε περιβάλλοντα με συνθήκες στις οποίες τα ηλεκτρονικά δεν μπορούν να λειτουργήσουν (πίεση ή θερμοκρασία εκτός των ορίων λειτουργίας τους) ή να μπορούν να λειτουργήσουν αλλά να επηρεάζεται σε μεγάλο βαθμό η αξιοπιστία τους. Μερικές φορές επηρεάζεται ακόμα και ο ίδιος ο αισθητήρας από τα ενσωματωμένα ηλεκτρονικά καθώς μπορεί να παρεμβάλουν στον τρόπο μέτρησης του φυσικού μεγέθους. Για να ξεπεραστούν τα παραπάνω προβλήματα, χρειάζεται μεγαλύτερη έρευνα στον τρόπο κατασκευής και ολοκλήρωσης, κάτι το οποίο αυξάνει ακόμα περισσότερο την πολυπλοκότητα και το κόστος. Επίσης παρόλο που οι έξοδοι των ευφυών αισθητήρων είναι ψηφιακές, αυτό δεν σημαίνει ότι έχουν προκαθορισμένη μορφή, καθώς κάθε κατασκευαστής ακολουθεί τον δικό του σχεδιασμό, οπότε η επικοινωνία με κάθε έναν αισθητήρα μπορεί να μην είναι τόσο εύκολη υπόθεση.

Αυτό που συμπεραίνουμε είναι ότι πιο σημαντικό από τους ίδιους τους ευφυείς αισθητήρες, είναι η προτυποποίηση των διεπαφών με αυτούς. Αυτό θα έχει ως αποτέλεσμα τον ορισμό ενός κοινού τρόπου επικοινωνίας των συστημάτων με τους αισθητήρες και του τρόπου λήψης των μετρήσεών τους. Για να επιτευχθεί αυτό είναι ανάγκη να αναπτυχθούν πρότυπα τα οποία θα καθορίζουν λεπτομερώς τις απαραίτητες διεπαφές, τα οποία όμως θα πρέπει να γίνουν ευρέως αποδεκτά και να ακολουθούνται από τους κατασκευαστές. Σε επόμενα κεφάλαια περιγράφονται τα πιο σημαντικά από αυτά.

1.4 Κόμβοι αισθητήρων

Όπως αναφέραμε, είναι αρκετές οι διαφορετικές τεχνολογίες και υλοποιήσεις που υπάρχουν σήμερα, όσον αφορά τους κόμβους αισθητήρων, με αρκετά δημοφιλείς (τουλάχιστον στην ακαδημαϊκή κοινότητα), τους Mica, T-Mote Sky και SunSPOT [59]. Προφανώς δεν μας ενδιαφέρει τόσο η εσωτερική τους δομή, αλλά οι τρόποι επικοινωνίας μαζί τους και ο προγραμματισμός τους.

Οι τρόποι επικοινωνίας μπορεί να ποικίλουν από εντελώς εξατομικευμένους, οι οποίοι μπορεί να χρησιμοποιούνται αποκλειστικά από κάποιον κατασκευαστή, σε διεπαφές οι οποίες είναι ανοιχτές και συνηθισμένες σε τέτοιους είδους κόμβων και όχι μόνο. Οι πιο συχνά χρησιμοποιούμενες ενσύρματες διεπαφές είναι η USB (Universal Serial Bus) και η παλαιότερη RS-232 και οι πιο διαδεδομένες ασύρματες είναι η IEEE 802.15.4 (ZigBee) και η IEEE 802.15.1 (Bluetooth).

Όσον αφορά τα λειτουργικά συστήματα, τα οποία μπορεί να τρέχουν στους κόμβους, υπάρχει μεγάλη ποικιλία καθώς πέρα από τα συχνά χρησιμοποιούμενα TinyOS [57] και Contiki [58], πολλοί κατασκευαστές καταφεύγουν σε δικές τους υλοποιήσεις, εστιασμένες στο υλικό που χρησιμοποιούν, οι οποίες μπορεί να βασίζονται πάνω σε Linux ή Java ή και ολοκληρωτικά δικές τους εξατομικευμένες υλοποιήσεις.

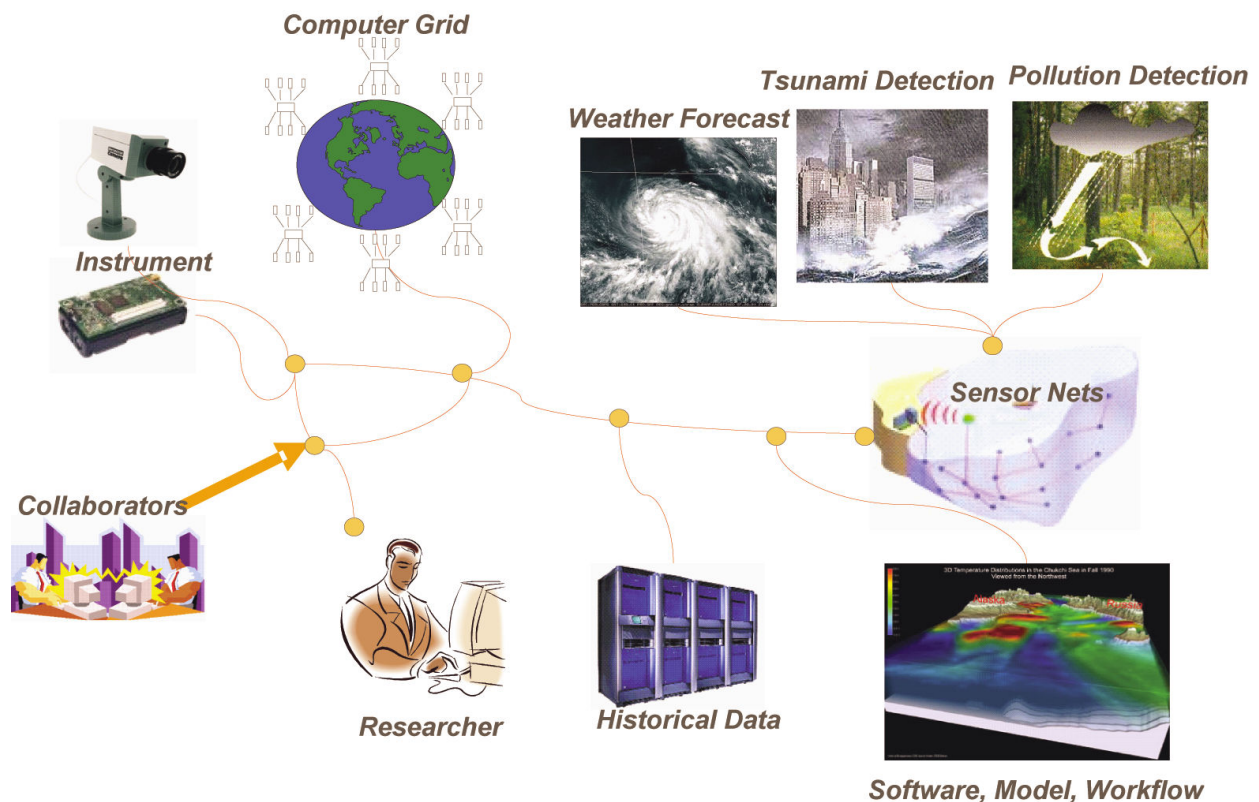
Ο προγραμματισμός των κόμβων συνήθως γίνεται είτε με τη χρήση των γλωσσών προγραμματισμού C και Java. Προφανώς ο τρόπος με τον οποίο μπορούμε να πάρουμε μία μέτρηση από έναν αισθητήρα και να την στείλουμε μέσω κάποιας διεπαφής επικοινωνίας στο σύστημα, είναι εντελώς διαφορετικός ανά κατασκευαστή ακόμα και αν χρησιμοποιείται κοινή γλώσσα προγραμματισμού και διεπαφή επικοινωνίας.

Λόγω αυτής της ποικιλομορφίας στους παραπάνω τομείς, αντιλαμβανόμαστε εύκολα ότι το να σχεδιαστεί και να αναπτυχθεί ένα σύστημα, χρησιμοποιώντας έστω και δύο διαφορετικούς τύπους κόμβων αισθητήρων, δεν είναι καθόλου εύκολη υπόθεση. Για αυτό το λόγο είναι απαραίτητος ο σχεδιασμός κάποιου κοινού αποδεκτού προτύπου, στο οποίο θα πρέπει να «υπακούουν» οι διαθέσιμοι κόμβοι ώστε να είναι εύκολη η χρησιμοποίηση διαφορετικών υλοποιήσεων χωρίς αλλαγές στο σύστημα. Στα επόμενα κεφάλαια περιγράφονται αναλυτικά κάποιες από τις λύσεις που έχουν προταθεί και είναι ευρέως αποδεκτές.

2. Ιστός αισθητήρων (Sensor Web)

Πέρα από τη τοποθέτηση αισθητήρων σε προκαθορισμένα σημεία και την εγκατάσταση της ανάλογης δικτυακής υποδομής για την επικοινωνία μαζί τους, υπάρχει μία άλλη αρχιτεκτονική η οποία μας επιτρέπει να έχουμε πρόσβαση στα δεδομένα των αισθητήρων μέσω του παγκόσμιου ιστού, με τη χρήση υπηρεσιοστρεφούς αρχιτεκτονικής (Service Oriented Architecture – SOA), χωρίς να μας ενδιαφέρει η διάταξη με την οποία είναι αυτοί τοποθετημένοι. Αν το σύστημα σχετίζεται με μετεωρολογικά ή ανάλογου τύπου στοιχεία, τότε συνήθως η απεικόνιση των δεδομένων γίνεται πάνω σε κάποιο χάρτη, ο οποίος μας επιτρέπει με εύκολο τρόπο να βλέπουμε όχι μόνο τις τιμές των μετρήσεων αλλά και την ακριβή θέση που αυτές έγιναν. Σε τέτοιου είδους περιβάλλοντα, υπάρχει η δυνατότητα είτε όλοι οι αισθητήρες να έχουν αυτόνομη πρόσβαση στο διαδίκτυο, οπότε να μπορούμε να έχουμε πρόσβαση στον καθένα απευθείας, ή κάποιοι να συνδέονται με το διαδίκτυο μέσω κοντινών αισθητήρων, οι οποίοι είναι μέσα στο ασύρματο εύρος τους και έχουν άμεση πρόσβαση με αυτό. Ο συνδυασμός των δικτύων αισθητήρων με το διαδίκτυο αναφέρεται ως ιστός αισθητήρων (sensor web). Σε αυτή την υποδομή αν προσθέσουμε και γεωγραφικά κατανομημένες υπηρεσίες τότε μιλάμε για μία νέου είδους αρχιτεκτονική η οποία αναφέρεται ως πλέγμα αισθητήρων (sensor grid). Το μεγάλο πλεονέκτημα της συγκεκριμένης αρχιτεκτονικής είναι ότι η απαιτητική επεξεργασία των δεδομένων μπορεί να γίνεται σε κατανομημένα υπολογιστικά συστήματα (και όχι στους υπολογιστικά αδύναμους ευφυείς αισθητήρες), κάτι το οποίο μειώνει την κατανάλωση ενέργειας στους αισθητήρες, παρατείνοντας το χρόνο ζωής τους και τους επιτρέπει να «αφοσιωθούν» αποκλειστικά στη διάγνωση και αποστολή των πληροφοριών που συλλέγουν.

Στην Εικόνα 1 απεικονίζεται ένα αφηρημένο όραμα της μορφής και της αρχιτεκτονικής του ιστού αισθητήρων. Διάφοροι αισθητήρες συνδέονται μεταξύ τους και σχηματίζουν μία μορφή ιστού και παρουσιάζονται ως διαθέσιμες υπηρεσίες σε όλους τους χρήστες του παγκόσμιου ιστού. Κάποιος χρήστης που επιθυμεί να λάβει κάποια πληροφορία, για παράδειγμα αν πρόκειται να συμβεί τσουνάμι σε κάποια περιοχή της γης, μπορεί απλά να απευθύνει την ερώτησή του στον ιστό αισθητήρων και αυτός να του παρέχει κάποια έγκυρη απάντηση κάνοντας συνδυασμό δεδομένων πραγματικού χρόνου από τους απαραίτητους αισθητήρες αλλά και αποθηκευμένων σε κάποια βάση δεδομένων. Η επεξεργασία των δεδομένων μπορεί να γίνει κατανομημένα με τη χρήση υπολογιστικών πλεγμάτων, τα οποία μπορεί να παρέχονται από διαφορετικές πηγές.



Εικόνα 1: Sensor Web

Τα τελευταία χρόνια, τέτοια περιβάλλοντα [21] έχουν αρχίσει να έχουν μεγάλη ανάπτυξη καθώς αυξάνεται συνεχώς η διαθεσιμότητα σε όλο και πιο φθηνούς ευφυείς αισθητήρες. Το όραμα πολλών, είναι να δημιουργηθεί κάποιο παγκόσμιο δίκτυο αισθητήρων, στο οποίο θα μπορούν να ενσωματωθούν οι υπάρχουσες ετερογενείς τεχνολογίες. Το συγκεκριμένο σύστημα θα πρέπει να υποστηρίζει την αυτόματη ένταξη και ανακάλυψη δικτύων αισθητήρων, να προσφέρει κατανομημένες αιτήσεις για συλλογή δεδομένων, φιλτράρισμα των δεδομένων, συνδυασμό των στοιχείων των αισθητήρων και δυναμική προσαρμογή και διαμόρφωση του συστήματος, κατά τη διάρκεια της λειτουργίας του, μέσω ανοιχτών προτύπων και διεπαφών, οι οποίες θα χρησιμοποιούν την XML (eXtensible Markup Language) ως γλώσσα επικοινωνίας.

Οι κύριοι στόχοι του συστήματος είναι οι παρακάτω:

- απλότητα: το σύστημα βασίζεται σε λίγες αλλά ισχυρές αφαιρέσεις οι οποίες θα επιτρέψουν την επικοινωνία και συντονισμό διαφορετικών τεχνολογιών
- προσαρμοστικότητα: η εισαγωγή ακόμα και καινούργιων τύπων αισθητήρων θα γίνεται δυναμικά κατά τη διάρκεια λειτουργίας του συστήματος

- επεκτασιμότητα: καθώς ο αριθμός των συνεργαζόμενων αισθητήρων, των αιτήσεων και των μεταφερόμενων δεδομένων μπορεί να αυξάνονται με μεγάλους ρυθμούς, θα πρέπει να υποστηρίζεται κατανεμημένη επεξεργασία αιτήσεων αλλά και κατανεμημένη ανακάλυψη νέων αισθητήρων
- χαμηλών απαιτήσεων υλοποίηση: ολόκληρη η εφαρμογή θα μπορεί να εκτελείται σε χαμηλού κόστους υπολογιστικά συστήματα χωρίς μεγάλες απαιτήσεις σε εύρος ζώνης. Θα πρέπει το σύστημα να είναι φορητό και να χρειάζεται ελάχιστες ρυθμίσεις για να εγκατασταθεί και να χρησιμοποιηθεί

2.1 SWE (Sensor Web Enablement)

Το Open Geospatial Consortium (OGC) [22] είναι μία διεθνής κοινοπραξία, στην οποία συμμετέχουν πάνω από 400 εταιρίες, κυβερνητικοί οργανισμοί και πανεπιστήμια, η οποία έχει ως σκοπό τον καθορισμό αλλά και ανάπτυξη ανοιχτών και δημόσια διαθέσιμων προτύπων. Τα πρότυπα αυτά δίνουν τη δυνατότητα ανάπτυξης περίπλοκων συστημάτων και εφαρμογών οι οποίες «υπακούουν» σε αυτά, οπότε υπάρχει η δυνατότητα της εύκολης επικοινωνίας και διαλειτουργικότητας μαζί τους.

Μία από τις πολλές ομάδες εργασίας, οι οποίες ασχολούνται με τα παραπάνω ζητήματα, είναι η Sensor Web Enablement (SWE). Τα πρότυπα που αναπτύσσει αυτή η ομάδα, επιτρέπουν την ανάπτυξη αισθητήρων και εκκινητήρων οι οποίοι μπορούν να προσπελαστούν και να χρησιμοποιηθούν μέσω του παγκόσμιου ιστού. Θα μπορούσαμε να πούμε ότι το SWE είναι ανάλογο του διαδικτύου, όμως από την πλευρά των αισθητήρων. Το κυριότερο όραμά του είναι στο μέλλον, να μπορούμε να έχουμε πρόσβαση σε αισθητήρες, σε παγκόσμιο επίπεδο, με ανάλογα απλό τρόπο που σήμερα έχουμε πρόσβαση στις πληροφορίες του παγκόσμιου ιστού.

Οι κύριοι στόχοι του SWE είναι η ανακάλυψη αισθητήρων διαφορετικών τύπων, ο προσδιορισμός των δυνατοτήτων των διαθέσιμων αισθητήρων, η παροχή γεωγραφικών πληροφοριών, η δυνατότητα για μετρήσεις πραγματικού χρόνου κωδικοποιημένες με προκαθορισμένο τρόπο και η υποστήριξη ειδοποιήσεων σε περιπτώσεις που ικανοποιούνται κάποια κριτήρια.

2.1.1 Πρότυπα

Τα κύρια πρότυπα [23], [53], [56] με τα οποία ασχολείται η συγκεκριμένη ομάδα χωρίζονται σε δύο μεγάλες κατηγορίες, στο μοντέλο πληροφοριών και το μοντέλο υπηρεσιών.

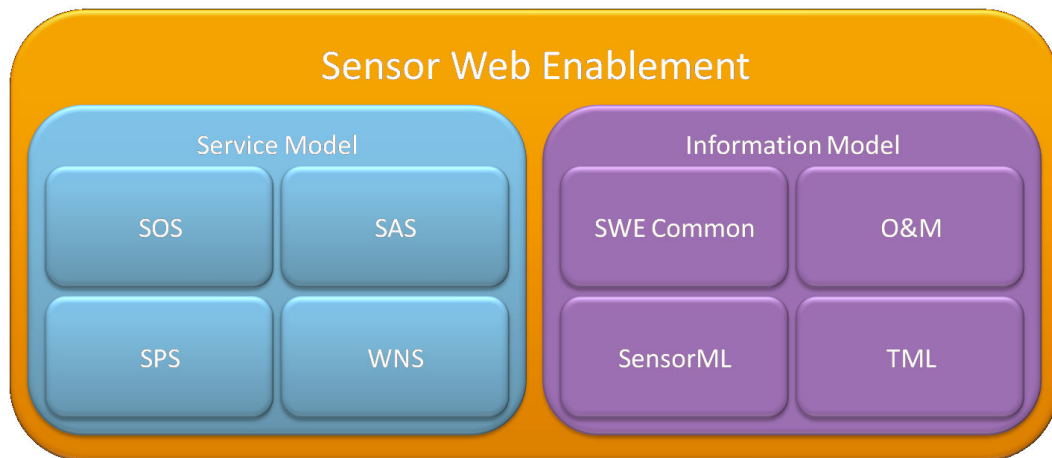
Στο μοντέλο πληροφοριών ανήκουν τα παρακάτω πρότυπα:

- **Παρατηρήσεις και Μετρήσεις (Observations & Measurements - O&M):** γενικά μοντέλα και XML κωδικοποιήσεις για παρατηρήσεις και μετρήσεις αισθητήρων. Με τη χρήση του επιτυγχάνεται κοινή μορφή δεδομένων ακόμα και αν αυτά έχουν ληφθεί από διαφορετικούς τύπους αισθητήρων.
- **Γλώσσα Μοντέλου Αισθητήρα (Sensor Model Language - SensorML):** μοντέλα και XML σχήματα τα οποία περιγράφουν τα συστήματα αισθητήρων και τις διεργασίες που σχετίζονται με τις μετρήσεις τους. Παρέχει πληροφορίες οι οποίες απαιτούνται για την ανακάλυψη αισθητήρων και τον καθορισμό της τοποθεσίας λήψης των μετρήσεων. Επίσης υποστηρίζει επεξεργασία των παρατηρήσεων αυτών.
- **Γλώσσα Σήμανσης Μορφοτροπία (Transducer Markup Language - TML):** εννοιολογικό μοντέλο και XML κωδικοποιήσεις για την πραγματικού χρόνου υποστήριξη ροής παρατηρήσεων και αποστολής εντολών από και προς τα συστήματα αισθητήρων

Στο μοντέλο υπηρεσιών ανήκουν τα παρακάτω πρότυπα:

- **Υπηρεσία Παρατήρησης Αισθητήρα (Sensor Observation Service - SOS):** ανοιχτή διεπαφή επικοινωνίας με τους αισθητήρες για λήψη παρατηρήσεων ή περιγραφών της πλατφόρμας με τη χρήση υπηρεσιών ιστού
- **Υπηρεσία Σχεδιασμού Αισθητήρα (Sensor Planning Service - SPS):** ανοιχτή διεπαφή υπηρεσίας ιστού μέσω της οποίας κάποιος πελάτης μπορεί να σχεδιάσει τη λήψη δεδομένων και μετρήσεων από έναν ή περισσότερους αισθητήρες καθώς και να την υποβάλει στο σύστημα
- **Υπηρεσία Ειδοποίησης Αισθητήρα (Sensor Alert Service - SAS):** υπηρεσία ιστού για τη δημοσίευση και εγγραφή σε ειδοποιήσεις παραγόμενες από τους αισθητήρες

- **Υπηρεσία Ειδοποίησης Ιστού (Web Notification Services - WNS):** υπηρεσία ιστού για ασύγχρονη αποστολή μηνυμάτων και ειδοποιήσεων

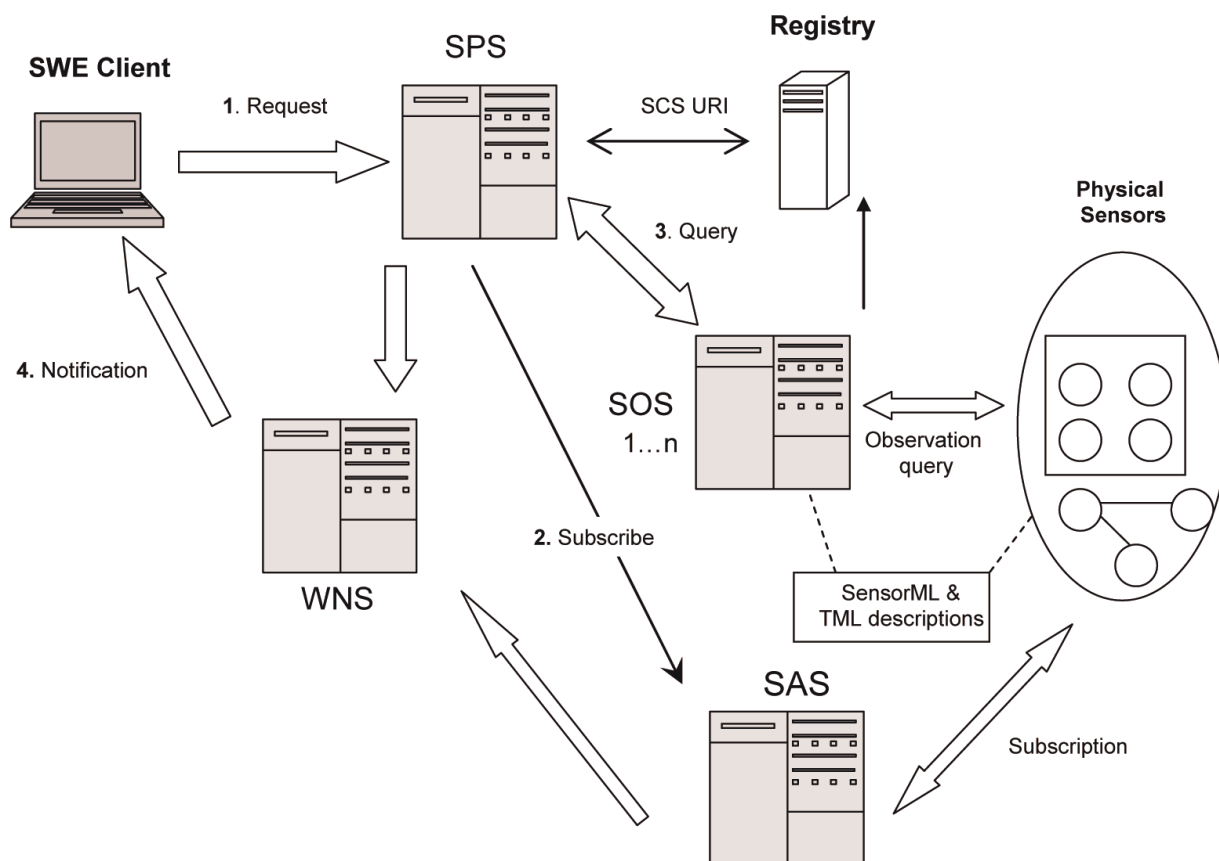


Σχήμα 2: Δομή SWE

2.1.2 Αρχιτεκτονική

Το βασικό όραμα του SWE είναι οι διαφορετικές οντότητες του συστήματος να μπορούν να συνεργάζονται μεταξύ τους, ώστε να μπορούν να εξυπηρετήσουν τις αιτήσεις των χρηστών. Στο Σχήμα 3 παρατηρούμε τις διαδικασίες που ακολουθούνται εσωτερικά στο σύστημα ώστε να ικανοποιηθεί η αίτηση ενός πελάτη.

Αρχικά ο χρήστης υποβάλλει στο σύστημα μία αίτηση, για παράδειγμα ζητάει κάποιες μετρήσεις από μία ομάδα αισθητήρων. Η αίτηση περιλαμβάνει πληροφορίες που σχετίζονται με το γεωγραφικό τόπο των επιθυμητών μετρήσεων, τη χρονική διάρκεια των μετρήσεων και τον τύπο τους (θερμοκρασία, υγρασία κτλ). Η αίτηση του χρήστη κατευθύνεται προς το SPS. Το SPS έπειτα ενημερώνεται από το μητρώο για τα διαθέσιμα SOS και την προωθεί σε ένα από αυτά. Έπειτα το SOS θα στείλει την αίτηση στους απαραίτητους αισθητήρες και θα συγκεντρώσει τα απαραίτητα δεδομένα. Όταν συγκεντρωθούν όλα, το SPS θα ενημερώσει το WNS για την επιτυχή ολοκλήρωση της αίτησης και τότε αυτό θα επιστρέψει τα απαιτούμενα δεδομένα στον πελάτη. Αν ο πελάτης επιθυμεί και κάποιου είδους ειδοποιήσεις, τότε το SPS κάνει εγγραφή και σε κάποιο SAS, το οποίο σε περίπτωση που θέλει να ειδοποιήσει το συγκεκριμένο πελάτη, θα ενημερώσει το WNS το οποίο θα προωθήσει την ειδοποίηση πίσω στον πελάτη.



Σχήμα 3: Αρχιτεκτονική SWE

2.1.3 Sensor Model Language (SensorML)

Η SensorML χρησιμοποιείται για να περιγράψει τις διεργασίες, τις διαδικασίες και τις επεξεργαστικές οντότητες που σχετίζονται με τις μετρήσεις αλλά και το μετασχηματισμό αυτών. Μεμονωμένες διεργασίες μπορούν να ομαδοποιηθούν μεταξύ τους, σχηματίζοντας εκτελέσιμες αλυσίδες διεργασιών, οι οποίες παίρνουν κάποιες εισόδους και παράγουν τελικά την επιθυμητή έξοδο. Οι αλυσίδες διεργασιών είναι χρήσιμες για την επίτευξη πολύπλοκων και συνδυαστικών εργασιών, όπου μία μεμονωμένη διεργασία δεν θα μπορούσε να διεξάγει. Η υπηρεσία O&M χρησιμοποιεί την SensorML για να μοντελοποιήσει τις μετρήσεις που πραγματοποιεί. Η υπηρεσία SOS την χρησιμοποιεί για να περιγράψει τις δυνατότητες και τα μεταδεδομένα κάθε διαθέσιμου αισθητήρα. Η υπηρεσία SPS δέχεται το σχεδιασμό των μετρήσεων από το χρήστη με χρήση της SensorML. Επίσης παίζει σημαντικό ρόλο στην ανακάλυψη των αισθητήρων καθώς μπορεί να κωδικοποιήσει πληροφορίες σχετικές με τη λειτουργία τους, την τοποθεσία τους και τις δυνατότητές τους.

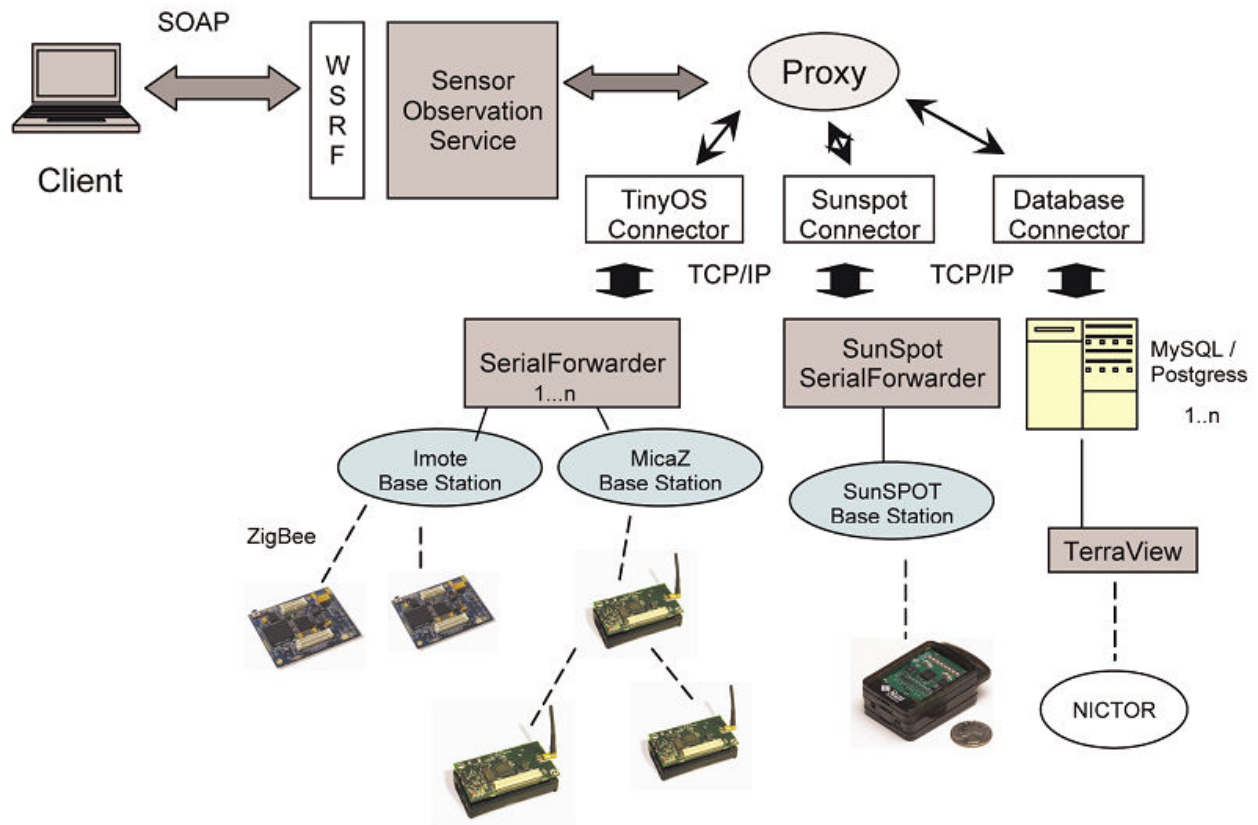
2.1.4 Transducer Markup Language (TML)

Η TML ορίζει ένα σύνολο από μοντέλα τα οποία χρησιμοποιούνται στην περιγραφή των δεδομένων που λαμβάνονται από τους αισθητήρες. Επίσης περιλαμβάνει πληροφορίες οι οποίες είναι απαραίτητες για την επεξεργασία αυτών των δεδομένων από τον παραλήπτη τους. Λόγω της καθορισμένης μορφής της, είναι κατάλληλη όχι μόνο για μεταφορά δεδομένων πραγματικού χρόνου από τους αισθητήρες αλλά και ιστορικών δεδομένων τα οποία έχουν αποθηκευτεί με κάποιο τρόπο.

2.1.5 Sensor Observation Service (SOS)

Η SOS είναι υπεύθυνη για τη διαβίβαση αιτημάτων από τους χρήστες στα διαφορετικά δίκτυα αισθητήρων. Οι τρεις βασικές λειτουργίες που περιλαμβάνει είναι η περιγραφή αισθητήρα η οποία επιστρέφει μεταδεδομένα σχετικά με τη μορφή και τις λειτουργίες των αισθητήρων, η λήψη μέτρησης η οποία επιστρέφει μετρήσεις από ένα δίκτυο αισθητήρων και η λήψη δυνατοτήτων που επιστρέφει πληροφορίες σχετικές με την SOS. Υποστηρίζονται και άλλες λειτουργίες που σχετίζονται με διαφορετικούς τύπους μετρήσεων.

Όταν κάποιος πελάτης συνδεθεί με την SOS, αρχικά ζητάει να λάβει περιγραφή για τους αισθητήρες και τις δυνατότητες της υπηρεσίας. Η SOS κάνει ανάκτηση αυτών των πληροφοριών από κάποια βάση δεδομένων, τις κωδικοποιεί σε γλώσσα SensorML και τις επιστρέφει στον πελάτη. Έπειτα ο πελάτης στέλνει κάποια αίτηση μέτρησης η οποία θα έχει σύνταξη ανάλογη των SQL (Structured Query Language) ερωτημάτων και μπορεί να περιλαμβάνει διάφορες συνθήκες όπως τον τύπο του αισθητήρα, την τοποθεσία του, το φαινόμενο για το οποίο ενδιαφέρεται (φως, θερμοκρασία, επιτάχυνση), τιμές κατωφλίου (μετρήσεις θερμοκρασίας μεγαλύτερες από Χ βαθμούς Κελσίου), τη διάρκεια και τη συχνότητα των μετρήσεων. Αφού επεξεργαστούν όλα τα δεδομένα της αίτησης, θα ειδοποιηθούν οι απαραίτητες οντότητες ώστε να επικοινωνήσουν με τους αντίστοιχους αισθητήρες και να επιστρέψουν στην SOS τις ζητούμενες μετρήσεις. Εφόσον συλλεχθούν όλα τα απαραίτητα δεδομένα, κωδικοποιούνται και αποστέλλονται στον πελάτη.



Σχήμα 4: Αρχιτεκτονική του SOS

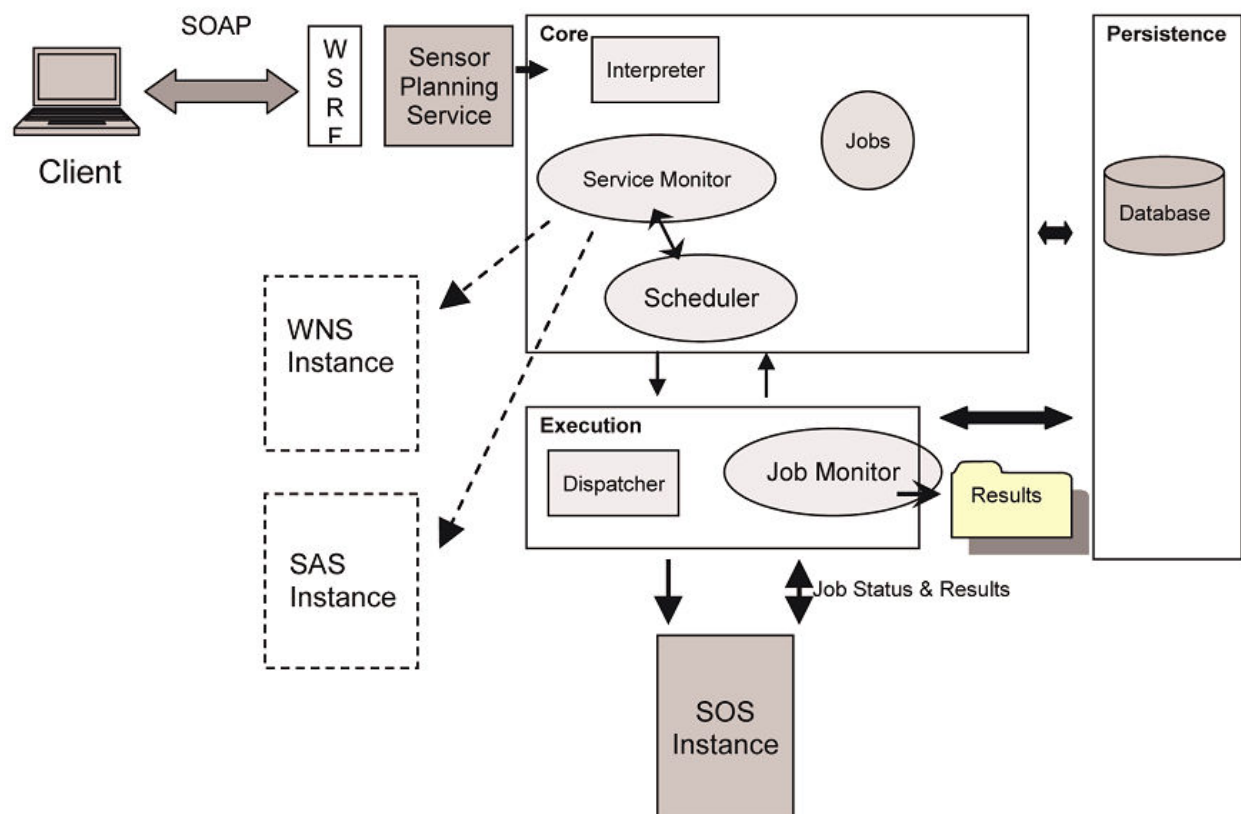
2.1.6 Sensor Planning Service (SPS)

Η SPS είναι υπεύθυνη για την παροχή υψηλού επιπέδου σχεδιασμό, προγραμματισμό, συλλογή, επεξεργασία και αρχειοθέτηση των αιτήσεων για όλες τις υπηρεσίες. Οι πελάτες μπορούν να υποβάλουν στην SPS σχέδια κωδικοποιημένα σε SensorML, τα οποία να περιλαμβάνουν το αίτημα της παρατήρησης, την τοποθεσία των αισθητήρων, τη διάρκεια των μετρήσεων και ότι άλλα στοιχεία και μεταδεδομένα είναι απαραίτητα για τη λήψη και επεξεργασία των μετρήσεων. Η SPS είναι υπεύθυνη για την ανακάλυψη διαθέσιμων SOS από ένα μητρώο υπηρεσιών και δυνατοτήτων, την επεξεργασία και προγραμματισμό του σχεδίου και τη διαχείριση αιτήσεων εγγραφής για την SAS και τη διαβίβαση ειδοποιήσεων στην WNS. Η χρήση της SPS θα πρέπει να περιορίζεται σε περίπλοκες αιτήσεις παρατηρήσεων, στις οποίες απαιτείται να συνδυαστούν δεδομένα και μετρήσεις από διαφορετικά SOS, οι οποίες απαιτούν επεξεργασία των συλλεγμένων δεδομένων.

Οι βασικές λειτουργίες που παρέχει είναι η λήψη των δυνατοτήτων της υπηρεσίας, η περιγραφή εργασίας από έναν πελάτη, η υποβολή εργασίας και η περιγραφή πρόσβασης αποτελέσματος που επιστρέφει την τοποθεσία της SOS που θα λάβει τις

απαραίτητες μετρήσεις. Κάποιες προαιρετικές λειτουργίες επιτρέπουν τη διαχείριση των εργασιών, καθώς δίνουν τη δυνατότητα να πληροφορηθεί ο πελάτης για την τρέχουσα κατάσταση της αίτησής του, να μεταβάλλει τις παραμέτρους της ή να την ακυρώσει.

Για να επικοινωνήσει κάποιος πελάτης με την SPS, χρησιμοποιεί το Απλό Πρωτόκολλο Πρόσβασης Αντικειμένων (Simple Object Access Protocol – SOAP), το οποίο χρησιμοποιείται στην υλοποίηση υπηρεσιών ιστού (Web Services). Στην ουσία βασίζεται στη μεταφορά δομημένων XML μηνυμάτων πάνω από το πρωτόκολλο HTTP (HyperText Transfer Protocol). Αν κάποιος πελάτης συνδεθεί για πρώτη φορά, μπορεί να ενημερωθεί για τις δυνατότητες της υπηρεσίας. Έπειτα μπορεί να αιτηθεί κάποια περιγραφή πρόσβασης αποτελέσματος για να ενημερωθεί για την τοποθεσία της SOS η οποία είναι υπεύθυνη για κάποιο συγκεκριμένο αισθητήρα που ενδιαφέρεται. Τα δεδομένα που θα λάβει, μπορούν να χρησιμοποιηθούν για την κατασκευή του σχεδίου του. Με τη χρήση της περιγραφής εργασίας μπορεί να ενημερωθεί για τις παραμέτρους που χρειάζονται για την υποβολή κάποιας εργασίας. Αφού συνδυάσει τις παραπάνω πληροφορίες μπορεί να υποβάλει μία εργασία την οποία θα εκτελέσει κάποια SOS.

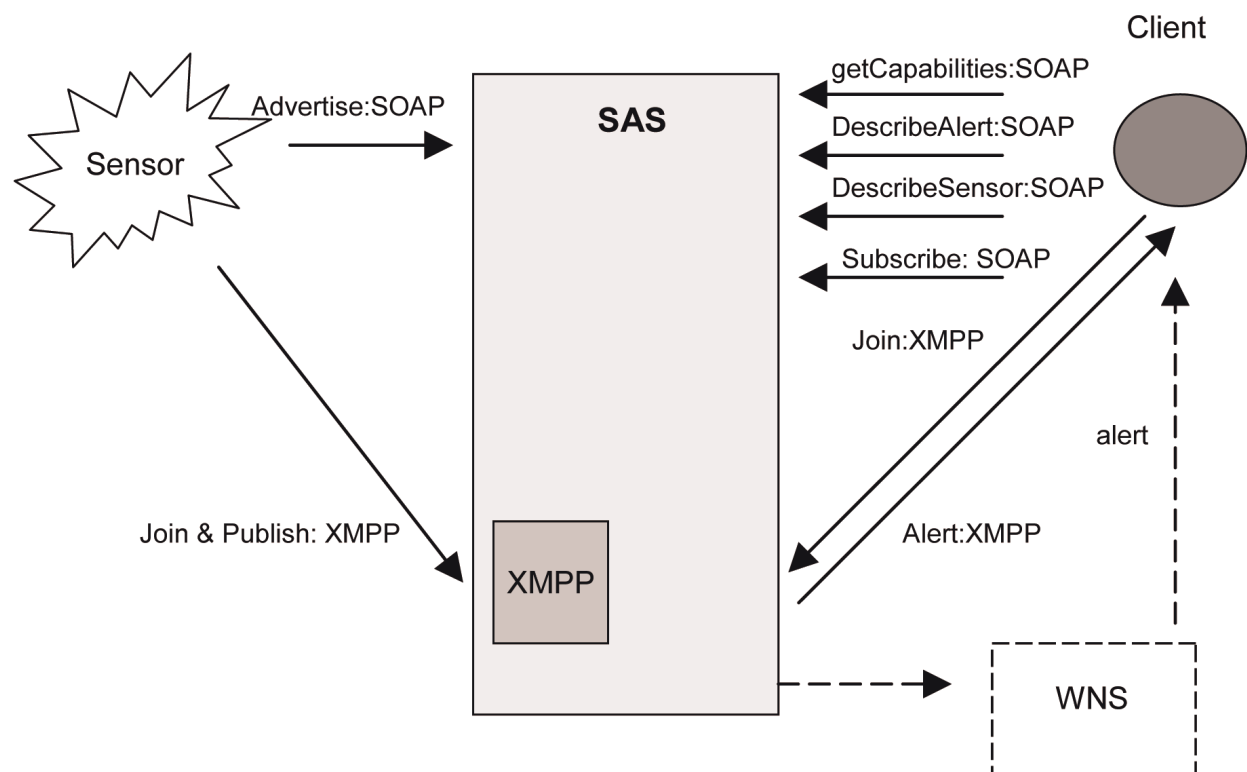


Σχήμα 5: Αρχιτεκτονική SPS

2.1.7 Sensor Alert Service (SAS)

Η SAS παρέχει μία διεπαφή για την εγγραφή και δημοσίευση προειδοποιήσεων. Οι πελάτες μπορούν να εγγραφούν για να λαμβάνουν ειδοποιήσεις όταν κάποια δεδομένα ικανοποιούν τα κριτήρια που έχουν θέσει, για παράδειγμα όταν η μπαταρία ενός αισθητήρα έχει μειωμένη ενέργεια ή όταν η θερμοκρασία είναι πάνω από κάποιους βαθμούς. Ευφυείς αισθητήρες μπορούν να συνδεθούν με την SAS και να κάνουν τους πόρους τους διαθέσιμους προς εγγραφή στους πελάτες. Η δημοσίευση των δεδομένων των αισθητήρων γίνεται χρησιμοποιώντας το Πρωτόκολλο Επεκτάσιμης Μηνυματοδοσίας και Παρουσίας (Extensible Messaging and Presence Protocol – XMPP) το οποίο βασίζεται στην ανταλλαγή XML μηνυμάτων και χρησιμοποιείται κυρίως σε εφαρμογές στιγμιαίας μηνυματοδοσίας (instant messaging).

Οι βασικές λειτουργίες που παρέχει είναι η λήψη δυνατοτήτων της υπηρεσίας, η εγγραφή πελατών, η ακύρωση εγγραφής, η ανανέωση εγγραφής, η περιγραφή προειδοποίησης η οποία περιλαμβάνει το φυσικό φαινόμενο που παρακολουθείται και τη μορφή των δεδομένων και η περιγραφή αισθητήρα.

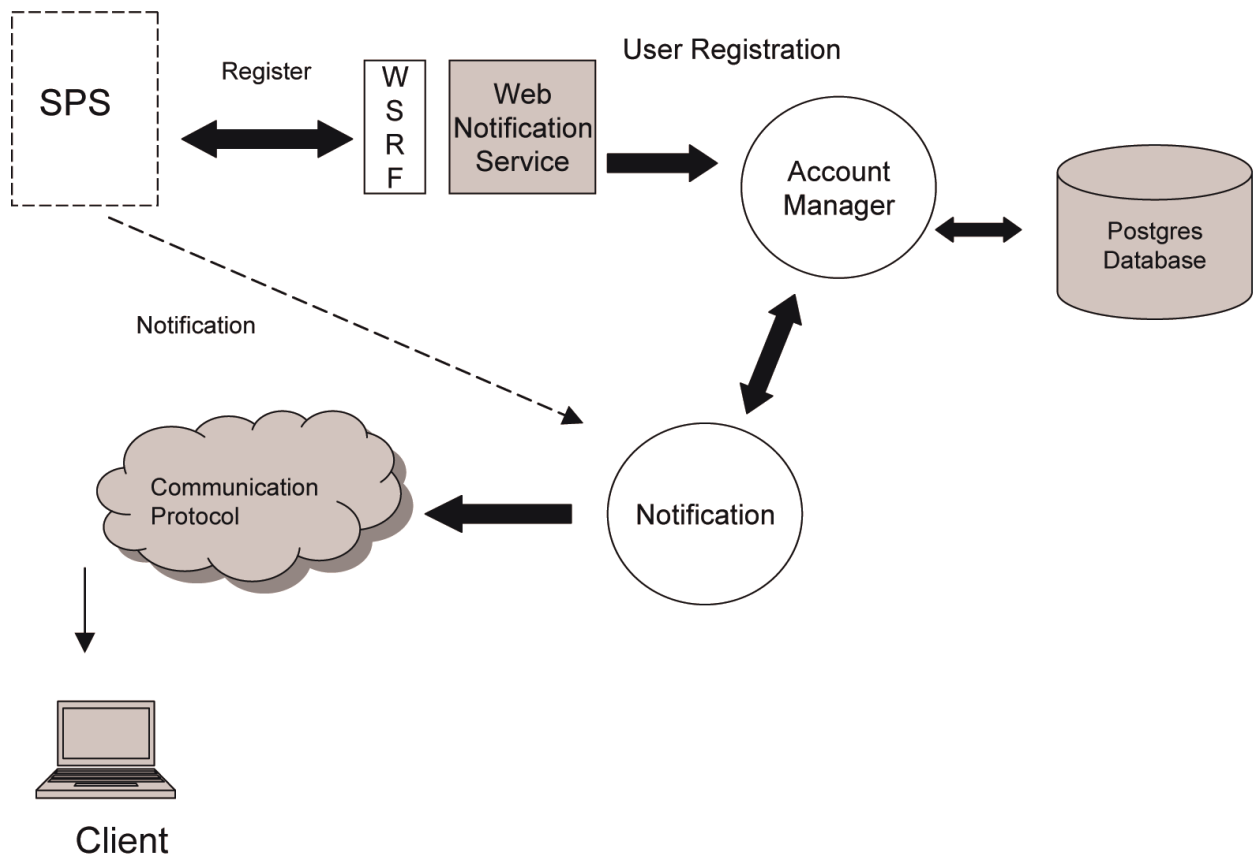


Σχήμα 6: Αρχιτεκτονική SAS

2.1.8 Web Notification Services (WNS)

Η WNS είναι μία ασύγχρονη υπηρεσία μηνυμάτων όπου οι χρήστες μπορούν να εγγραφούν και να λαμβάνουν κοινοποιήσεις σχετικές με την κατάσταση και τα συμβάντα που λαμβάνουν μέρος στην SWE. Υποστηρίζονται δύο τύποι ειδοποιήσεων, η μονόδρομη κατά την οποία οι ειδοποιήσεις κατευθύνονται αποκλειστικά από τις υπηρεσίες προς τους πελάτες και η αμφίδρομη όπου αναμένεται και κάποια απάντηση από τον πελάτη. Οι ειδοποιήσεις μπορούν να έχουν διάφορες μορφές όπως ηλεκτρονικά μηνύματα ή μηνύματα για κινητά.

Οι βασικές λειτουργίες που παρέχει είναι η λήψη δυνατοτήτων της υπηρεσίας, η εγγραφή χρήστη για να δέχεται ειδοποιήσεις και η έναρξη αποστολής ειδοποιήσεων.



Σχήμα 7: Αρχιτεκτονική WNS

2.1.9 Πλεονεκτήματα

Το κύριο πλεονέκτημα του SWE είναι ότι προσθέτει ένα επίπεδο αφαίρεσης ανάμεσα στην εφαρμογή ή τον χρήστη και τους ίδιους τους αισθητήρες κρύβοντας την πολυπλοκότητα της υποδομής και των πρωτοκόλλων. Έτσι οι εφαρμογές αναπτύσσονται ανεξάρτητα από τα κατώτερα επίπεδα. Επίσης μειώνει την

πολυπλοκότητα των αισθητήρων καθώς θα πρέπει να είναι συμβατοί και να επικοινωνούν μόνο με την SWE υποδομή χωρίς να τους ενδιαφέρουν τα ανώτερα επίπεδα. Λόγω των διαφόρων παρεχόμενων προτύπων και δυνατοτήτων, είναι δυνατό να υλοποιηθούν εύκολα περίπλοκες και απαιτητικές εφαρμογές πραγματικού χρόνου, οι οποίες μπορούν να απεικονίζουν διαφόρων τύπων δεδομένα και ειδοποιήσεις πάνω σε δυναμικούς χάρτες.

2.1.10 Ανακάλυψη αισθητήρων

Η ανακάλυψη των αισθητήρων είναι ένα από τα πιο σημαντικά κομμάτια σε τέτοιου είδους δίκτυα. Γενικά θα μπορούσαμε να πούμε ότι υπάρχουν δύο τύποι ανακάλυψης [54], η ανακάλυψη στιγμιότυπου αισθητήρα και η ανακάλυψη υπηρεσίας αισθητήρα. Η πρώτη αναφέρεται στην αναζήτηση κάποιου συγκεκριμένου αισθητήρα ή δικτύου αισθητήρων. Αντίθετα, η δεύτερη αναζητάει οποιονδήποτε αισθητήρα ή ομάδα αισθητήρων μπορεί να προσφέρει κάποια συγκεκριμένη υπηρεσία. Παρόλο που η προσέγγιση είναι διαφορετική, ο κάθε τρόπος μπορεί να χρησιμοποιεί τον άλλο. Για παράδειγμα, αν συνδυάσουμε την ανακάλυψη στιγμιότυπου αισθητήρα με τις πληροφορίες που μπορούμε να λάβουμε από την υποδομή για τις δυνατότητες του κάθε αισθητήρα, τότε σχετικά εύκολα μπορούμε να παρέχουμε και ανακάλυψη υπηρεσίας.

Κατά την ανακάλυψη υπάρχει η δυνατότητα για πέρασμα παραμέτρων οι οποίες μπορεί να καθορίζουν τις ικανότητες που μας ενδιαφέρουν (π.χ. μέτρηση θερμοκρασίας), τις γεωγραφικές περιοχές που επιθυμούμε να είναι τοποθετημένοι οι αισθητήρες αλλά και χρονικούς περιορισμούς ως προς την αποκρισιμότητα αλλά και τα διαθέσιμα ιστορικά δεδομένα.

Στην ανακάλυψη των αισθητήρων μπορεί να βοηθήσει το πρότυπο IEEE 1451 (γίνεται περιγραφή σε επόμενο κεφάλαιο) αντιστοιχίζοντας τις πληροφορίες που περιέχουν τα TEDS¹ (Transducer Electronic Data Sheet) σε δομή συμβατή με την SensorML [55]. Η βασική ιδέα είναι να υλοποιηθεί μία αντιστοίχιση των πληροφοριών και των ιδιοτήτων που υπάρχουν στα TEDS σε συμβατή περιγραφή με τη χρήση της SensorML ώστε οι

¹ Τα TEDS είναι μία μέθοδος αποθήκευσης πληροφοριών σχετικές με τους αισθητήρες και τις δυνατότητές τους η οποία περιγράφεται στο πρότυπο IEEE 1451

διαθέσιμες πληροφορίες να μπορούν να περάσουν εύκολα στο ανώτερο επίπεδο του SWE και να μπορούν να επεξεργαστούν και να χρησιμοποιηθούν.

2.2 OSWA (Open Sensor Web Architecture)

Η SWE περιγράφει κυρίως τις οντότητες, τις υπηρεσίες και τη μεταξύ τους επικοινωνία. Ο τρόπος υλοποίησης των οντοτήτων όμως είναι ένα πάρα πολύ σημαντικό κομμάτι το οποίο καθορίζει την αποτελεσματικότητα, απόδοση και ασφάλεια του συστήματος. Η πιο σημαντική υλοποίηση των προτύπων αυτών είναι η Ανοιχτή Αρχιτεκτονική Ιστού Αισθητήρων (Open Sensor Web Architecture - OSWA) [23], η οποία σε αντίθεση με ανάλογες αρχιτεκτονικές, μπορεί να υποστηρίξει διάφορες ετερογενείς τεχνολογίες αισθητήρων (SunSPOT, TinyOS, Linux).

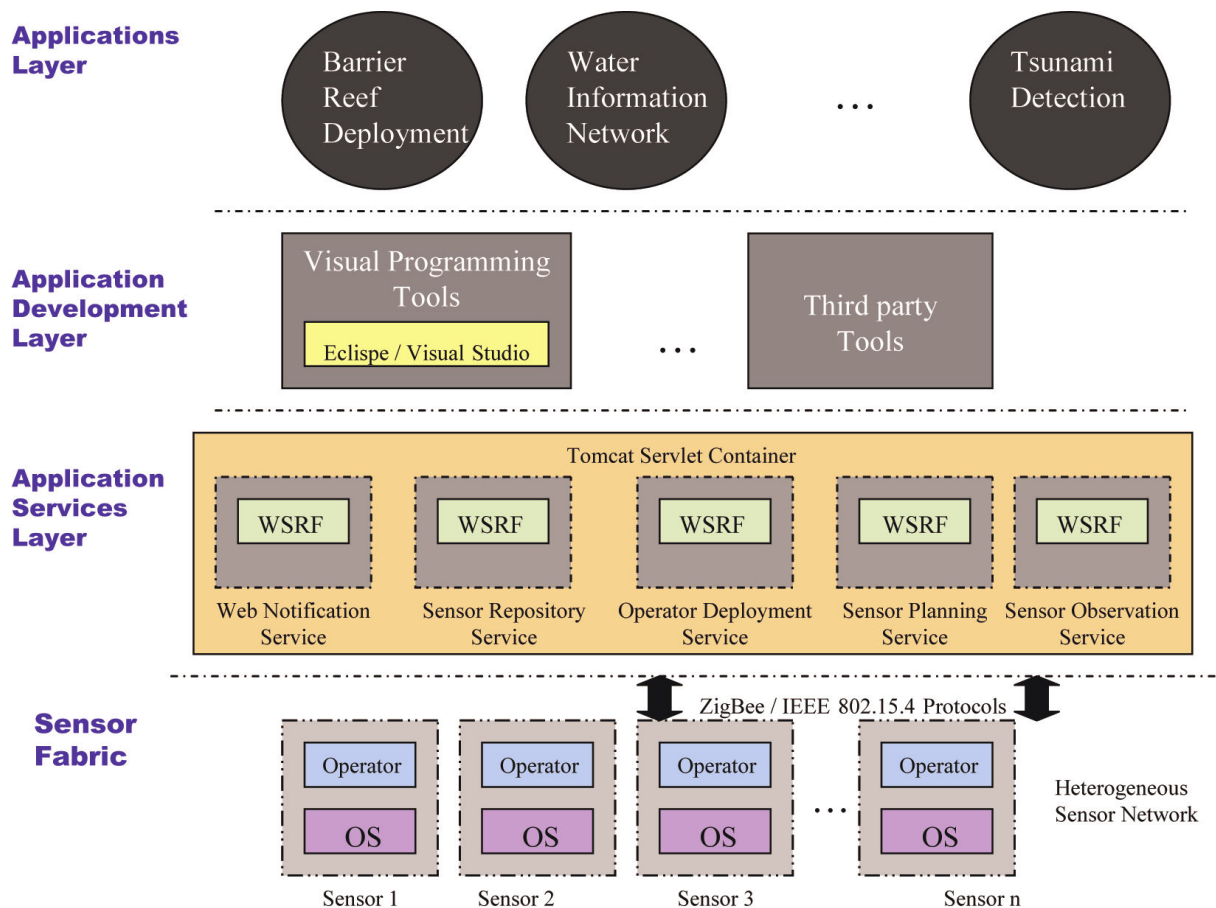
2.2.1 Αρχιτεκτονική

Η OSWA αποτελείται από τέσσερα επίπεδα, παρατηρώντας την από κάτω προς τα πάνω έχουμε:

- **Δικτυοδομή αισθητήρων (sensor fabric):** είναι το επίπεδο στο οποίο βρίσκονται οι διαφορετικοί τύποι αισθητήρων. Ο κώδικας των υπηρεσιών που εκτελείται σε αυτό το επίπεδο είναι διαφορετικός για κάθε ξεχωριστό τύπο αισθητήρων. Η λύση σε αυτό το πρόβλημα θα ήταν η ανάπτυξη κάποιου ενδιάμεσου λογισμικού (middleware), το οποίο θα μας επέτρεπε να επικοινωνούμε με αυτό το επίπεδο με κοινό και προκαθορισμένο τρόπο, ανεξαρτήτου του τύπου του αισθητήρα. Η επικοινωνία με το από πάνω επίπεδο γίνεται μέσω ασύρματων πρωτοκόλλων.
- **Υπηρεσίες εφαρμογών:** είναι εκεί όπου υπάρχουν οι βασικές οντότητες της αρχιτεκτονικής τις οποίες έχουμε περιγράψει παραπάνω. Για να αυξηθούν οι δυνατότητες, οι οντότητες αυτές είναι βασισμένες πάνω στο Πλαίσιο Πόρων Υπηρεσιών Ιστού (WSRF - Web Services Resource Framework). Το WSRF είναι ένα πλαίσιο το οποίο επιτρέπει σε υπηρεσίες ιστού να γίνουν διατηρούν κατάσταση (stateful), δηλαδή να έχουν τη δυνατότητα να διατηρούν δεδομένα και πληροφορίες μεταξύ διαφορετικών αιτήσεων. Αυτό διευκολύνει και προσθέτει πολλές νέες δυνατότητες στην επικοινωνία με τους πελάτες, καθώς σε κάθε αίτηση μπορούν να προσθέτουν κάποιο χαρακτηριστικό αναγνώρισής τους, οπότε η υπηρεσία μπορεί να ανακαλέσει προηγούμενες συνομιλίες με αυτό τον πελάτη ως τότε και να

απαντήσει ανάλογα. Επίσης η διατήρηση κατάστασης επιτρέπει σε πελάτες να μπορούν να εγγραφούν σε κάποια υπηρεσία, ώστε να ενημερώνονται όποτε συμβεί κάποια αλλαγή σε μετρήσεις ή όποτε κάποιες μετρήσεις ξεπεράσουν κάποια όρια. Αυτό εξαλείφει την ανάγκη του πελάτη να ρωτάει ανά τακτά χρονικά διαστήματα την υπηρεσία, κάτι το οποίο μειώνει το δικτυακό φόρτο και αυξάνει τη συνολική απόδοση του συστήματος.

- **Ανάπτυξη εφαρμογών:** εδώ κατατάσσονται τα εργαλεία ανάπτυξης εφαρμογών, οι οποίες μπορούν έπειτα να τρέξουν πάνω από τη συγκεκριμένη υποδομή.
- **Εφαρμογή:** είναι το ανώτερο επίπεδο στο οποίο εκτελούνται οι εφαρμογές



Σχήμα 8: Αρχιτεκτονική OSWA

Οι τεχνολογίες που χρησιμοποιούνται στο OSWA, κάνουν χρήση κυρίως της γλώσσας προγραμματισμού Java και της XML. Αυτό προσφέρει πολλές δυνατότητες και πλεονεκτήματα, με σημαντικότερο το ότι ο ίδιος κώδικας μπορεί να εκτελεστεί σε ένα μεγάλο εύρος διαφορετικών αρχιτεκτονικών και υποδομών. Επίσης οι περισσότεροι

κατασκευαστές παρέχουν βιβλιοθήκες για προγραμματισμό των αισθητήρων τους σε Java (Crossbow, Sun). Βέβαια το μεγάλο μειονέκτημα της Java είναι η μειωμένη της ταχύτητα, σε σχέση για παράδειγμα με τη C, και η μεγαλύτερη χρήση μνήμης. Τα παραπάνω επηρεάζουν τη συνολική απόδοση της υποδομής συνήθως μόνο σε αυξημένο αριθμό ταυτόχρονων αιτήσεων.

3. Η Οικογένεια Προτύπων IEEE 1451

Το IEEE 1451 περιγράφει μία ομάδα ανοιχτών και ανεξάρτητων από το δίκτυο διεπαφών, για τη διασύνδεση μορφοτροπέων με όργανα, συστήματα και δίκτυα μέσω αναλογικών, ψηφιακών, ενσύρματων ή ασύρματων διεπαφών.

Ο κυριότερος στόχος του προτύπου είναι ο διαχωρισμός των μορφοτροπέων από το δίκτυο, ώστε η ανάπτυξη δικτύων και διεπαφών να είναι ανεξάρτητες από εταιρίες και κατασκευαστές. Επίσης θα πρέπει να επιτρέπει στους μορφοτροπείς να μπορούν να αντικατασταθούν και να μεταφερθούν με την ελάχιστη δυνατή προσπάθεια, υποστηρίζοντας παράλληλα θερμή εναλλαγή (hot swap). Θα πρέπει να ελαχιστοποιεί την χειροκίνητη παραμετροποίηση η οποία είναι επιρρεπής σε λάθη και σφάλματα. Επίσης θα πρέπει να υποστηρίζει ένα γενικό μοντέλο δεδομένων, ελέγχου, χρονισμού, παραμετροποίησης και βαθμονόμησης για τους μορφοτροπείς. Τέλος θα πρέπει να αναπτυχθούν κάποιες δομές οι οποίες κατά προτίμηση θα είναι αποθηκευμένες μέσα στους μορφοτροπείς και θα περιέχουν πληροφορίες για αυτούς.

Σύμφωνα με το πρότυπο, κάθε ευφυής αισθητήρας αποτελείται από δύο κύριες οντότητες, ένα Άρθρωμα Διεπαφής Μορφοτροπείας (Transducer Interface Module – TIM) και έναν Επεξεργαστή Εφαρμογών με Δυνατότητα Δικτύωσης (Network Capable Application Processor – NCAP). Κάθε TIM μπορεί να αποτελείται από έναν ή περισσότερους μορφοτροπείς (έως 255), μονάδες επεξεργασίας σήματος, μετατροπείς σήματος (από αναλογικό σε ψηφιακό και αντίστροφα) και μία διεπαφή μέσω της οποίας μπορεί να επικοινωνεί με το NCAP. Το NCAP είναι ένα σύστημα (συνήθως περιλαμβάνει κάποιο μικροεπεξεργαστή και ενσωματωμένο λειτουργικό σύστημα) το οποίο διασυνδέει ένα ή περισσότερα TIM, μέσω του δικτύου επικοινωνίας, με το χρήστη ή την εφαρμογή και μπορεί να έχει πολλές και διαφορετικού τύπου διεπαφές με τον έξω κόσμο. Με τη χρήση των Ηλεκτρονικών Φύλλων Δεδομένων Μορφοτροπείας (Transducer Electronic Data Sheet – TEDS) (τα οποία συνήθως βρίσκονται αποθηκευμένα μέσα στο TIM), είναι πολύ εύκολη η λήψη και αποκωδικοποίηση δεδομένων από τους μορφοτροπείς, καθώς περιέχουν πληροφορίες σχετικές με τους διαθέσιμους μορφοτροπείς αλλά και γενικότερα στοιχεία για ολόκληρο το TIM. Είναι ίσως το πιο σημαντικό κομμάτι του συστήματος, καθώς έκανε τη μεγάλη διαφορά σε σχέση με παρόμοια συστήματα. Επίσης τα TIM έχουν τη δυνατότητα της άμεσης βυσμάτωσης και λειτουργίας (plug and play), κάτι το οποίο σημαίνει ότι μπορούμε να έχουμε άμεση προσπέλαση σε αυτά και τα δεδομένα των αισθητήρων τους, αμέσως μετά της σύνδεσής τους με το NCAP.

3.1 Ιστορική αναδρομή

Εταιρίες που ασχολούνταν με τον σχεδιασμό, κατασκευή και ανάπτυξη αισθητήρων [2] και περιφερειακών συστημάτων για αυτούς, αλλά και διάφοροι κυβερνητικοί φορείς όπως το Ινστιτούτο Ηλεκτρολόγων και Ηλεκτρονικών Μηχανικών (Institute of Electrical and Electronics Engineers – IEEE), Εθνικό Ινστιτούτο Προτύπων και Τεχνολογίας των Ηνωμένων Πολιτειών Αμερικής (U.S. National Institute of Standards and Technology – NIST) και τα Εργαστήρια του Υπουργείου Ενέργειας των Ηνωμένων Πολιτειών Αμερικής (U.S. Department of Energy Laboratories), άρχισαν να συζητάνε την πιθανότητα έναρξης κάποιας οργανωμένης προσπάθειας για την ανάπτυξη ενός κοινού προτύπου για αυτά τα συστήματα. Οι συζητήσεις ξεκίνησαν τον Σεπτέμβριο του 1993 που πραγματοποιήθηκε η σύνοδος TC-9 της IEEE (IEEE Instrumentation and Measurement Society Technical Committee on Sensors). Τα επόμενα δύο χρόνια πραγματοποιήθηκαν πέντε IEEE/NIST workshops και τέσσερις σύνοδοι ομάδων εργασίας της IEEE, οι οποίες έθεσαν τα θεμέλια προς αυτό το μεγάλο στόχο. Επίσης οργανώθηκαν τρεις επιτροπές οι οποίες ήταν υπεύθυνες για τη μελέτη των δικτύων, της επικοινωνίας των μορφοτροπέων με τα υπόλοιπα συστήματα και τις απαιτήσεις της αγοράς. Αποτέλεσμα των παραπάνω, ήταν η σύνταξη δύο σχεδίων προτύπου μέχρι το τέλος του 1995. Το πρώτο ήταν το IEEE 1451.1 στο οποίο περιγραφόταν η βασική δομή των TEDS και το IEEE 1451.2 στο οποίο περιγραφόταν το κοινό μοντέλο αντικειμένων.

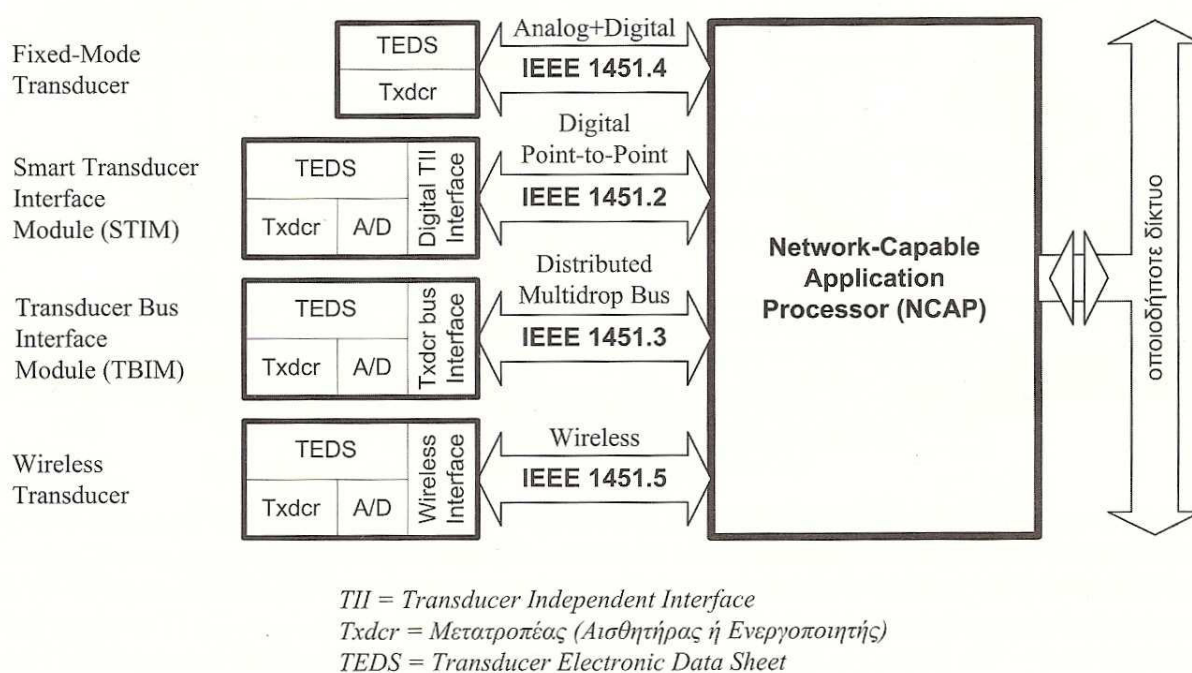
Στις αρχές του 1996, η IEEE ενέκρινε δύο ομάδες εργασίας για να αναπτύξουν πρότυπα για ευφυείς μορφοτροπείς. Η IEEE 1451.1 θα καθόριζε ένα κοινό μοντέλο αντικειμένων για τη δικτυακή επικοινωνία μεταξύ των μορφοτροπέων και των NCAP και η IEEE 1451.2 θα καθόριζε τα TEDS, τα TIM και την Ανεξάρτητη Διεπαφή Μορφοτροπέα (Transducer Independent Interface – TII) η οποία είναι μία διεπαφή δέκα σημάτων και χρησιμοποιείται για την επικοινωνία NCAP και TIM. Η δουλειά των ομάδων αυτών, ανέδειξε την ανάγκη ανάπτυξης ενός ακόμα προτύπου κατάλληλο για τους χρήστες μικρών αλλά και κατανεμημένων συστοιχιών αισθητήρων.

Το 1997 δημοσιεύθηκε το πρώτο πρότυπο από την ομάδα IEEE 1451.2 με τίτλο “IEEE Std 1451.2-1997 IEEE Standard for a Smart Transducer Interface for Sensors and Actuators - Transducer to Microprocessor Communication Protocols and Transducer Electronic Data Sheet (TEDS) Formats.” το οποίο το 2004 υπέστη αναθεώρηση. Έπειτα εγκρίθηκαν άλλες δύο ομάδες εργασίας, μία για πολλαπλά συνδεδεμένους αισθητήρες (IEEE 1451.3) και μία για αισθητήρες με αναλογικό και ψηφιακό τρόπο λειτουργίας

(IEEE 1451.4). Το 2001 δημιουργήθηκε η IEEE P1451.5 που ασχολήθηκε με ασύρματους αισθητήρες.

3.2 Πρότυπα

Το IEEE 1451 αποτελεί μία οικογένεια προτύπων [9], [29] τα οποία περιγράφουν είτε διαφορετικά επίπεδα του πρωτοκόλλου ή διαφορετικούς τύπους διεπαφών. Στο Σχήμα 9 υπάρχει μία σύνοψη των προτύπων αυτών και έπειτα ακολουθεί μία περιληπτική περιγραφή τους.



Σχήμα 9: Αρχιτεκτονική του IEEE 1451

3.2.1 IEEE 1451.0

Περιγράφει ένα σύνολο από κοινές εντολές και λειτουργίες, με τη χρήση των οποίων μπορεί να υπάρξει πρόσβαση και επικοινωνία με κάθε μορφοτροπέα συμβατό με το πρότυπο, ανεξαρτήτου του τρόπου διασύνδεσής του με το σύστημα. Με αυτό τον τρόπο, χρησιμοποιώντας κάποιο από τα υπάρχοντα ή μελλοντικά 1451.X πρότυπα τα οποία περιγράφουν το φυσικό στρώμα, μπορεί να εξασφαλιστεί η ορθή επικοινωνία με ενσύρματους αλλά και ασύρματους μορφοτροπείς. Επίσης καθορίζει την κοινή μορφή και τα περιεχόμενα των TEDS.

Είναι από τα τελευταία πρότυπα που οριστικοποιήθηκαν [5] και ο κύριος σκοπός του είναι να διευκολύνει τη διαλειτουργικότητα μεταξύ των υπαρχών προτύπων του IEEE 1451. Για να το πετύχει αυτό όρισε κάποιες βασικές αρχές:

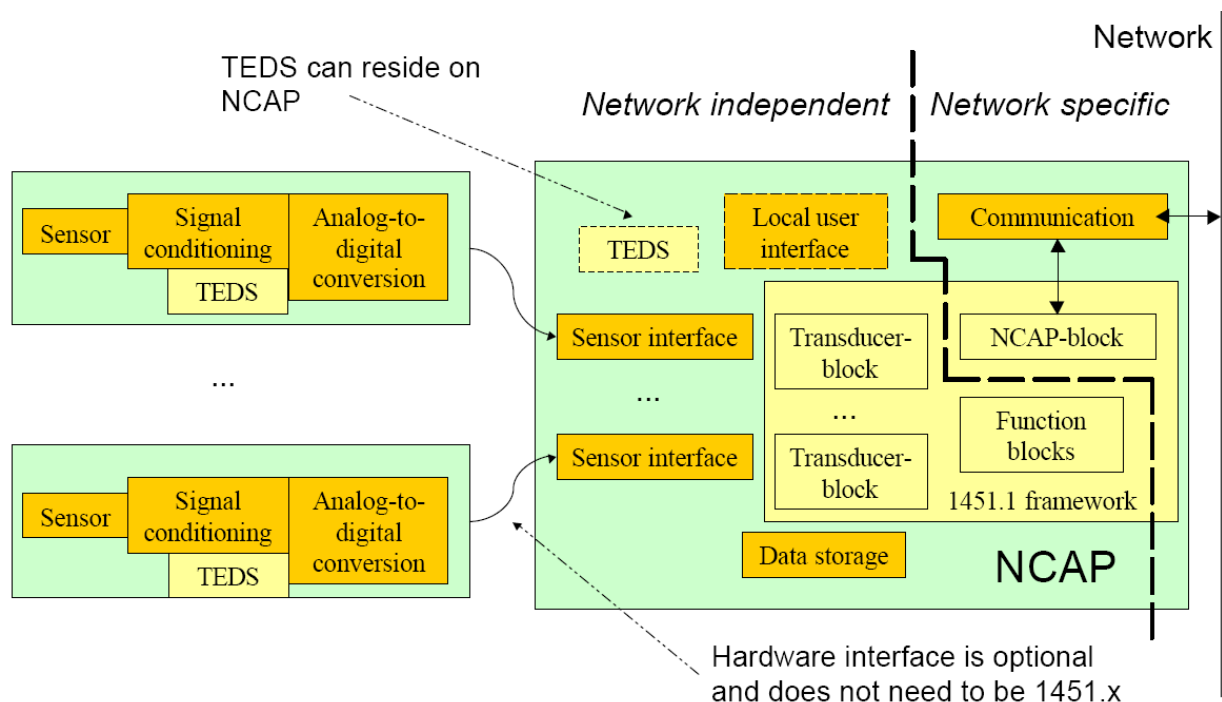
- θα πρέπει να υποστηρίζονται οι ήδη υπάρχουσες βασικές οντότητες του IEEE 1451 οι οποίες περιγράφονται και υποστηρίζονται από τα υπόλοιπα πρότυπα
- τα πρότυπα πρέπει να περιγράφουν τις διεπαφές μεταξύ των δομοστοιχείων (modules) και όχι τα ίδια τα δομοστοιχεία. Για να το καταφέρουν αυτό, πρέπει να συμπεριφέρονται στα δομοστοιχεία σαν μαύρα κουτιά που δεν τους ενδιαφέρει η εσωτερική τους δομή και λειτουργία, αλλά ορίζουν με λεπτομέρειες τη λειτουργία τους και τις εξόδους κάτω υπό συγκεκριμένες συνθήκες, εισόδους και μηνύματα
- πρέπει να υπάρχει διαχωρισμός των επιπέδων και των διεπαφών. Κάθε δομοστοιχείο επικοινωνεί αποκλειστικά με το επίπεδο ακριβώς από πάνω του και κάτω του, χωρίς να το ενδιαφέρει η υπόλοιπη αρχιτεκτονική του υπάρχοντος συστήματος
- πρέπει να υποστηρίζεται η θερμή εναλλαγή (hot swapping) των TIM χωρίς να χρειάζεται να βγει από την τροφοδοσία το NCAP στο οποίο είναι συνδεδεμένα ή πρόκειται να συνδεθούν. Προφανώς ύστερα από μία τέτοια διαδικασία το NCAP θα πρέπει να συνεχίζει να λειτουργεί κανονικά, χωρίς προβλήματα ή βλάβες

Τα κυριότερα πλεονεκτήματα από την ανάπτυξη του συγκεκριμένου προτύπου είναι:

- μείωση στην επικάλυψη μεταξύ των προτύπων οπότε και της συνολικής απαιτούμενης δουλειάς
- αύξηση της συμβατότητας και των κοινών χαρακτηριστικών των επιμέρους προτύπων
- κοινή πρόσβαση μέσω δικτύου για όλες τις συσκευές
- ευκολία προσθήκης νέου προτύπου το οποίο θα υποστηρίζει κάποιο νέο ή μη υποστηριζόμενο μέχρι στιγμής φυσικό μέσο
- τυποποίηση του μηνύματος και όχι του μέσου

3.2.2 IEEE 1451.1

Καθορίζει ένα κοινό μοντέλο αντικειμένων το οποίο περιγράφει τη συμπεριφορά των ευφυών μορφοτροπέων. Επίσης ορίζει τα μοντέλα επικοινωνίας τα οποία χρησιμοποιούνται στο πρότυπο, στα οποία περιλαμβάνονται το πελάτης-εξυπηρετητής (client-server) και δημοσιοποίηση-εγγραφή (publish-subscribe). Με τη χρήση του, εφαρμογές οι οποίες μπορούν να εκτελούνται στο NCAP, μπορούν να επικοινωνούν με κοινό τρόπο με όλους τους συμβατούς μορφοτροπείς, οι οποίοι είναι συνδεδεμένοι με το σύστημα. Επίσης δίνει τη δυνατότητα για κοινό τρόπο επικοινωνίας μεταξύ των NCAP αλλά και μεταξύ των NCAP και των εξωτερικών συστημάτων.



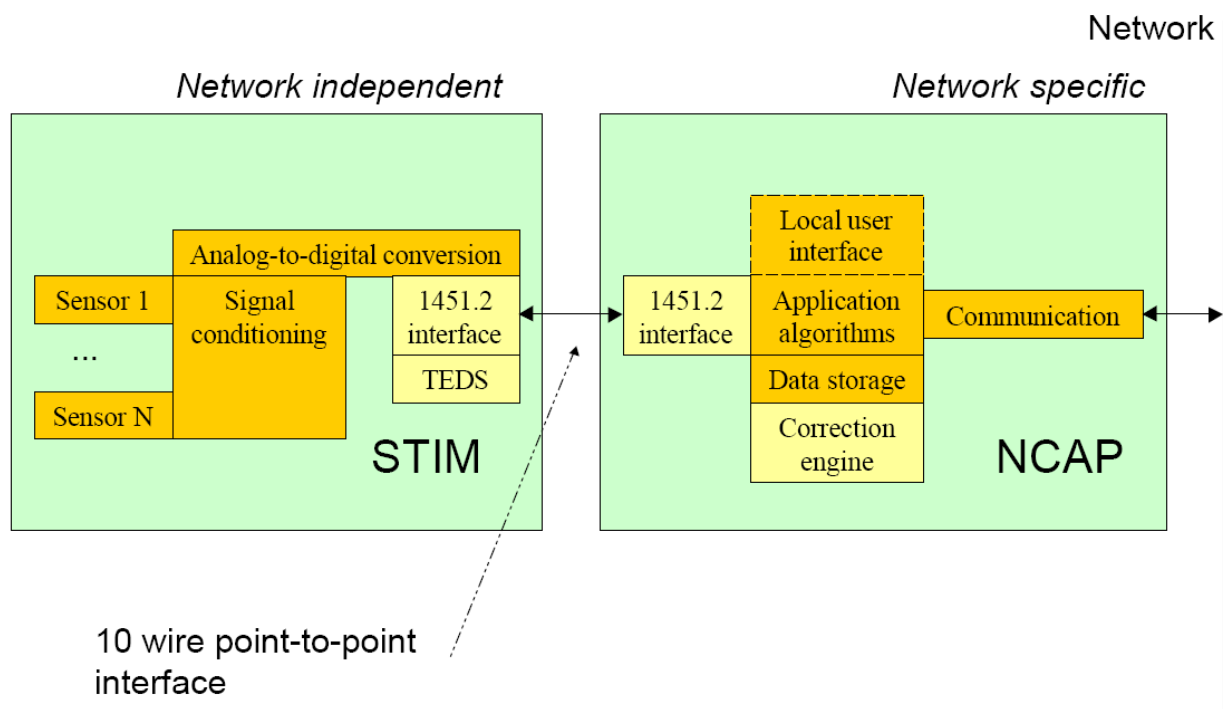
Σχήμα 10: Αρχιτεκτονική του IEEE 1451.1

3.2.3 IEEE 1451.2

Καθορίζει τη διεπαφή μεταξύ ενός TIM και ενός NCAP και τα TEDS για σύνδεση σημείο προς σημείο. Το αρχικό πρότυπο περιέγραφε ένα επίπεδο επικοινωνίας το οποίο βασιζόταν στη χρήση Σειριακής Περιφερειακής Διεπαφής (Serial Peripheral Interface – SPI) στην οποία είχαν προστεθεί κάποιες επιπλέον γραμμές επικοινωνίας για έλεγχο της ροής των δεδομένων και του χρονισμού. Έπειτα από αναθεώρηση, προστέθηκε η υποστήριξη και για σειριακή θύρα. Εφόσον όλα τα TIM είναι σχεδιασμένα ώστε να υποστηρίζουν τη συγκεκριμένη διεπαφή, το μόνο που χρειάζεται είναι να

χρησιμοποιήσουμε τον κατάλληλο τύπο NCAP το οποίο να υποστηρίζει τη δικτυακή υποδομή η οποία μας ενδιαφέρει.

Όπως παρατηρούμε και στο Σχήμα 11, το TIM είναι η διεπαφή μεταξύ TIM-NCAP. Η πρόσβαση στο TIM γίνεται ανάλογα με συστήματα που έχουν τις λειτουργίες και τα δεδομένα χαρτογραφημένα στη μνήμη. Δηλαδή κάθε ενέργεια (διάβασμα από αισθητήρα, εντολή σε εκκινητήρα) είναι αντιστοιχημένη σε μία διεύθυνση, την οποία στέλνει το NCAP στο TIM όταν θέλει να πραγματοποιηθεί η συγκεκριμένη ενέργεια.



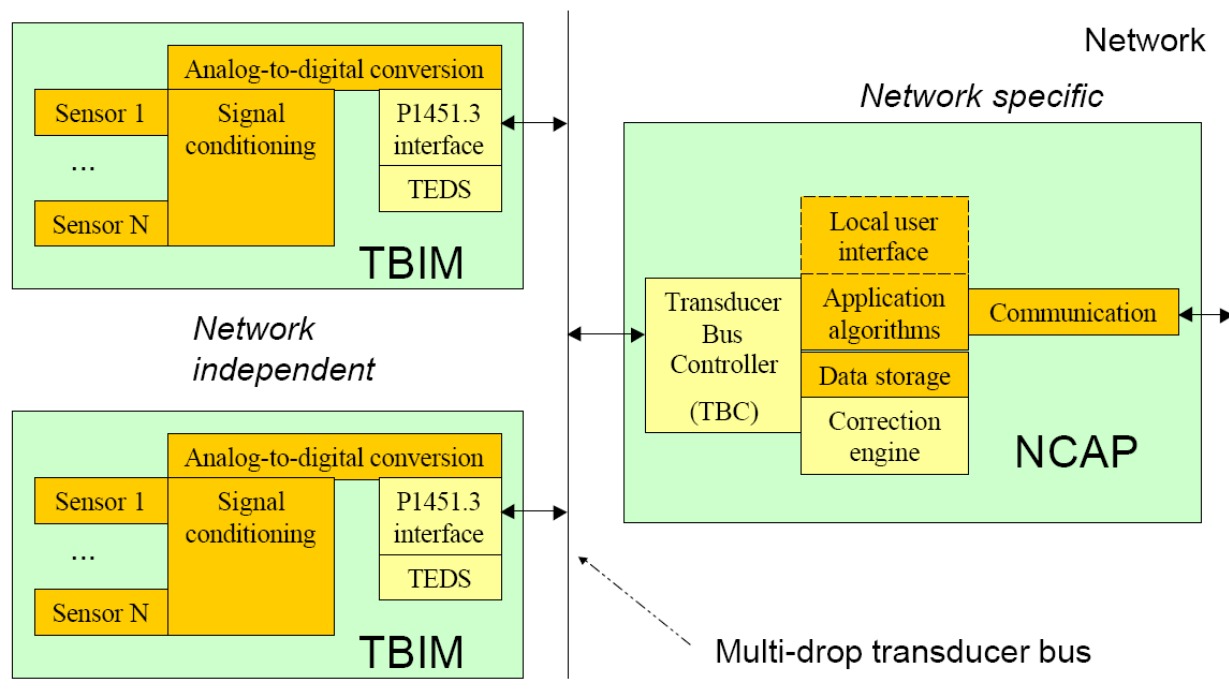
Σχήμα 11: Αρχιτεκτονική του IEEE 1451.2

3.2.4 IEEE 1451.3

Αποτελεί επέκταση του IEEE 1451.2 καθορίζοντας τη διεπαφή μεταξύ ενός NCAP και πολλών TIM και τα TEDS για μορφοτροπίες οι οποίοι επικοινωνούν όλοι μέσω ενός κοινού διαύλου χρησιμοποιώντας μία κατανεμημένη αρχιτεκτονική.

Το πρότυπο περιγράφει τα Αρθρώματα Διεπαφής Διαύλου Μετατροπέα (Transducer Bus Interface Modules – TBIM), τα οποία συνδέονται μέσω ενός κοινού διαύλου με το NCAP, το οποίο τα συνδέει με το επιθυμητό εξωτερικό δίκτυο. Ο κοινός δίαυλος αποτελείται από μία γραμμή σύνδεσης που χρησιμοποιείται τόσο για τη μεταφορά ηλεκτρικής ενέργειας για την τροφοδοσία των TBIM, όσο και για τη μεταφορά δεδομένων. Ο κοινός δίαυλος ελέγχεται από ένα ελεγκτή διαύλου, ο οποίος

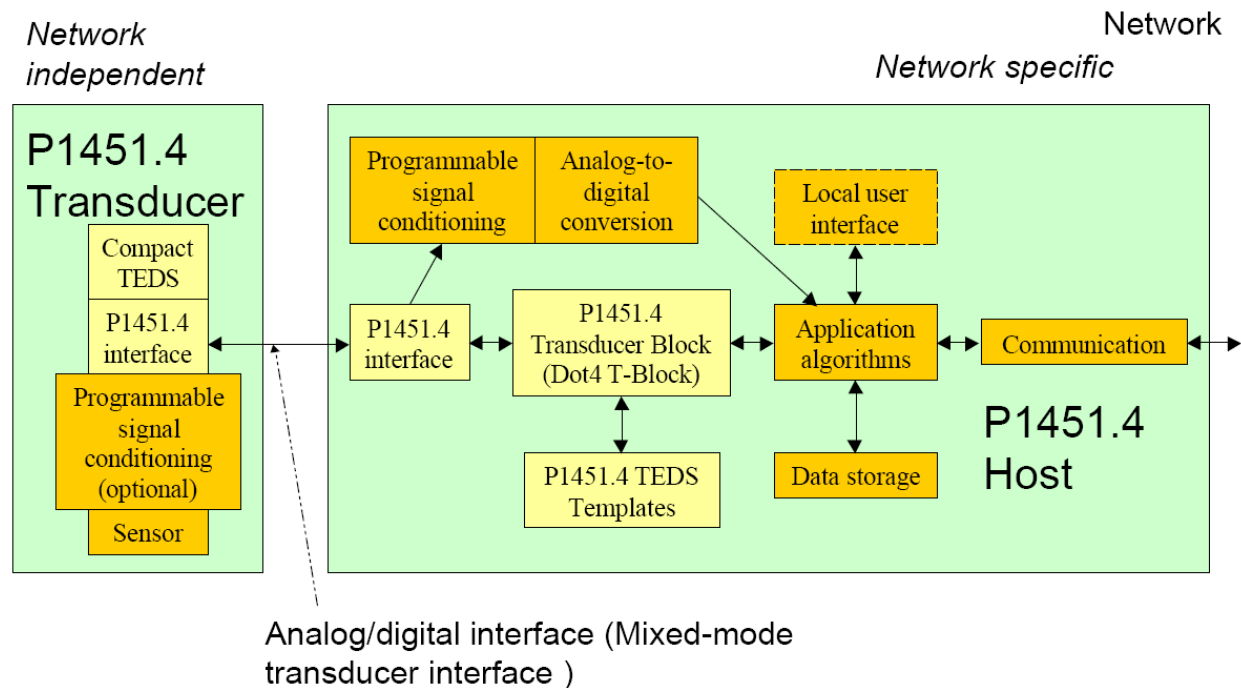
εμπεριέχεται μέσα στο NCAP και ονομάζεται Ελεγκτής Διαύλου Μετατροπέα (Transducer Bus Controller - TBC).



Σχήμα 12: Αρχιτεκτονική του IEEE 1451.3

3.2.5 IEEE 1451.4

Καθορίζει μία διεπαφή για αναλογικούς μορφοτροπές οι οποίοι έχουν δυνατότητα για αναλογική μεταφορά μετρήσεων αλλά ψηφιακή μεταφορά TEDS. Επειδή και τα αναλογικά αλλά και τα ψηφιακά δεδομένα μεταφέρονται πάνω από την ίδια γραμμή επικοινωνίας, κάθε χρονική στιγμή είναι ενεργός μόνο ένας από τους δύο τρόπους μεταφοράς. Υπάρχει δυνατότητα και για διαχωρισμό των αναλογικών από τα ψηφιακά δεδομένα, προσθέτοντας δύο ακόμα σήματα για τη ψηφιακή μεταφορά των TEDS. Επίσης περιγράφει κάποια τροποποιημένα TEDS, τα οποία περιλαμβάνουν λιγότερες πληροφορίες, ώστε να μπορούν να αποθηκευτούν στις σχετικά μικρές μνήμες των μικροσκοπικών αισθητήρων. Τα συγκεκριμένα TEDS είναι ικανά να περιγράψουν αρκετά διαφορετικά είδη μορφοτροπέων όπως αισθητήρες επιτάχυνσης, θερμοκρασίας, ρεύματος, μικρόφωνα.



Σχήμα 13: Αρχιτεκτονική του IEEE 1451.4

3.2.6 IEEE 1451.5

Καθορίζει τη διεπαφή ανάμεσα σε ασύρματους μορφοτροπείς και NCAP. Τα ασύρματα πρωτόκολλα που υποστηρίζονται είναι τα 802.11 (WiFi), 802.15.1 (Bluetooth) και 802.15.4 (ZigBee).

3.2.7 IEEE 1451.6

Καθορίζει τη διεπαφή ανάμεσα σε μορφοτροπείς και NCAP οι οποίοι συνδέονται με το υψηλής ταχύτητας Ανοιχτό Δίκτυο Ελεγκτή (Controller Area Network open – CANopen). Ορίζει την αντιστοίχιση των στοιχείων των TEDS με τις καταχωρήσεις του CANopen αλλά και με τα μηνύματα επικοινωνίας, τις παραμέτρους ρύθμισης και τις διαγνωστικές πληροφορίες. Επίσης υιοθετεί το μοντέλο συσκευών του CANopen για να περιγράψει τα όργανα μέτρησης.

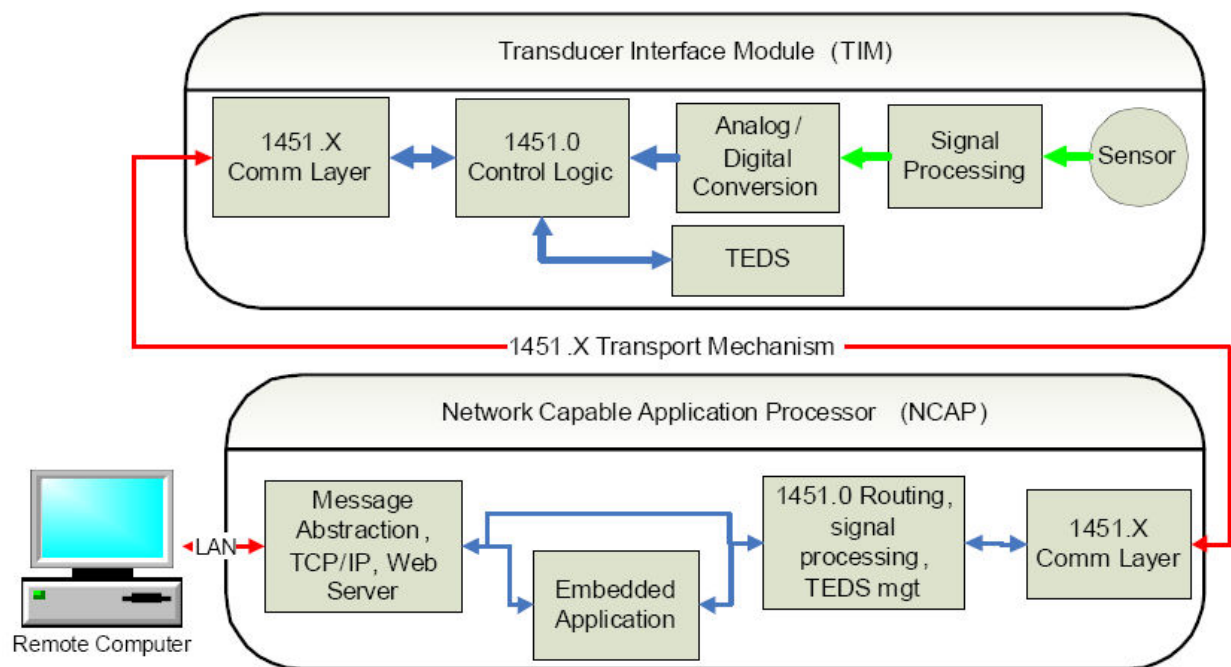
3.2.8 IEEE 1451.7

Περιγράφει τη διεπαφή ανάμεσα σε μορφοτροπείς οι οποίοι επικοινωνούν με το υπόλοιπο σύστημα με Ραδιοσυχνотική Αναγνώριση (Radio Frequency Identification –

RFID). Περιγράφει τις μεθόδους επικοινωνίας, τη μορφή των δεδομένων αλλά και τα TEDS για αυτού του είδους τους μορφοτροπείς.

3.3 Λειτουργία

Η λειτουργία του IEEE 1451 [10], βασίζεται κυρίως πάνω σε δύο οντότητες, το TIM και το NCAP. Στο TIM είναι συνδεδεμένοι οι μορφοτροπείς (αισθητήρες και εκκινητήρες) και επίσης περιέχει τα TEDS τα οποία περιέχουν πληροφορίες για τους μορφοτροπείς και τον τρόπο επικοινωνίας μαζί τους, αλλά και για το ίδιο το TIM. Το NCAP είναι η κεντρική οντότητα το οποίο μπορεί να επικοινωνεί με τα TIM με διάφορα ενσύρματα αλλά και ασύρματα πρωτόκολλα.



Σχήμα 14: Σχηματικό διάγραμμα του IEEE 1451

Όταν το NCAP αρχίζει να λειτουργεί, αρχικά αναζητά TIM τα οποία πιθανόν να είναι συνδεδεμένα πάνω του. Για κάθε ένα από αυτά, μεταφέρει ένα αντίγραφο των TEDS που περιέχουν, σε μία προσωρινή μνήμη η οποία βρίσκεται μέσα στο NCAP. Όταν κάποιος πελάτης ή κάποια εφαρμογή κάνει κάποια αίτηση στο NCAP για διάβασμα της τιμής κάποιου αισθητήρα, το NCAP θα στείλει τις απαραίτητες εντολές στο ανάλογο TIM, το οποίο θα επιστρέψει την τιμή σε ψηφιακή μορφή, οπότε το NCAP με τη βοήθεια των TEDS θα μπορέσει να τη μορφοποιήσει και να την μετατρέψει σε συμβατή μορφή

με το διεθνές σύστημα μονάδων SI (Système International). Έπειτα δημιουργεί ένα XML έγγραφο το οποίο περιέχει την τιμή μαζί με κάποια άλλα δεδομένα και το αποστέλλει μέσω δικτύου με τη χρήση του HTTP πρωτοκόλλου.

Όπως έχουμε ήδη αναφέρει, λόγω του κοινού προτύπου, οντότητες από διαφορετικές εταιρίες μπορούν να συνεργαστούν χωρίς πρόβλημα. Επίσης οι αισθητήρες είναι plug and play όπως και τα TIM. Για παράδειγμα αν επιθυμούμε να προσθέσουμε έναν ασύρματο αισθητήρα σε ένα υπάρχον σύστημα, το μόνο που έχουμε να κάνουμε είναι απλά να τον τοποθετήσουμε εντός εμβέλειας του ασύρματου δικτύου.

3.4 TEDS

Τα TEDS [8], [27] είναι μία μέθοδος αποθήκευσης πληροφοριών οι οποίες είναι απαραίτητες για την αναγνώριση ενός μορφοτροπέα αλλά και για τον τρόπο επικοινωνίας μαζί του (π.χ. ορθή ανάγνωση της τιμής ενός αισθητήρα). Τα TEDS περιέχουν κρίσιμες πληροφορίες οι οποίες είναι απαραίτητες σε ένα όργανο ή σύστημα μέτρησης ώστε να αναγνωρίσει, να επικοινωνήσει και να χρησιμοποιήσει σωστά το σήμα από τον μορφοτροπέα. Οι πληροφορίες αυτές μπορεί να περιλαμβάνουν παράγοντες βαθμονόμησης (calibration factors), μονάδες μέτρησης, εύρος τιμών και διάφορους χρόνους σχετικούς με τον αισθητήρα (προθέρμανσης, απόκρισης, δειγματοληψίας). Επίσης μπορούν να περιλαμβάνουν στοιχεία όπως το όνομα του κατασκευαστή, την ημερομηνία κατασκευής και κάποιο σειριακό αριθμό. Τα TEDS μπορούν να βρίσκονται αποθηκευμένα είτε σε κάποια ενσωματωμένη μνήμη, συνήθως σε Ηλεκτρικά Σβέσιμη Προγραμματίσιμη Μνήμη Μόνο Ανάγνωσης (Electrically Erasable Programmable Read-Only Memory – EEPROM), μέσα στον transducer ή σε κάποιο ανεξάρτητο αρχείο διαθέσιμο στο διαδίκτυο. Στη δεύτερη περίπτωση ονομάζονται εικονικά TEDS (Virtual TEDS). Τα Virtual TEDS δίνουν τη δυνατότητα ακόμα και σε παραδοσιακούς αισθητήρες οι οποίοι δεν έχουν διαθέσιμη κάποια ενσωματωμένη μνήμη, να εκμεταλλευτούν όλα τα πλεονεκτήματα των TEDS και γενικά του προτύπου.

3.4.1 Πλεονεκτήματα

Τα TEDS είναι από τα πιο σημαντικά κομμάτια του προτύπου, καθώς λόγω αυτών υπάρχουν πολλά πλεονεκτήματα και δυνατότητες, όπως:

- επιτρέπουν αυτοαναγνώριση και αυτοπεριγραφή των μορφοτροπέων

- βοηθάνε στην αυτορύθμιση ενός συστήματος αισθητήρων
- απλοποιούν την εγκατάσταση, την αναβάθμιση και τη συντήρηση ενός συστήματος αισθητήρων καθώς επιτρέπουν την προσθήκη ή αφαίρεση συσκευών χωρίς ιδιαίτερες διαδικασίες
- παρέχουν αυτοτεκμηρίωση των μορφοτροπέων

3.4.2 Δομή

Τα TEDS αποτελούνται από τρία κύρια μέρη:

- οι πρώτες τέσσερις δυφιοσυλλαβές (bytes) απεικονίζουν το συνολικό μέγεθος των TEDS (εκτός του μεγέθους του ίδιου του μέρους του μεγέθους)
- έπειτα υπάρχουν οι ομάδες των δεδομένων (datablocks) οι οποίες έχουν τη μορφή τύπος-μέγεθος τιμής-τιμή (TLV – Type-Length-Value)
- τέλος υπάρχει το άθροισμα ελέγχου το οποίο υπολογίζεται σύμφωνα με τον παρακάτω τύπο (στην ουσία είναι το συμπλήρωμα ως προς ένα του αθροίσματος όλων των bytes του TEDS εκτός προφανώς των δύο bytes του ίδιου του αθροίσματος ελέγχου):

$$\text{checksum} = 0xFFFF - \sum_{i=1}^{\text{TotalOctets}-2} \text{TEDSOctet}(i)$$

Field	Description	Type	# octets
—	TEDS length	UInt32	4
1 to N	Data block	Variable	Variable
—	Checksum	UInt16	2

Σχήμα 15: Βασική δομή TEDS

3.4.3 Κατηγορίες

Τα TEDS χωρίζονται σε δύο μεγάλες κατηγορίες, σε αυτά που είναι υποχρεωτικά να υπάρχουν βάσει του προτύπου και σε αυτά που προαιρετικά μπορούν να περιλαμβάνονται.

Τα υποχρεωτικά είναι τα:

- **MetaTEDS:** αντιστοιχεί ένα σε κάθε TIM. Περιέχει περιγραφή του TIM, τα γενικά χαρακτηριστικά (παράμετροι χρονισμού), τη δομή του και τα κανάλια που περιέχει

Πίνακας 1: Δομή MetaTEDS

Field type	Field name	Description	Data type	# octets
—	—	Length	UInt32	4
0–2	—	Reserved	—	—
3	TEDSID	TEDS Identification Header	UInt8	4
4	UUID	Globally Unique Identifier	UUID	10
5–9	—	Reserved	—	—
Timing-related information				
10	OholdOff	Operational time-out	Float32	4
11	SHoldOff	Slow-access time-out	Float32	4
12	TestTime	Self-Test Time	Float32	4
Number of implemented TransducerChannels				
13	MaxChan	Number of implemented TransducerChannels	UInt16	2
14	CGroup	ControlGroup information sub-block	—	—
Types 20, and 21 define one ControlGroup.				
20	GrpType	ControlGroup type	UInt8	1
21	MemList	ControlGroup member list	array of UInt16	NTc
15	VGroup	VectorGroup information sub-block	—	—
Types 20 and 21 define one VectorGroup.				
20	GrpType	VectorGroup type	UInt8	1
21	MemList	VectorGroup member list	array of UInt16	NTv
16	GeoLoc	Specialized VectorGroup for geographic location	—	—
Types 24, 20, and 21 define one set of geographic location information.				
24	LocEnum	An enumeration defining how location information is provided	UInt8	1
20	GrpType	VectorGroup type	UInt8	1
21	MemList	VectorGroup member list	array of UInt16	NTv
17	Proxies	TransducerChannel proxy definition sub-block	—	—
Types 22, 23, and 21 define one TransducerChannel proxy.				
22	ChanNum	TransducerChannel number of the TransducerChannel proxy	UInt16	1
23	Organiz	TransducerChannel proxy data-set organization	UInt8	1
21	MemList	TransducerChannel proxy member list	array of UInt16	NTp
18–19	—	Reserved	—	—
25–127	—	Reserved	—	—
128–255	—	Open to manufacturers	—	—
—	—	Checksum	UInt16	2

- **Transducer Channel TEDS:** αντιστοιχεί ένα σε κάθε κανάλι. Περιέχει πληροφορίες σχετικές με τον τύπο του μορφοτροπέα (εκκινητήρας ή αισθητήρας) και σχετικές με τα χαρακτηριστικά του καναλιού όπως εύρος τιμών, χρόνοι απόκρισης/προθέρμανσης και δεδομένα βαθμονόμησης

Πίνακας 2: Δομή Transducer Channel TEDS

Field	Field name	Description	Type	# octets
—	—	TEDS length	UInt32	4
0–2	—	Reserved	—	—
3	TEDSID	TEDS Identification	UInt8	4
4–9	—	Reserved	—	—
TransducerChannel related information				
10	CalKey	Calibration key	UInt8	1
11	ChanType	TransducerChannel type key	UInt8	1
12	PhyUnits	Physical Units	UNITS	11
50	UnitType	Physical Units interpretation enumeration	UInt8	1
51	Radians	The exponent for Radians	UInt8	1
52	SterRad	The exponent for Steradians	UInt8	1
53	Meters	The exponent for Meters	UInt8	1
54	Kilogram	The exponent for Kilograms	UInt8	1
55	Seconds	The exponent for Seconds	UInt8	1
56	Amperes	The exponent for Amperes	UInt8	1
57	Kelvins	The exponent for Kelvins	UInt8	1
58	Moles	The exponent for Moles	UInt8	1
59	Candelas	The exponent for Candelas	UInt8	1
60	UnitsExt	TEDS access code for units extension	UInt8	1
13	LowLimit	Design operational lower range limit	Float32	4
14	HiLimit	Design operational upper range limit	Float32	4
15	OError	Worst-case uncertainty	Float32	4
16	SelfTest	Self-test key	UInt8	1
17	MRange	Multi-range capability	UInt8	1
—	—	Data converter-related information	—	—
18	Sample		—	—
40	DatModel	Data model	UInt8	1
41	ModLenth	Data model length	UInt8	1
42	SigBits	Model significant bits	UInt16	2
19	DataSet			
43	Repeats	Maximum data repetitions	UInt16	2
44	SOrigin	Series origin	Float32	4
45	StepSize	Series increment	Float32	4
46	SUnits	Series units	UNITS	11
47	PreTrigg	Maximum pre-trigger samples	UInt16	2
Timing-related information				
20	UpdateT	TransducerChannel update time (t_u)	Float32	4
21	WSetupT	TransducerChannel write setup time (t_{ws})	Float32	4
22	RSetupT	TransducerChannel read setup time (t_{rs})	Float32	4
23	SPeriod	TransducerChannel sampling period (t_{sp})	Float32	4
24	WarmUpT	TransducerChannel warm-up time	Float32	4
25	RDelayT	TransducerChannel read delay time (t_{ch})	Float32	4
26	TestTime	TransducerChannel self-test time requirement	Float32	4
Time of the sample information				
27	TimeSrc	Source for the time of sample	UInt8	1
28	InPropDI	Incoming propagation delay through the data transport logic	Float32	4
29	OutPropD	Outgoing propagation delay through the data transport logic	Float32	4
30	TSError	Trigger-to-sample delay uncertainty	Float32	4
Attributes				
31	Sampling	Sampling attribute	—	—
48	SampMode	Sampling mode capability	UInt8	1
49	SDefault	Default sampling mode	UInt8	1
32	DataXmit	Data transmission attribute	UInt8	1
33	Buffered	Buffered attribute	UInt8	1
34	EndOfSet	End-of-data-set operation attribute	UInt8	1
35	EdgeRpt	Edge-to-report attribute	UInt8	1
36	ActHalt	Actuator-halt attribute	UInt8	1
Sensitivity				
37	Directon	Sensitivity direction	Float32	4
38	DAngles	Direction angles	Two Float32	8
Options				
39	ESOption	Event sensor options	UInt8	1
61–127	—	Reserved	—	—
128–255	—	Open to manufacturers	—	—
—	—	Checksum	UInt16	2

- **User's Transducer Name TEDS:** περιέχουν ένα όνομα που δίνει ο χρήστης για το συγκεκριμένο μορφοτροπέα, με το οποίο μπορεί να αναγνωρίζεται από το σύστημα
- **Physical TEDS:** χρησιμοποιούνται για την περιγραφή λεπτομερειών του φυσικού μέσου επικοινωνίας το οποίο διασυνδέει το TIM με το NCAP οι οποίες δεν παρέχονται από το πρότυπο

Τα προαιρετικά είναι τα:

- **Calibration TEDS:** αντιστοιχεί ένα σε κάθε κανάλι. Περιέχει όλα τα δεδομένα που είναι απαραίτητα για τη βαθμονόμηση των μετρήσεων αλλά και πληροφορίες για το κανάλι που την χρειάζεται
- **Frequency Response TEDS:** παρέχουν πληροφορίες σχετικές με τη απόκριση συχνότητας του μορφοτροπέα
- **Transfer Function TEDS:** περιγράφουν έναν τρόπο σύνδεσης μίας σειράς από επιμέρους συναρτήσεις μεταφοράς και την αποτύπωση της απόκρισης συχνότητας του μορφοτροπέα σε αλγοριθμική μορφή
- **Geographic Location TEDS:** περιέχουν γεωγραφικές πληροφορίες θέσης του TIM τις οποίες μπορούν να προσθέσουν οι χρήστες ώστε να αναγνωρίζουν την ακριβή τοποθεσία του
- **Units Extension TEDS:** περιέχουν κείμενο και χρησιμοποιούνται σε περιπτώσεις όπου η μετρήσιμη ποσότητα δεν μπορεί να εκφραστεί εξ' ολοκλήρου σε μονάδες SI
- **Manufacturers Defined TEDS:** μπορούν να οριστούν από τους κατασκευαστές και να έχουν όποια μορφή και περιεχόμενα αυτοί επιθυμούν
- **Extension TEDS:** μπορούν να χρησιμοποιηθούν για μελλοντικές επεκτάσεις είτε από το ίδιο το πρότυπο ή από κατασκευαστές
- **End-user application-specific TEDS:** μπορούν να περιέχουν πληροφορίες που χρειάζονται κάποιες εφαρμογές, όπως την τοποθεσία που βρίσκεται το TIM ή πληροφορίες για επικοινωνία με κάποιον τεχνικό υπεύθυνο
- **Meta/Channel/Calibration ID TEDS:** περιέχουν κείμενο και παρέχουν τις ίδιες πληροφορίες με τα αντίστοιχά τους αλλά σε μορφή κατανοητή από τον άνθρωπο (και όχι από μηχανή)

- **Command TEDS:** περιέχουν κείμενο και παρέχουν έναν τρόπο στους κατασκευαστές να ορίζουν πρόσθετες εντολές εκτός αυτών που περιλαμβάνονται στο πρότυπο

3.5 Είδη μορφοτροπέων

Το πρότυπο περιγράφει έξι είδη μορφοτροπέων [42] από τα οποία τα τέσσερα είναι είδη αισθητήρων, ένας εκκινητήρας και ένας ανιχνευτής συμβάντος:

- **αισθητήρας (sensor):** μετράει κάποια φυσική παράμετρο, έπειτα από αίτηση, και επιστρέφει ψηφιακά δεδομένα που αντιπροσωπεύουν την παράμετρο αυτή
- **αισθητήρας με ενδιάμεση μνήμη (buffered sensor):** είναι ότι και ο απλός αισθητήρας αλλά μπορεί και αποθηκεύει την τιμή της τελευταίας μέτρησης στην ενδιάμεση μνήμη που διαθέτει
- **αισθητήρας ακολουθίας δεδομένων (data sequence sensor):** δειγματοληπτεί μία ροή από δεδομένα για όσο χρόνο του οριστεί
- **αισθητήρας ακολουθίας δεδομένων με ενδιάμεση μνήμη (buffered data sequence sensor):** είναι ότι και ο απλός αισθητήρας ακολουθίας δεδομένων αλλά με ενδιάμεση μνήμη
- **εκκινητήρας (actuator):** εκτελεί μία φυσική ή εικονική ενέργεια η οποία σχετίζεται με τα δεδομένα που του αποστέλλονται
- **αισθητήρας ανίχνευσης συμβάντος (event sensor):** παράγει ένα σήμα κάθε φορά που αντιληφθεί κάποιο συγκεκριμένο συμβάν

3.6 Αρχιτεκτονική

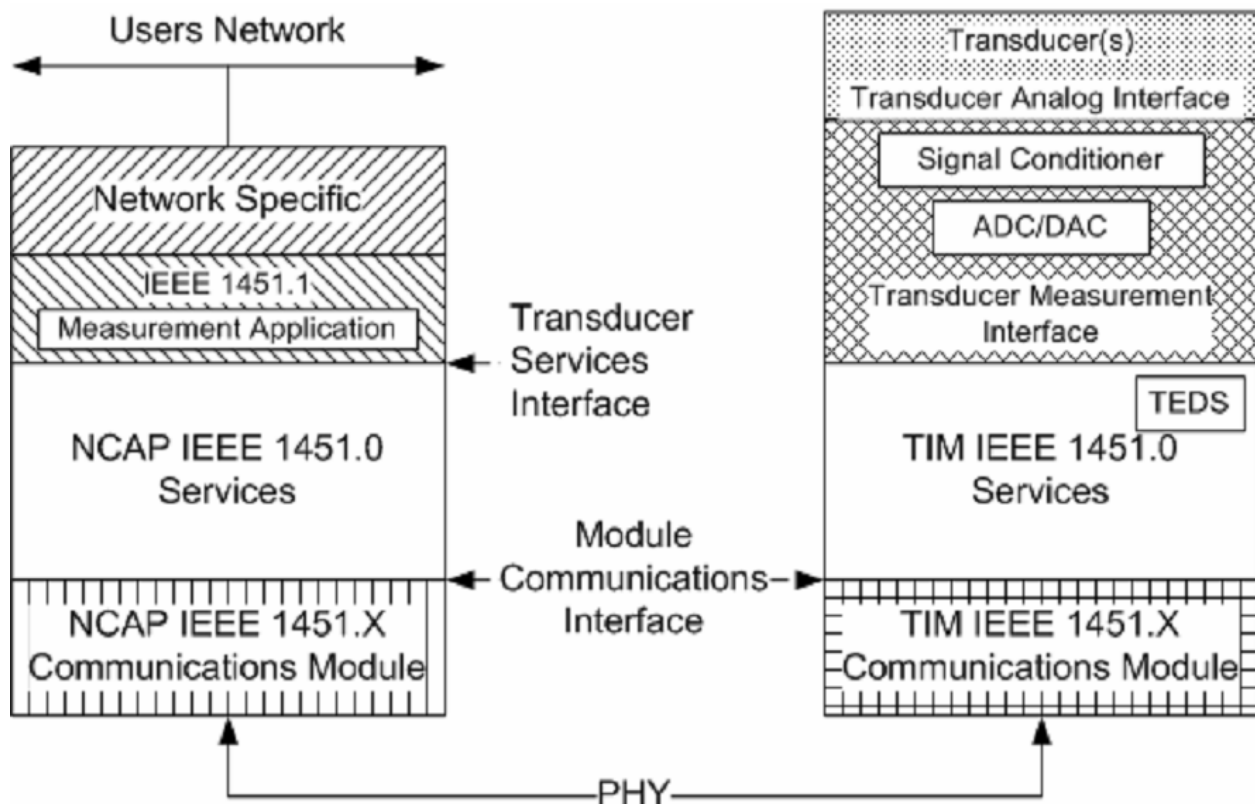
Όπως έχουμε ήδη αναφέρει, το πρότυπο [24] περιγράφει δύο κύριες οντότητες, τα TIM και τα NCAP. Το TIM ασχολείται κυρίως με την λήψη μετρήσεων (με τη βοήθεια των αισθητήρων που περιέχει) και τη μετατροπή τους σε καθορισμένη ψηφιακή μορφή. Η μορφή των ψηφιακών δεδομένων προσδιορίζεται μέσα στα TEDS, οπότε το NCAP με συνδυασμό τους μπορεί να κατανοήσει την ακριβή μορφή τους. Επίσης κάθε TIM διαθέτει κάποιους εσωτερικούς καταχωρητές οι οποίοι αποθηκεύουν πληροφορίες σχετικές με την κατάστασή του και πιθανά σφάλματα που συμβαίνουν. Το NCAP

διαβάζοντάς τους μπορεί να αντιληφθεί την εσωτερική κατάσταση του TIM αλλά και αν κάποια ζητούμενη ενέργεια δεν κατάφερε να εκτελεστεί σωστά.

Γενικά το TIM είτε απαντάει σε ερωτήσεις του NCAP ή το ειδοποιεί σε καταστάσεις ανάγκης. Οι κυριότερες ενέργειες που εκτελεί το TIM είναι οι απαντήσεις σε ερωτήματα ανακάλυψης από το NCAP και σε αιτήματα πρόσβασης σε μορφοτροπίες, η απάντηση ή και εκκίνηση εργασιών διαχείρισης και οι ενέργειες που έχουν σχέση με τη διαχείριση των TEDS.

Το NCAP είναι στην ουσία μία δικτυακή πύλη μεταξύ του δικτύου που ενώνει τα υπόλοιπα στοιχεία του προτύπου (το οποίο καθορίζεται πλήρως από το πρότυπο) και των εξωτερικών δικτύων διασύνδεσης τα οποία μπορεί να είναι είτε ανοιχτά ή κάποιου συγκεκριμένου κατασκευαστή.

Οι κυριότερες ενέργειες που εκτελεί το NCAP είναι η έναρξη ενεργειών ανακάλυψης των διαθέσιμων TIM, η αιτήσεις προς τα TIM για πρόσβαση στους μορφοτροπίες τους, η διαχείριση τους και τέλος η λήψη των TEDS και πιθανόν η προσωρινή αποθήκευσή τους εσωτερικά τους.



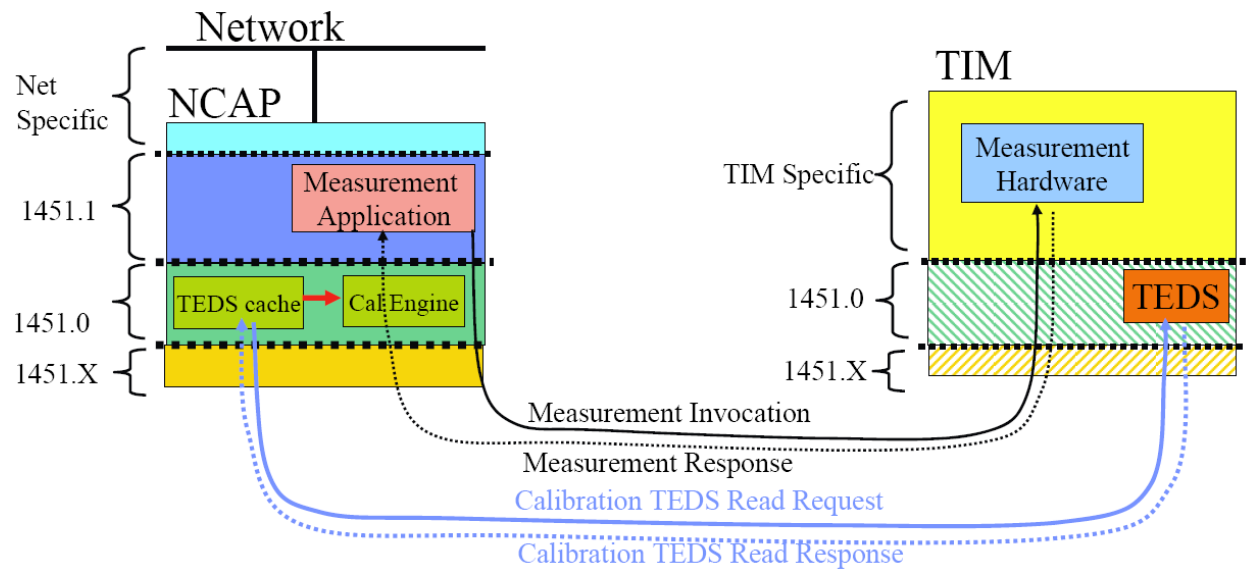
Σχήμα 16: Δομή IEEE 1451 ευφυούς αισθητήρα

Όπως παρατηρούμε στο Σχήμα 16, ανάμεσα στο επίπεδο του IEEE 1451.0 και οποιουδήποτε IEEE 1451.X επιπέδου, βρίσκεται η Διεπαφή Επικοινωνίας Αρθρωμάτων (Module Communications Interface). Αυτή η διεπαφή είναι κοινή μεταξύ TIM και NCAP. Οι κυριότεροι στόχοι της είναι οι παρακάτω:

- Να διευκολύνει την προσθήκη νέου φυσικού επιπέδου επικοινωνίας μεταξύ του TIM και του NCAP
- Να παρέχει ένα επίπεδο αφαίρεσης στην επικοινωνία TIM και NCAP
- Να ορίσει μία Διεπαφή Προγραμματισμού Εφαρμογής (Application Programming Interface – API) ανεξάρτητο από γλώσσα προγραμματισμού και συσκευές
- Να ορίσει μία προκαθορισμένη μορφή επικοινωνίας, την οποία εφόσον ακολουθήσουν οι κατασκευαστές θα εγγυάται τη απρόσκοπτη λειτουργία και επικοινωνία μεταξύ TIM και NCAP

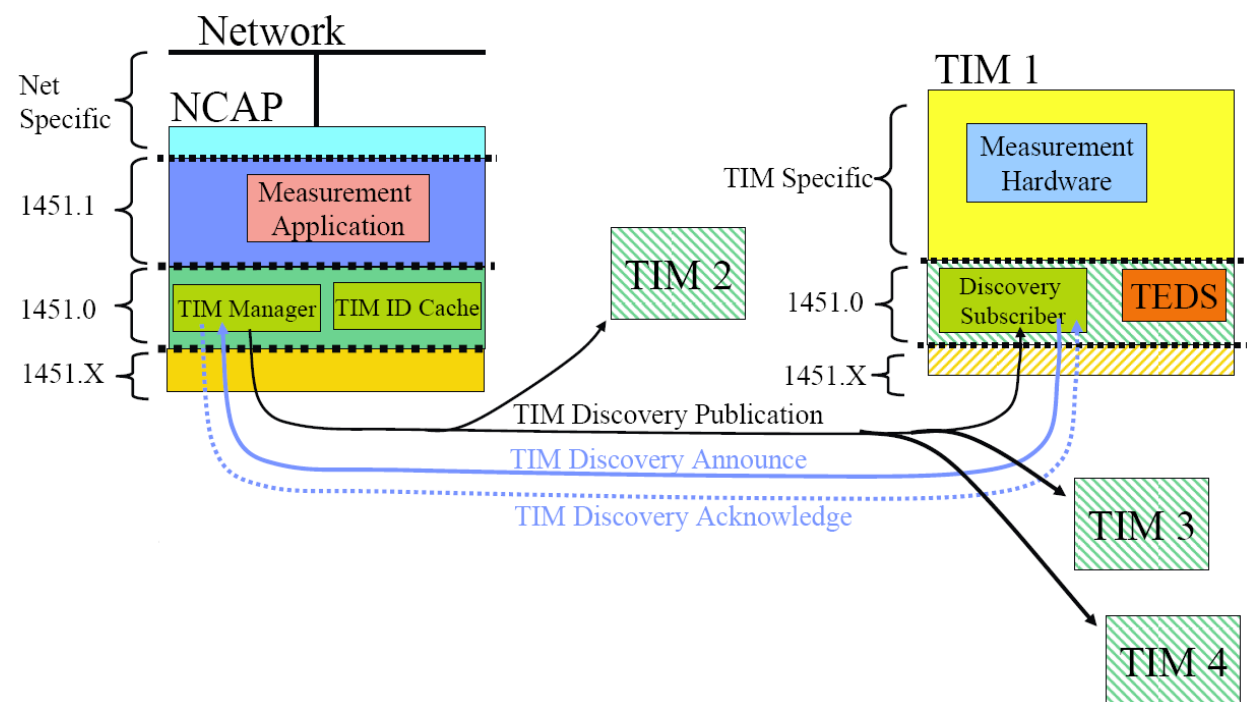
Οι τρόποι επικοινωνίας [11] που υποστηρίζονται από τη συγκεκριμένη διεπαφή είναι:

- **Πελάτη-Εξυπηρετητή:** κατά την οποία υποστηρίζεται μονόδρομη αλλά και αμφίδρομη ροή πληροφοριών με σύγχρονο ή ασύγχρονο τρόπο. Στο Σχήμα 17 παρατηρούμε τον τρόπο με τον οποίο μπορεί να πραγματοποιηθεί μία μέτρηση από το NCAP. Αρχικά το NCAP στέλνει στο TIM μία αίτηση διαβάσματος των TEDS βαθμονόμησης. Εφόσον το TIM τα στείλει, αποθηκεύονται στην προσωρινή μνήμη η οποία βρίσκεται μέσα στο NCAP. Έπειτα το NCAP στέλνει στο TIM μία αίτηση μέτρησης. Το TIM επιστρέφει σε ψηφιακή μορφή την τιμή της μέτρησης στο NCAP, το οποίο με τη χρήση των TEDS βαθμονόμησης, την επεξεργάζεται και στέλνει μέσω δικτύου το αποτέλεσμα.



Σχήμα 17: NCAP-TIM (Πελάτης-Εξυπηρετητής)

- Δημοσιοποίηση-Εγγραφή:** κατά την οποία μπορεί να υπάρξει επικοινωνία είτε ένα προς ένα ή ένα προς πολλά. Υπάρχει μονής κατεύθυνσης ροή πληροφορίας οπότε δεν υπάρχει γνωστοποίηση ως προς την επιτυχή παραλαβή κάποιας πληροφορίας. Όπως παρατηρούμε στο Σχήμα 18, αρχικά το NCAP προσπαθεί να ανακαλύψει όλα τα TIM τα οποία είναι συνδεδεμένα πάνω του. Κάθε διαθέσιμο TIM απαντάει στο αίτημα ανακάλυψης του NCAP και έπειτα το NCAP στέλνει μία επιβεβαίωση σε κάθε ένα από αυτά.



Σχήμα 18: NCAP-TIM (Δημοσιοποίηση-Εγγραφή)

- **Πλήρως συμμετρική επικοινωνία:** κατά την οποία κάθε NCAP και κάθε TIM μπορεί να είναι πελάτης, εξυπηρετητής, εκδότης και συνδρομητής ταυτόχρονα.

3.7 Τρόποι πρόσβασης

Ένα από τα σημαντικότερα κομμάτια τα οποία ενδιαφέρουν το χρήστη (αν όχι το μοναδικό) είναι οι τρόποι πρόσβασης στην υποδομή [43], οι οποίοι είναι οι παρακάτω:

- **IEEE 1451.1 πρότυπο:** περιγράφει μία ιεραρχία κλάσεων με τη χρήση των οποίων μπορούμε να προσπελάσουμε διαφόρων τύπων δεδομένα του συστήματος [44]
- **HTTP διεπαφή:** περιγράφεται στο IEEE 1451.0 [27] και αποτελείται από τέσσερις βασικές κατηγορίες διεπαφών:
 - **ανακάλυψης (discovery):** παρέχει πληροφορίες σχετικές με τα διαθέσιμα TIM και τα κανάλια μορφοτροπέων που περιέχουν
 - **πρόσβασης μορφοτροπέα (transducer access):** παρέχει πρόσβαση στους εκκινητήρες και τους αισθητήρες
 - **διαχειριστή μορφοτροπέα (transducer manager):** παρέχει μεγαλύτερο έλεγχο πάνω στο TIM, όπως η αποστολή μη προκαθορισμένων εντολών
 - **διαχειριστή TEDS:** παρέχει πρόσβαση στα TEDS, υποστηρίζει διάβασμα, γράψιμο αλλά και διαχείρισης της κρυφής μνήμης

Επίσης ορίζεται άλλη μία διεπαφή η οποία ονομάζεται αναστροφή εφαρμογής (App Callback) και χρησιμοποιείται από εφαρμογές που χρειάζονται προηγμένα χαρακτηριστικά ειδοποίησης και ασύγχρονης επικοινωνίας με το σύστημα.

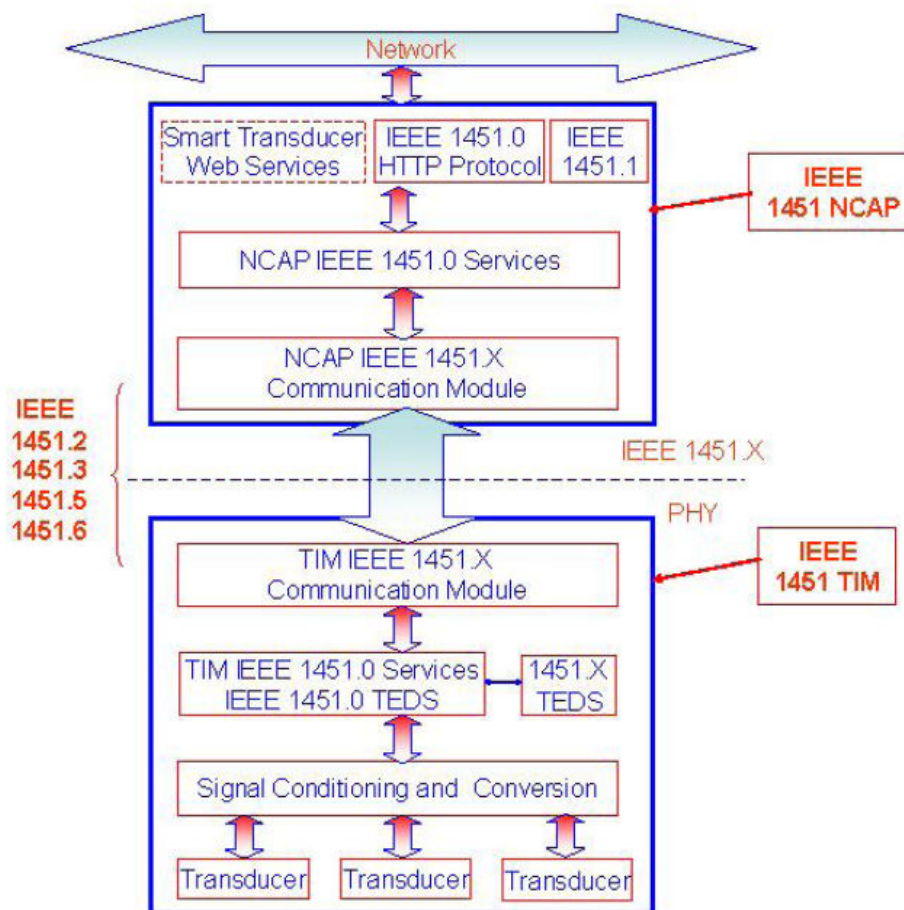
Στο πρότυπο περιγράφονται πλήρως οι συγκεκριμένες διεπαφές και επίσης παρέχονται οι δομές των αποκρίσεων οι οποίες είναι κωδικοποιημένες σε XML γλώσσα.

Πίνακας 3: IEEE 1451 HTTP API

API type	Name	Path
Discovery API	TIMDiscovery	1451/Discovery/TIMDiscovery
	TransducerDiscovery	1451/Discovery/TransducerDiscovery
Transducer Access API	ReadData	1451/TransducerAccess/ReadData
	StartReadData	1451/TransducerAccess/StartReadData
	MeasurementUpdate	1451/TransducerAccess/MeasurementUpdate
	WriteData	1451/TransducerAccess/WriteData
	StartWriteData	1451/TransducerAccess/StartWriteData
TEDS Manager API	ReadTeds	1451/TEDSManager/ReadTeds
	ReadRawTeds	1451/TEDSManager/ReadRawTeds
	WriteTeds	1451/TEDSManager/WriteTeds
	WriteRawTeds	1451/TEDSManager/WriteRawTeds
	UpdateTedsCache	1451/TEDSManager/UpdateTedsCache
Transducer Manager API	SendCommand	1451/TransducerManager/SendCommand
	StartCommand	1451/TransducerManager/StartCommand
	CommandComplete	1451/TransducerManager/CommandComplete
	Trigger	1451/TransducerManager/Trigger
	StartTrigger	1451/TransducerManager/StartTrigger

- **υπηρεσίες ιστού:** σε αντίθεση με τους δύο προηγούμενους τρόπους, ο συγκεκριμένος δεν αναφέρεται πουθενά στο πρότυπο. Η κυριότερη και επικρατέστερη υλοποίηση ονομάζεται Υπηρεσίες Ιστού Ευφυών Μορφοτροπέων (Smart Transducer Web Services – STWS) [45], [46], [47] και έχει αναπτυχθεί από το NIST. Βασίζεται πάνω σε υπηρεσιοστρεφή αρχιτεκτονική σύμφωνα με την οποία με τη χρήση της Γλώσσας Περιγραφής Υπηρεσιών Ιστού (Web Services Description Language – WSDL) περιγράφονται τα μηνύματα, οι παράμετροι, η λειτουργία και τα αποτελέσματα κάθε υπηρεσίας ιστού. Με τη χρήση αυτών των υπηρεσιών μπορούμε να έχουμε προκαθορισμένο τρόπο πρόσβασης, ανεξάρτητο από το λειτουργικό σύστημα ή την γλώσσα προγραμματισμού που χρησιμοποιούμε.

Ένας ενδιαφέρον συνδυασμός είναι αυτός του IEEE 1451 με το OGC-SWE [48]. Καθώς το πρώτο περιγράφει κυρίως τον τρόπο λήψης δεδομένων από τους αισθητήρες και την αποστολής τους στο επίπεδο δικτύου ενώ το δεύτερο παίρνει τα δεδομένα που προέρχονται από τους αισθητήρες και τα μεταφέρει στις εφαρμογές μέσω του παγκόσμιου ιστού.



Σχήμα 19: Τρόποι πρόσβασης στο IEEE 1451

3.8 Ενδιαφέροντα στοιχεία

3.8.1 Μονάδες μέτρησης

Οι μετρήσεις από τους αισθητήρες εκφράζονται σε μονάδες μέτρησης που ανήκουν στο SI. Χρησιμοποιώντας τις βασικές μονάδες, μπορούμε να εκφράσουμε όποια άλλη χρειάζεται, καθώς θα ισούται με κάποιον τύπο ο οποίος θα αποτελείται αποκλειστικά από βασικές μονάδες. Αυτό που χρειάζεται είναι να γνωρίζουμε τον τύπο αυτό. Σύμφωνα με το πρότυπο, θα πρέπει να ορίσουμε τους εκθέτες κάθε βασικής μονάδας εφόσον τους διπλασιάσουμε και τους προσθέσουμε 128, σύμφωνα με τον Πίνακα 4.

Πίνακας 4: Αναπαράσταση μονάδων μέτρησης

Field	Description	Data type	Number of octets
1	Physical Units interpretation—see Table 3	UInt8	1
2	(2 * <exponent of radians>) + 128	UInt8	1
3	(2 * <exponent of steradians>) + 128	UInt8	1
4	(2 * <exponent of meters>) + 128	UInt8	1
5	(2 * <exponent of kilograms>) + 128	UInt8	1
6	(2 * <exponent of seconds>) + 128	UInt8	1
7	(2 * <exponent of amperes>) + 128	UInt8	1
8	(2 * <exponent of kelvins>) + 128	UInt8	1
9	(2 * <exponent of moles>) + 128	UInt8	1
10	(2 * <exponent of candelas>) + 128	UInt8	1

Για παράδειγμα για να απεικονίσουμε τάση θα πρέπει να χρησιμοποιήσουμε $\text{Volts} = \text{m}^2\text{kg}/(\text{sec}^3\text{A})$, οπότε θα έχουμε:

Πίνακας 5: Παράδειγμα απεικόνισης μονάδων (Volts)

	rad	sr	m	kg	sec	A	K	mol	cd
Εκθέτης	0	0	2	1	-3	-1	0	0	0
Πεδίο	128	128	132	130	122	126	128	128	128

3.8.2 Εντολές/Απαντήσεις

Το πρότυπο περιγράφει δύο τύπους μηνυμάτων τα οποία ανταλλάσσονται μεταξύ του NCAP και του TIM, τις εντολές και τις απαντήσεις. Οι εντολές περιλαμβάνουν το κανάλι μορφοτροπεία στο οποίο απευθύνεται (αν είναι ίσο με μηδέν σημαίνει ότι απευθύνεται σε όλο το TIM), την κλάση, την λειτουργία της και το μήκος των επιπλέον bytes που είναι πιθανόν να περιέχει. Η κλάση αντιπροσωπεύει τον προορισμό της εντολής (κανάλι μορφοτροπεία ή TIM) αλλά και την κατάστασή του (λειτουργία, αναμονή).

Η απάντηση περιέχει ένα πεδίο το οποίο δείχνει αν η εντολή της οποίας είναι απάντηση εκτελέστηκε επιτυχώς ή όχι και το μήκος των επιπλέον bytes που είναι πιθανόν να περιέχει.

Πίνακας 6: Δομή εντολής IEEE 1451

Byte Number	Description
1	Destination Transducer Channel Number (Most significant byte)
2	Destination Transducer Channel Number (Least significant byte)
3	Command Class
4	Command Function
5	Length (Most significant byte)
6	Length (Least significant byte)
7-N	Command dependent bytes

Πίνακας 7: Δομή απάντησης IEEE 1451

Byte Number	Description
1	Success/Fail Flag
2	Length (Most significant byte)
3	Length (Least significant byte)
4-N	Reply dependent bytes

3.9 Πλεονεκτήματα

Τα κυριότερα πλεονεκτήματα του IEEE 1451 είναι τα παρακάτω:

- είναι ανοιχτό πρότυπο το οποίο κάθε κατασκευαστής μπορεί να ακολουθήσει, κάτι το οποίο μειώνει το κόστος
- βοηθάει την ανάπτυξη διεπαφών ανεξάρτητων από το δίκτυο και τον κατασκευαστή
- περιγράφει και υποστηρίζει ένα γενικό και ολοκληρωμένο μοντέλο για δεδομένα, έλεγχο, χρονισμό, βαθμονόμηση, διαμόρφωση και ρύθμιση μορφοτροπών
- υποστηρίζει ένα μεγάλο εύρος αισθητήρων και εκκινητήρων, οι οποίοι μπορούν να προσπελαστούν με κοινό τρόπο
- καθορίζει διεπαφές μεταξύ αισθητήρων/εκκινητήρων και οργάνων/ μικροεπεξεργαστών/δικτύων σε φυσικό αλλά και σε λειτουργικό επίπεδο
- υποστηρίζει άμεση βυσμάτωση και λειτουργία των επιμέρους οντοτήτων του συστήματος
- καθορίζει αναλογικές/ψηφιακές και ενσύρματες/ασύρματες διεπαφές για τη διασύνδεση μορφοτροπών

- περιγράφει αυτοπεριγραφόμενους αισθητήρες με τη βοήθεια των TEDS
- επιτρέπει στους αισθητήρες να εγκατασταθούν/αναβαθμιστούν/αντικατασταθούν/μεταφερθούν με την ελάχιστη δυνατή προσπάθεια και αλλαγές στο υπάρχον σύστημα
- εξαλείφει τη χειροκίνητη ρύθμιση και εισαγωγή δεδομένων και παραμέτρων στο σύστημα, κάτι το οποίο είναι χρονοβόρο, πολλές φορές δύστροπο και επιρρεπές σε λάθη
- είναι ευέλικτο και εύκολα επεκτάσιμο με νέες λειτουργίες και δυνατότητες
- οι τιμές των μετρήσεων από τους αισθητήρες επιστρέφονται σε μονάδες στο σύστημα μέτρησης SI

3.10 Μειονεκτήματα

Τα κυριότερα μειονεκτήματα του IEEE 1451 είναι τα παρακάτω:

- είναι αρκετά πολύπλοκο, αλλά ένα τόσο ευρύ και περιεκτικό πρότυπο δεν θα μπορούσε να μην είναι. Η πολυπλοκότητα δεν έγκειται μόνο στο ίδιο το πρότυπο αλλά και στις επιμέρους οντότητες που περιγράφει
- για κάθε μορφοτροπία θα πρέπει να παράγονται και τα ανάλογα TEDS, κάτι το οποίο δημιουργεί την ανάγκη για ανάπτυξη κωδικοποιητών για την εύκολη δημιουργία τους
- υπάρχουν ελάχιστα συστήματα συμβατά με το πρότυπο τα οποία να λειτουργούν αυτή τη στιγμή, οπότε δεν είναι ακόμα σίγουρο ότι ο γενικός σχεδιασμός του είναι σωστός και χωρίς προβλήματα
- δεν έχει υιοθετηθεί ακόμα από μεγάλο αριθμό κατασκευαστών
- η διεπαφή μεταξύ TIM και NCAP που περιγράφεται στο IEEE 1451.2 (TII) χρησιμοποιείται συγκεκριμένα στο συγκεκριμένο πρότυπο ενώ θα μπορούσε να χρησιμοποιηθεί κάποια ήδη υπάρχουσα (π.χ. RS232) η οποία δεν υπολείπεται σε κάτι [34]

3.11 Εφαρμογές

Παρόλο που η ανάπτυξη των οντοτήτων του πρωτοκόλλου σε υλικό είναι μία δύσκολη υπόθεση, έχουν γίνει κάποιες προσπάθειες για ανάπτυξη TIM [32] [36] και NCAP [33] αλλά και ολοκληρωμένες λύσεις [35] με τη χρήση υπαρχόντων δικτύων (όπως είναι το CAN). Επίσης έχουν γίνει προσπάθειες για την ανάπτυξη κωδικοποιητών για TEDS [37] [38] των οποίων η δημιουργία είναι μεγάλος φόρτος, αφού χρειάζονται για κάθε μορφοτροπία συμβατό με το πρότυπο.

Παρόλα τα μειονεκτήματά του, είναι το επικρατέστερο πρότυπο του είδους του και για αυτό το λόγο έχει ήδη χρησιμοποιηθεί ως βάση σε διάφορες εφαρμογές.

Το Κέντρο Διαστήματος Κένεντυ της Εθνικής Υπηρεσίας Αεροναυτικής και Διαστήματος (National Aeronautics and Space Administration – NASA) των Ηνωμένων Πολιτειών Αμερικής έχει αναπτύξει μία χαμηλών απαιτήσεων έκδοση του προτύπου [28] για μετρήσεις πίεσης και πολύ χαμηλών θερμοκρασιών.

Επίσης έχει αναπτυχθεί ένα σύστημα παρακολούθησης της περιβαλλοντικής ρύπανσης του αέρα [31] το οποίο παίρνει μετρήσεις συγκεντρώσεων επικίνδυνων αερίων, όπως τα CO, NO₂, SO₂ και O₃. Το σύστημα έχει δυνατότητα να απεικονίζει αυτές τις μετρήσεις γραφικά αλλά και να ειδοποιεί σε περίπτωση που κάποια από αυτές ξεπεράσει τα όρια ασφαλείας.

Άλλος ένας τομέας εφαρμογής είναι αυτός της υγείας [40] [41] όπου έχουν αναπτυχθεί εφαρμογές για την παρακολούθηση διαφόρων στοιχείων των ασθενών, καθώς και μεταφορά και αποθήκευσή τους. Επίσης κάποια από αυτά επιτρέπουν την έγκαιρη ειδοποίηση του νοσηλευτικού προσωπικού σε περιπτώσεις ανάγκης.

Άλλη μία εφαρμογή του προτύπου είναι στα δίκτυα των αυτοκινήτων [39] όπου υπάρχει η ανάγκη για πολλές μετρήσεις οι οποίες συμβάλλουν στην ασφάλεια των επιβατών.

Για να έχει μεγαλύτερη διάδοση και αποδοχή, είναι ανάγκη να υπάρξουν εργαλεία τα οποία θα βοηθήσουν την ανάπτυξη και τον έλεγχο ορθής λειτουργίας των επιμέρους οντοτήτων που θα επιλέξει ο καθένας να υλοποιήσει. Προς αυτή την κατεύθυνση έχουν αναπτυχθεί εικονικά όργανα [30] τα οποία μας βοηθάνε σε δοκιμές σωστής λειτουργίας αλλά και απόδοσης τόσο των TIM όσο και των NCAP.

4. ΥΛΟΠΟΙΗΣΗ ΠΡΩΤΟΤΥΠΟΥ

Στόχος μας ήταν η υλοποίηση ενός πρωτοτύπου με τη χρήση λογισμικού, το οποίο θα περιελάμβανε τα βασικά κομμάτια του IEEE 1451, ώστε να έχουμε ένα λειτουργικό αποτέλεσμα το οποίο θα ήταν όσο πιο κοντά γίνεται στο πρότυπο. Με αυτό τον τρόπο θα μπορούσαμε να διαπιστώσουμε την ορθότητα του προτύπου αλλά και να προτείνουμε τυχόν βελτιώσεις. Ως κόμβοι αισθητήρων χρησιμοποιήθηκαν τα SunSPOT (Sun's Small Programmable Object Technology) της εταιρίας Sun Microsystems. Η ανάπτυξη έγινε εξ' ολοκλήρου σε Java, καθώς μπορούν να την εκτελούν εγγενώς τα SunSPOTs.

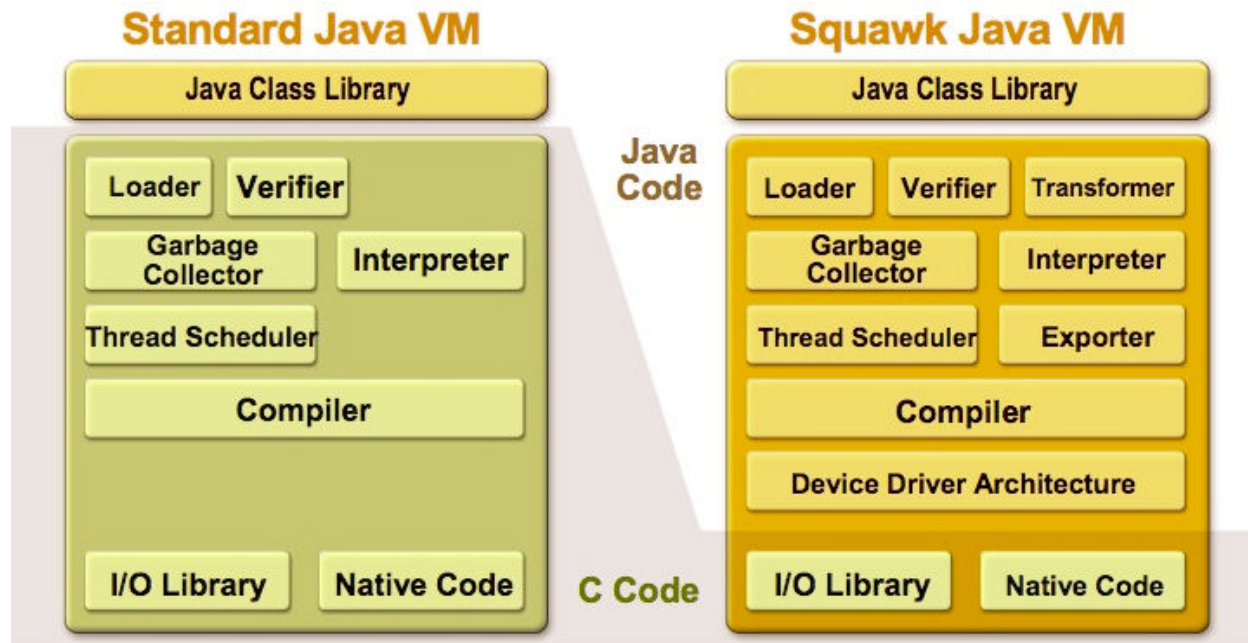
4.1 SunSPOT



Εικόνα 2: SunSPOT

Τα SunSPOT είναι η πρόταση της Sun για ανάπτυξη ασύρματων δικτύων αισθητήρων. Είναι μία μικρή συσκευή η οποία προγραμματίζεται εξ' ολοκλήρου σε Java (J2ME) και μπορεί να χρησιμοποιηθεί σε εφαρμογές αισθητήρων και όχι μόνο. Η συγκεκριμένη συσκευή τρέχει την εικονική μηχανή Squawk [19], η οποία είναι μία μικρού μεγέθους εικονική μηχανή της Java, κατάλληλη για φορητές και ενσωματωμένες συσκευές. Ενώ οι περισσότερες εικονικές μηχανές της Java είναι γραμμένες κυρίως σε C/C++ και assembly, η Squawk είναι γραμμένη κυρίως σε Java, εκτός από κάποια μικρά κομμάτια τα οποία έχουν να κάνουν με τον πρωτογενή κώδικα και την είσοδο/έξοδο δεδομένων. Αυτό την κάνει εύκολα μεταφέρσιμη και ενσωματώσιμη διαφανώς με τους πόρους της εφαρμογής όπως είναι τα αντικείμενα, τα νήματα και οι διεπαφές του λειτουργικού συστήματος.

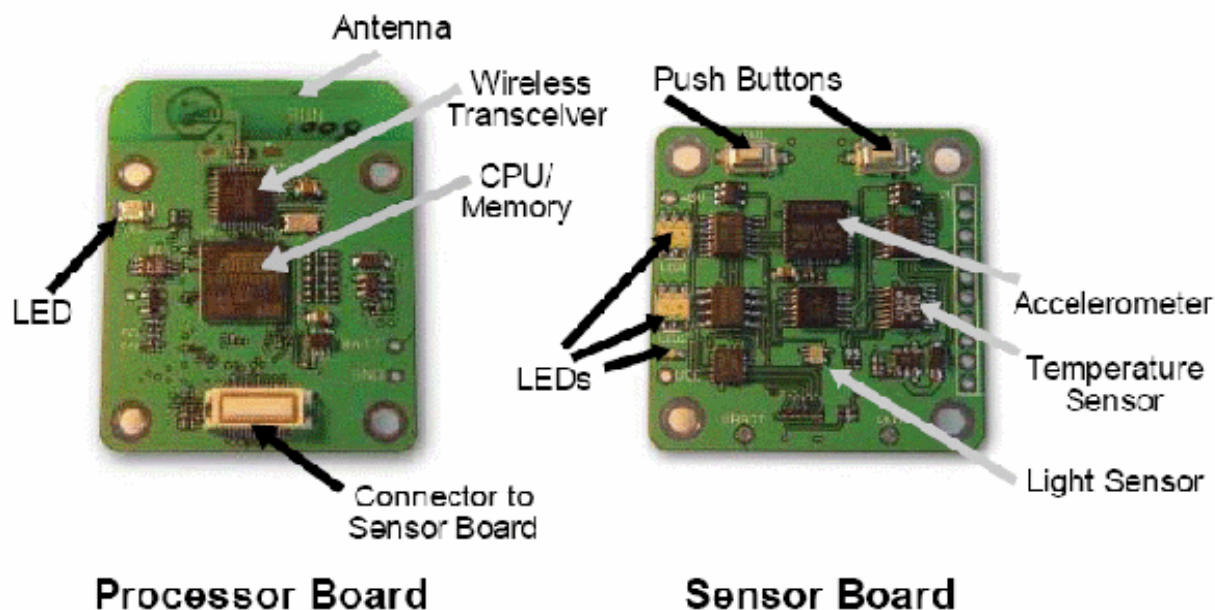
Κάποια από τα βασικά χαρακτηριστικά της συγκεκριμένης συσκευής είναι ότι οι εφαρμογές εκτελούνται απευθείας από την αστραπιαία μνήμη (flash memory), οι οδηγοί των συσκευών είναι και αυτοί γραμμένοι σε Java και η διαχείριση της μπαταρίας είναι αυτόματη προσφέροντας μεγάλους χρόνους λειτουργίας και αναμονής.



Εικόνα 3: Σύγκριση πρότυπης και Squawk εικονικής συσκευής

Το υλισμικό του SunSPOT περιλαμβάνει τα παρακάτω:

- επεξεργαστή ARM (180 MHz) – 512K RAM – 4M Flash
- ασύρματη διεπαφή IEEE 802.15.4 (2.4 GHz)
- διεπαφή USB
- 2G/6G τριών αξόνων επιταχυνσιόμετρο
- αισθητήρα θερμοκρασίας
- αισθητήρα φωτός
- οχτώ τρίχρωμα LEDs
- έξι αναλογικές εισοδοί
- δύο διακόπτες
- πέντε γενικού σκοπού ακίδες
- τέσσερις υψηλού ρεύματος ακίδες

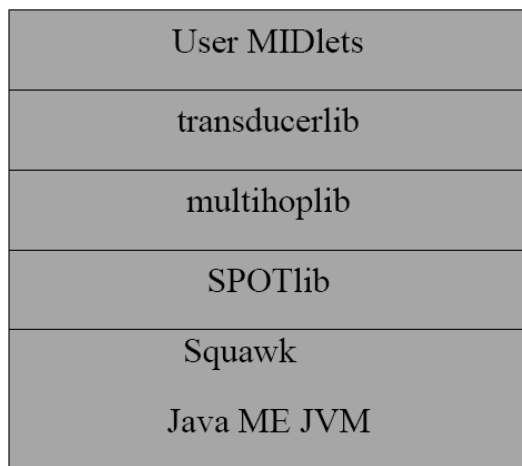


Εικόνα 4: Υλισμικό του SunSPOT

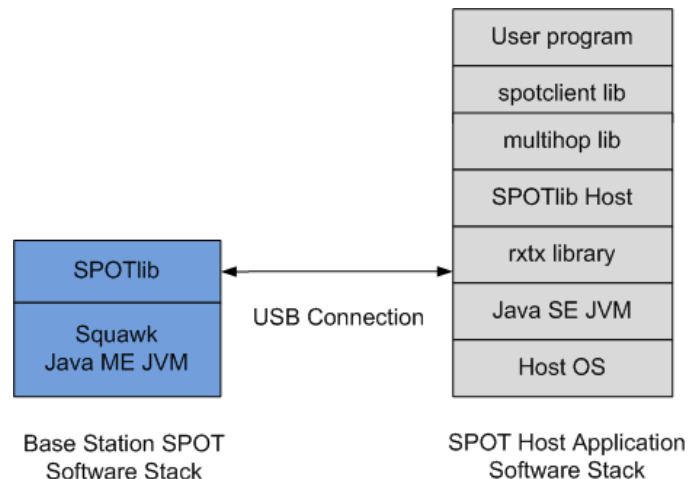
Στο Σχήμα 20 παρατηρούμε την στοίβα του λογισμικού το οποίο τρέχει στο SunSPOT. Όπως παρατηρούμε στο ανώτερο επίπεδο βρίσκονται τα MIDlets του χρήστη, τα οποία είναι μία κατηγορία εφαρμογών γραμμένες σε Java για κινητά τηλέφωνα ή ενσωματωμένες συσκευές. Ακριβώς από κάτω υπάρχει η *transducerlib*, η οποία είναι η βιβλιοθήκη που παρέχει πρόσβαση στο υλικό του SunSPOT, για παράδειγμα στους αισθητήρες, τους διακόπτες, τα LEDs και τις εισόδους/εξόδους. Από κάτω βρίσκεται η *multihorlib*, η οποία είναι η βιβλιοθήκη η οποία παρέχει υψηλού επιπέδου πρωτόκολλα σχετικά με τη δικτυακή δρομολόγηση πακέτων. Έπειτα είναι η βιβλιοθήκη *SPOTlib* η οποία περιλαμβάνει διάφορες συναρτήσεις που παρέχουν χαμηλού επιπέδου πρόσβαση στην ασύρματη διεπαφή και τις βασικές λειτουργίες εισόδου/εξόδου του SunSPOT. Τέλος στο κατώτερο επίπεδο βρίσκεται η εικονική μηχανή Squawk, η οποία μπορεί να εκτελείται απ' ευθείας στα SunSPOT χωρίς την ανάγκη ύπαρξης λειτουργικού συστήματος.

Για επικοινωνήσει μία εφαρμογή ξενιστή (host application) ενσύρματα (μέσω της USB διεπαφής) με ένα SunSPOT, πρέπει να χρησιμοποιήσει τις βιβλιοθήκες που παρατηρούμε στο Σχήμα 21. Ξενιστής είναι μία εφαρμογή η οποία εκτελείται σε έναν υπολογιστή, στον οποίο μπορεί να είναι συνδεδεμένο ένα ή περισσότερα SunSPOT. Στο ανώτερο επίπεδο προφανώς υπάρχει η εφαρμογή του χρήστη η οποία πρέπει να είναι γραμμένη σε Java ώστε να μπορεί να χρησιμοποιήσει τις υπάρχουσες βιβλιοθήκες οι οποίες είναι επίσης υλοποιημένες σε Java. Ακριβώς από κάτω βρίσκεται η

spotclientlib βιβλιοθήκη η οποία παρέχει ένα σύνολο από εντολές οι οποίες μπορούν να αποσταλούν σε κάποιο SunSPOT. Πιο κάτω βρίσκονται οι multihoplib και SPOTlib βιβλιοθήκες οι οποίες παρέχουν τις ίδιες δυνατότητες με αυτές που βρίσκονται στο SunSPOT. Για παράδειγμα με την χρήση της SPOTlib μπορούμε να έχουμε χαμηλού επιπέδου πρόσβαση στην ασύρματη διεπαφή από την εφαρμογή ξενιστή. Η rxtx βιβλιοθήκη είναι υπεύθυνη να παρέχει σειριακή επικοινωνία μέσω της USB διεπαφής. Τέλος υπάρχει η εικονική μηχανή της Java και το λειτουργικό σύστημα πάνω στο οποίο εκτελείται η εφαρμογή.



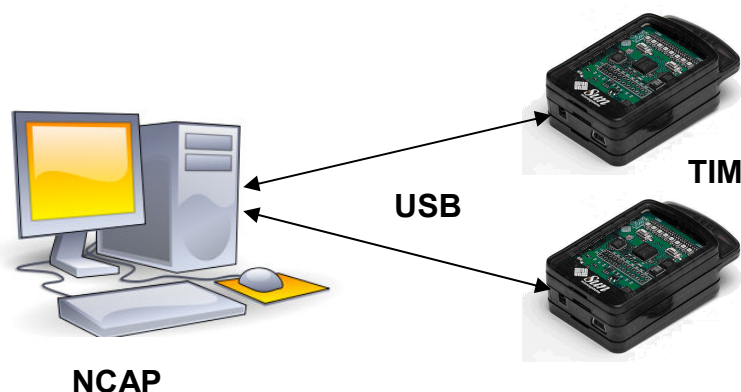
Σχήμα 20: Στοίβα λογισμικού του SunSPOT



Σχήμα 21: Στοίβα λογισμικού εφαρμογής ξενιστή

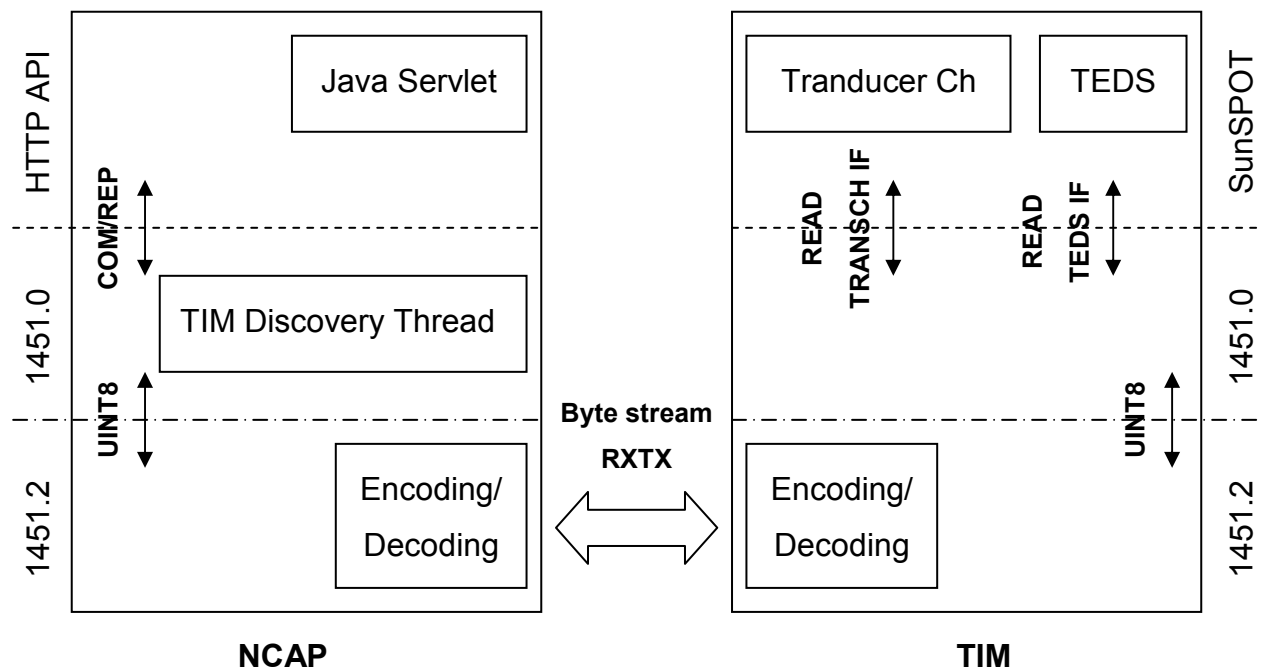
4.2 Αρχιτεκτονική

Στην υλοποίησή μας τα SunSPOT παίζουν το ρόλο των TIM και ένας υπολογιστής χρησιμοποιείται για NCAP. Η επικοινωνία μεταξύ τους γίνεται μέσω της USB θύρας.



Σχήμα 22: Αρχιτεκτονική #1 της υλοποίησής μας

Έχει υλοποιηθεί το μεγαλύτερο μέρος του IEEE 1451.0 καθώς, η HTTP διεπαφή που περιγράφει και ένα μικρό αλλά λειτουργικό κομμάτι του IEEE 1451.2 το οποίο προσπαθήσαμε να κρατήσουμε όσο πιο απλό και λιτό γίνεται καθώς προτυποποιήθηκε το 1997 οπότε σε αρκετά σημεία δεν είναι καν συμβατό με το IEEE 1451.0 με το οποίο έχουν διαφορά πάνω από δέκα χρόνια. Επίσης χρησιμοποιήσαμε κατά βάση τις έτοιμες λύσεις επικοινωνίας μέσω USB που προσφέρει η Java.



Σχήμα 23: Αρχιτεκτονική #2 της υλοποίησής μας

Όπως παρατηρούμε στο Σχήμα 23, στο NCAP υπάρχει ένα Java Servlet (το οποίο εκτελείται σε ένα διακομιστή HTTP) μέσω του οποίου οι εξωτερικές εφαρμογές και οι χρήστες μπορούν να έχουν πρόσβαση στο σύστημα. Όταν γίνει κάποια αίτηση, μετασχηματίζεται σε ερώτηση (COM) συμβατή με το IEEE 1451.0 και στέλνεται στο IEEE 1451.2 επίπεδο του NCAP υπό μορφή μη προσημασμένων bytes (UInt8). Το IEEE 1451.2 κωδικοποιεί τα bytes σε ένα μήνυμα με κατάλληλη μορφή και στέλνει στο TIM τα bytes του μηνύματος. Το TIM καθώς λαμβάνει το ρεύμα από bytes και μόλις αντιληφθεί την ύπαρξη ολοκληρωμένου μηνύματος, το μετατρέπει σε μη προσημασμένα bytes, βγάζοντας την επιπλέον πληροφορία του πρωτοκόλλου επικοινωνίας και τις στέλνει στο IEEE 1451.0 του TIM. Έπειτα αποκωδικοποιείται το μήνυμα και εκτελείται η επιθυμητή ενέργεια. Αν περιλαμβάνει την αποστολή TEDS, τότε το TIM τα ζητάει μέσω της αντίστοιχης διεπαφής που έχει υλοποιηθεί (είναι

αποθηκευμένα ως αρχεία μέσα στο SunSPOT) ενώ αν περιλαμβάνει την ανάγνωση τιμής κάποιου αισθητήρα, αυτό γίνεται μέσω μίας διεπαφής που επιστρέφει τις τιμές σε προκαθορισμένη μορφή. Και στις δύο περιπτώσεις σχηματίζεται κάποιου είδους απάντηση (REP) η οποία ακολουθεί την αντίθετη ακριβώς κατεύθυνση και ενέργειες από την ερώτηση. Όταν φτάσει στο IEEE 1451.0 του NCAP, κωδικοποιείται σε γλώσσα XML και στέλνεται στον χρήστη ή την εφαρμογή που είχε κάνει την αντίστοιχη αίτηση.

Επίσης στο NCAP εκτελείται το νήμα ανακάλυψης, το οποίο είναι υπεύθυνο για την ανακάλυψη νέων TIM (SunSPOT). Αρχίζει να εκτελείται μετά την πρώτη HTTP αίτηση που λαμβάνει το συγκεκριμένο NCAP.

4.3 Υλοποίηση

Η υλοποίηση του πρωτοτύπου αποτελείται από 130 κλάσεις γραμμένες σε γλώσσα προγραμματισμού Java και περισσότερες από 14.000 γραμμές κώδικα. Ο κώδικας που εκτελείται στο TIM δεν ξεπερνάει τις 7.000 γραμμές, οπότε δεν έχουμε προβλήματα με την περιορισμένη μνήμη των κόμβων αισθητήρων.

4.3.1 Μέρη

Η υλοποίησή μας αποτελείται από τέσσερα μέρη:

- **CommonLayers:** είναι γραμμένο σε J2ME η οποία είναι συμβατή με τα SunSPOT και περιλαμβάνει όλα τα κοινά στοιχεία μεταξύ του NCAP και του TIM, όπως είναι οι τύποι δεδομένων, τα μηνύματα και οι βασικές κλάσεις σειριακής επικοινωνίας. Το ότι υλοποιήθηκε σε J2ME έχει το μεγάλο πλεονέκτημα ότι χρειάστηκε να γραφτεί μία και μόνο φορά και χρησιμοποιείται αυτούσιο και στο NCAP αλλά και στα TIM χωρίς προβλήματα ασυμβατότητας λόγω διαφορετικών εκδόσεων μεταξύ τους. Το μειονέκτημα είναι ότι είχαμε στη διάθεσή μας ένα περιορισμένο αριθμό από διαθέσιμα πακέτα κλάσεων (για παράδειγμα δεν είχαμε διαθέσιμη τη δυνατότητα απαριθμήσεων και μεγάλης ποικιλίας δομών συλλογής και οργάνωσης δεδομένων), οπότε κάποια πράγματα ήταν λίγο πιο δύσκολα στην υλοποίηση ή ήθελαν περισσότερη δουλειά με το να χρησιμοποιούσαμε J2SE από την πλευρά του NCAP.
- **COM:** περιέχει μία και μόνο εκτελέσιμη κλάση η οποία απλά εκτυπώνει τις COM θύρες που βρίσκει ενεργές. Χρησιμοποιείται για την ανακάλυψη των TIM.

- **NCAP:** είναι μία εφαρμογή ιστού (web application) η οποία υλοποιεί στην ουσία τη λειτουργία του NCAP. Χρησιμοποιεί τα CommonLayers και COM. Περιλαμβάνει ένα servlet το οποίο δέχεται και επεξεργάζεται όλες τις HTTP αιτήσεις επιστρέφοντας τις απαντήσεις κωδικοποιημένες σε XML.
- **TIM:** είναι γραμμένο σε J2ME (συμβατή με τα SunSPOTs) και χρησιμοποιεί το κομμάτι CommonLayers. Εκτελείται σε κάθε SunSPOT και στην ουσία υλοποιεί τη λειτουργία του TIM.

Ακολουθώντας την παραπάνω αρχιτεκτονική, σε περίπτωση που θέλουμε να κάνουμε χρήση κάποιας άλλης συσκευής ως TIM, χρειάζεται να αλλάξουμε μόνο κάποια κομμάτια κώδικα στο μέρος του TIM. Συγκεκριμένα αυτά που έχουν σχέση με την επικοινωνία της συσκευής μέσω της USB θύρας, κάποια πράγματα σχετικά τον τρόπο αποθήκευσης και προσπέλασης των TEDS και τη λήψη μετρήσεων από τους αντίστοιχους αισθητήρες.

4.4 Λεπτομέρειες για Ειδικά Μέρη της Υλοποίησης

4.4.1 Ανακάλυψη των TIM

Το πρότυπο αναφέρει ότι θα πρέπει τα TIM να ανακαλύπτονται αυτόματα αλλά δεν κάνει καμία αναφορά στον τρόπο με τον οποίο θα γίνει αυτό. Εμείς το υλοποιήσαμε με μοναδικές προϋποθέσεις ότι τα TIM θα συνδέονται μέσω της USB θύρας τους με τον υπολογιστή ο οποίος παίζει τον ρόλο του NCAP και ότι ο υπολογιστής έχει ως λειτουργικό σύστημα τα Windows.

Όταν συνδεθεί κάποιο SunSPOT μέσω της USB θύρας με έναν υπολογιστή που έχει Windows λειτουργικό σύστημα (και εφόσον υπάρχουν οι απαραίτητοι οδηγοί), αυτόματα δημιουργείται μία COM θύρα επικοινωνίας, η οποία προσφέρει δυνατότητα σειριακής επικοινωνίας με τη συσκευή. Το πρόβλημα είναι ότι πρέπει να γνωρίζουμε κάθε SunSPOT σε ποια θύρα επικοινωνίας έχει αντιστοιχηθεί. Για να λύσουμε αυτό το πρόβλημα, χρησιμοποιήσαμε το jUSB [25], [26] το οποίο παρέχει κάποιες δυνατότητες πρόσβασης σε USB συσκευές με τη χρήση της Java. Δεν χρησιμοποιήσαμε τις δυνατότητες επικοινωνίας αλλά μόνο τις δυνατότητες απαρίθμησης και συμβάντων.

Η ιδέα είναι μόλις (απο)συνδέεται κάποιο SunSPOT, να ειδοποιείται η εφαρμογή και να βρίσκει τη διαφορά μεταξύ της αποθηκευμένης λίστας με τις θύρες επικοινωνίας και της τρέχουσας κατάστασης. Με αυτό τον τρόπο μπορούμε πολύ εύκολα να αντιληφθούμε

ότι όταν έχουμε κάποιο συμβάν αποσύνδεσης, σημαίνει ότι αποσυνδέθηκε αυτή η θύρα που υπάρχει στην αποθηκευμένη λίστα και όχι στην τρέχουσα, ενώ αν έχουμε κάποιο συμβάν σύνδεσης τότε η θύρα που θα αντιστοιχηθεί στο καινούργιο SunSPOT είναι αυτή που υπάρχει στην τρέχουσα λίστα αλλά όχι στην αποθηκευμένη.

Για να δουλεύει χωρίς προβλήματα η παραπάνω υλοποίηση χρειάζεται πριν συνδεθεί οποιοδήποτε SunSPOT να ενημερωθεί η εφαρμογή για τη λίστα των ενεργών θυρών, οι οποίες είναι προφανώς θύρες που χρησιμοποιούνται από άλλες συσκευές του συστήματος.

Επίσης για να μπορεί η εφαρμογή μας να ενημερώνεται για συμβάντα σύνδεσης και αποσύνδεσης, θα πρέπει να είναι συνεχώς ενεργή. Για να το καταφέρουμε αυτό μετά την πρώτη αίτηση που θα δεχθεί, ξεκινάει το νήμα ανακάλυψης, το οποίο είναι υπεύθυνο μόνο για τη λήψη και επεξεργασία των συμβάντων.

Ένα άλλο πρόβλημα που υπήρχε ήταν ότι κατά τη διάρκεια εκτέλεσης της εφαρμογής, όσες φορές και να ζητούσαμε τη λίστα των θυρών επικοινωνίας, μας επέστρεφε πάντα τις ίδιες και συγκεκριμένα αυτές που ήταν ενεργές όταν ξεκινούσε η εκτέλεση της εφαρμογής. Για να ξεπεράσουμε αυτό το πρόβλημα και με σκοπό να χρησιμοποιήσουμε αποκλειστικά Java, φτιάξαμε ένα εκτελέσιμο JAR (Java Archive) αρχείο, το οποίο απλά εκτυπώνει τις διαθέσιμες θύρες και το εκτελούμε μέσα από την εφαρμογή μας. Με αυτό τον τρόπο, λαμβάνουμε την ενημερωμένη λίστα των θυρών χωρίς προβλήματα.

4.4.2 Χαμηλού επιπέδου επικοινωνία

Καθώς το IEEE 1451.2 περιγράφει πράγματα που τα περισσότερα αναφέρονται και στο IEEE 1451.0 και λόγω του ότι δίνει πολλές λεπτομέρειες για την διεπαφή από την πλευρά του υλικού (κάτι το οποίο δεν μας ενδιαφέρει), δεν το ακολουθήσαμε σχεδόν καθόλου για την υλοποίηση του πρωτοκόλλου χαμηλού επιπέδου επικοινωνίας μεταξύ του NCAP και των TIM.

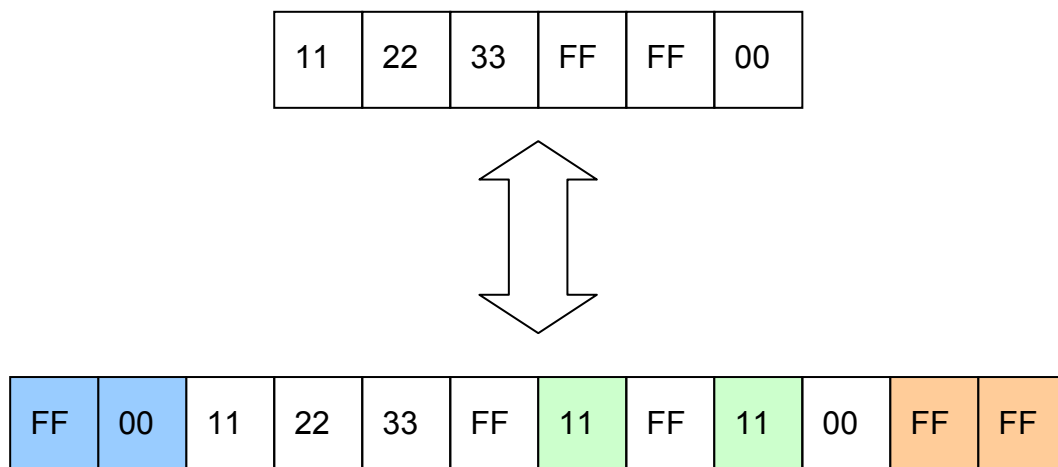
Για τις ανάγκες της επικοινωνίας χαμηλού επιπέδου, υλοποιήσαμε ένα είδος σειριακού πρωτοκόλλου δυφιοσυλλαβικής πλήρωσης (byte stuffing). Καθώς μεταξύ NCAP και TIM έχουμε στη διάθεσή μας μόνο μία σειριακή αμφίδρομη επικοινωνία μέσω μίας θύρας επικοινωνίας, έπρεπε με κάποιο τρόπο να οριοθετήσουμε τα μηνύματα, καθώς μέσα από το ίδιο κανάλι επικοινωνίας περνάνε και τα μηνύματα του πρωτοκόλλου αλλά και τα αλφαριθμητικά μηνύματα αποσφαλμάτωσης του TIM.

Για να το καταφέρουμε αυτό ορίσαμε τους παρακάτω κανόνες:

- πριν από κάθε μήνυμα προστίθενται δύο bytes με τιμές 0xFF, 0x00
- μετά από κάθε μήνυμα προστίθενται δύο bytes με τιμές 0xFF, 0xFF
- αν μέσα στο σώμα του μηνύματος εμφανιστεί κάποιο byte με τιμή 0xFF, προστίθεται μετά από αυτό άλλο ένα με τιμή 0x11

Με βάση τους παραπάνω κανόνες μπορούμε να δούμε στο Σχήμα 24, ένα παράδειγμα αποστολής δεδομένων μέσω της θύρας επικοινωνίας. Προφανώς ακολουθείται η ακριβώς αντίθετη διαδικασία κατά τη λήψη του μηνύματος.

Για να μπορούμε να διαχειριζόμαστε και τα μηνύματα κειμένου αποσφαλμάτωσης, τα οποία είναι αλφαριθμητικές αλληλουχίες οι οποίες τερματίζουν με ένα χαρακτήρα αλλαγής γραμμής, το πρωτόκολλο εφόσον δεχτεί κάποια αλληλουχία από bytes, η οποία δεν ξεκινάει με το προοίμιο που έχουμε ορίσει και κάποια στιγμή έχει ένα χαρακτήρα αλλαγής γραμμής ('\n'), θεωρεί ότι είναι κάποιο μήνυμα αποσφαλμάτωσης και το μεταφέρει στο παραπάνω επίπεδο για να εμφανιστεί στο χρήστη, εφόσον έχει ενεργοποιημένη αυτή τη δυνατότητα.



Σχήμα 24: Πρωτόκολλο επικοινωνίας χαμηλού επιπέδου

4.4.3 Υλοποίηση TEDS

Όπως έχουμε περιγράψει στο αντίστοιχο κομμάτι, τα TEDS αποτελούνται από datablocks τα οποία έχουν τη μορφή τύπος-μέγεθος τιμής-τιμή. Ενώ το πεδίο του τύπου έχει προκαθορισμένο μέγεθος ίσο με ένα byte, το πεδίο της τιμής είναι μεταβλητό και καθορίζεται από την τιμή στο πεδίο του μεγέθους τιμής. Το μέγεθος του μεγέθους τιμής

δεν είναι προκαθορισμένο αλλά καθορίζεται από ένα ειδικό datablock που υπάρχει σε κάθε TEDS με όνομα TEDSID και τύπο ίσο με 4. Επειδή το μεταβλητό μέγεθος του πεδίου μεγέθους τιμής θα έκανε αρκετά πολύπλοκη και πιο χρονοβόρα την υλοποίηση και καθώς σύμφωνα και με το πρότυπο η τιμή αυτή είναι συνήθως ίση με ένα, το έχουμε προκαθορίσει ίσο με ένα.

Καθώς ο αριθμός των datablocks είναι δυναμικός, χρησιμοποιήσαμε έναν πίνακα κατατεμαχισμού (hashtable) για την αποθήκευσή τους, μία από τις ελάχιστες δομές οργάνωσης και αποθήκευσης δεδομένων που είχαμε διαθέσιμες σε Java ME. Κάθε datablock προστίθεται στη δομή με κλειδί την τιμή του τύπου της.

Από όλους τους διαφορετικούς τύπους των TEDS έχουμε υλοποιήσει τα MetaTEDS και τα TransducerChannelTEDS τα οποία είναι τα άκρως απαραίτητα για την ορθή λειτουργία του συστήματος.

Στα MetaTEDS υπάρχει ένα πεδίο το οποίο αποτελείται από είκοσι δεκαεξαδικά ψηφία και το οποίο είναι το μοναδικό αναγνωριστικό του TIM και ονομάζεται Καθολικό Μοναδικό Αναγνωριστικό (Universal Unique IDentification – UUID). Για να είμαστε σίγουροι ότι θα είναι μοναδικό σε κάθε TIM που θα χρησιμοποιήσουμε, το έχουμε ορίσει να έχει ως τιμή τέσσερα μηδενικά ακολουθούμενα από τη διεύθυνση Ελέγχου Πρόσβασης Μέσου (Media Access Control – MAC) του SunSPOT, η οποία είναι μοναδική σε κάθε δικτυακή συσκευή.

4.4.4 Αποθήκευση TEDS

Τα TEDS σύμφωνα με το πρότυπο είναι αποθηκευμένα μέσα στο TIM. Για να το καταφέρουμε αυτό, διαπιστώσαμε ότι μπορούσαμε να προσθέτουμε αρχεία μέσα στον κατάλογο “resources” της εφαρμογής που γράψαμε για το TIM και κατά τη μεταφορά της στο SunSPOT, μεταφέρονταν και αυτά τα αρχεία τα οποία έπειτα μπορούσαμε να προσπελάσουμε μέσα από το SunSPOT. Επιλέξαμε να τα αποθηκεύσουμε ως bytes χωρισμένα με κόμμα.

4.4.5 Κωδικοποίηση/Αποκωδικοποίηση

Ένα σημαντικό κομμάτι της υλοποίησης είναι η κωδικοποίηση όλων των ερωτήσεων, των απαντήσεων αλλά και των TEDS σε αλληλουχία από bytes και η αποκωδικοποίηση των αλληλουχιών στην αντίστοιχη οντότητα που αντιπροσωπεύουν. Για την

κωδικοποίηση έχει φτιαχτεί μία κλάση η οποία παίρνει ως παράμετρο οποιαδήποτε από τις προηγούμενες οντότητες και επιστρέφει μία αλληλουχία από bytes τα οποία μπορούμε να περάσουμε για αποστολή στο IEEE 1451.2. Η αποκωδικοποίηση είναι μία πολύ πιο περίπλοκη και σύνθετη διαδικασία, καθώς η αντίστοιχη κλάση θα πρέπει να συμπεράνει αρχικά το είδος της οντότητας η οποία είναι κωδικοποιημένη και έπειτα να εξάγει σωστά τα επιμέρους στοιχεία της.

4.4.6 Διάβασμα τιμών αισθητήρων

Όσον αφορά το διάβασμα τιμών από τους αισθητήρες, θέλαμε να προσθέσουμε ένα επίπεδο αφαίρεσης ώστε να είναι εύκολη η λήψη τιμών από νέους αισθητήρες ή ακόμα και από συστήματα όπου ως TIM θα χρησιμοποιηθούν συσκευές διαφορετικές από SunSPOT. Ως λύση στο παραπάνω πρόβλημα υλοποιήσαμε μία διεπαφή (interface) στην Java η οποία περιγράφει μόνο μία συνάρτηση η οποία παίρνει ως παράμετρο τον αριθμό του καναλιού μορφοτροπέα και επιστρέφει την τιμή κωδικοποιημένη ως πίνακα μη προσημασμένων bytes. Έπειτα υλοποιούμε αυτή την συνάρτηση στην οποία προσθέτουμε την χαμηλού επιπέδου επικοινωνία με τους αισθητήρες του SunSPOT και την κωδικοποίηση των τιμών τους στη μορφή που ορίζεται στη διεπαφή.

4.4.7 Αποσφαλμάτωση

Καθώς η υλοποίηση ενός τέτοιου μεγέθους συστήματος είναι αρκετά περίπλοκη, είναι αρκετά συχνό φαινόμενο τα λάθη ή οι παραλείψεις στον προγραμματισμό. Κατά την ανάπτυξη την υποδομής, αλλά και μετά, είναι απαραίτητο να γνωρίζουμε όσες παραπάνω πληροφορίες μπορούμε για την κατάσταση των επιμέρους συστημάτων και κυρίως όταν συμβεί κάποιο σφάλμα ή λάβουμε κάποιο μη αναμενόμενο αποτέλεσμα, να μπορούμε να διαπιστώσουμε εύκολα την πηγή και τον λόγο του προβλήματος. Ενώ από την πλευρά του NCAP θα μπορούσαμε να πούμε ότι τα πράγματα ήταν σχετικά εύκολα, καθώς σε περίπτωση προβλήματος μπορούσαμε να εκτυπώνουμε διαγνωστικά μηνύματα και να τα βλέπουμε είτε στην κονσόλα μας, είτε στο φυλλομετρητή (browser) ή σε κάποιο αρχείο, από την πλευρά του TIM τα πράγματα είναι πολύ πιο δύσκολα, καθώς δεν διαθέτει κάποιο όργανο για απεικόνιση πληροφορίας στο χρήστη (εκτός των LEDs που διαθέτει με τα οποία όμως μπορούμε να αποτυπώσουμε ελάχιστες πληροφορίες). Ο μοναδικός τρόπος ενημέρωσης του χρήστη είναι μέσω του μοναδικού

τρόπου επικοινωνίας μαζί του, που είναι η USB σύνδεση με τον υπολογιστή. Όπως αναφέραμε και παραπάνω για να συμβεί αυτό έπρεπε με κάποιο τρόπο τα μηνύματα αυτά να μην επηρεάζουν το πρωτόκολλο του ίδιου του προτύπου. Επίσης έχουμε προσθέσει τη δυνατότητα για απενεργοποίηση των μηνυμάτων αποσφαλμάτωσης είτε στο TIM (αλλάζοντας τη τιμή σε μία παράμετρο στο Manifest αρχείο) ή στο NCAP (αλλάζοντας τη τιμή σε μία παράμετρο στο web.xml αρχείο).

4.5 Εγκατάσταση

Για να εγκατασταθεί και να λειτουργήσει σωστά η εφαρμογή, χρειάζονται κάποια βήματα:

- εγκατάσταση του οδηγού και των απαραίτητων βιβλιοθηκών για επικοινωνία με τα SunSPOT [49]
- εγκατάσταση του πακέτου επικοινωνίας της Java με τις COM θύρες επικοινωνίας [50]
- εγκατάσταση του πακέτου επικοινωνίας της Java με τις USB θύρες [51]
- εγκατάσταση ενός servlet container (όπως ο Tomcat της Apache [52])
- ενημέρωση, στην κλάση `IEEE1451.layer0.discovery.Discover` του NCAP, της τοποθεσίας του εκτελέσιμου `COM.jar`

4.6 Προσθήκη αισθητήρα

Στην υλοποίησή μας έχουμε χρησιμοποιήσει δύο αισθητήρες, θερμοκρασίας και τάσης της μπαταρίας του SunSPOT. Σε περίπτωση που θέλουμε να προσθέσουμε στο πρωτότυπο κάποιον επιπλέον αισθητήρα, θα πρέπει να διεξάγουμε τα παρακάτω βήματα:

- ανανέωση της τιμής του αριθμού των μορφοτροπέων στα MetaTEDS
- δημιουργία και ενσωμάτωση των ChanTEDS με τις πληροφορίες για το νέο αισθητήρα
- ανανέωση της κλάσης που υπάρχει στο TIM και σχετίζεται με τη λήψη μετρήσεων ώστε να μπορεί να παίρνει μετρήσεις από το νέο αισθητήρα

5. ΣΥΜΠΕΡΑΣΜΑΤΑ ΚΑΙ ΑΝΟΙΧΤΑ ΘΕΜΑΤΑ

5.1 Συμπεράσματα

Στην παρούσα διπλωματική εργασία πραγματοποιήθηκε μία εκτενής έρευνα των ευφυών αισθητήρων και του ιστού αισθητήρων και παρουσιάστηκαν τα χαρακτηριστικά τους, οι αρχιτεκτονικές των συστημάτων και οι εφαρμογές και οι λύσεις που δίνουν στην κοινωνία. Επίσης περιγράφηκαν τα σημαντικότερα πρότυπα του είδους, τα οποία είναι κοινώς αποδεκτά σήμερα. Τέλος έγινε περιγραφή της υλοποίησης ενός πρωτοτύπου, του πιο σημαντικού προτύπου από αυτά, που είναι το IEEE 1451.

Η επιστημονική συνεισφορά της εργασίας μπορεί να συνοψιστεί στα ακόλουθα:

- *Σχεδίαση και υλοποίηση ολοκληρωμένου συστήματος ευφυών αισθητήρων.* Το σύστημα που αναπτύχθηκε βασίζεται στο πρότυπο IEEE 1451. Έπειτα από μελέτη διαπιστώθηκε ότι έχουν πραγματοποιηθεί λίγες υλοποιήσεις, αλλά καμία ολοκληρωμένη και πλήρως λειτουργική. Συμπεράναμε ότι το πρότυπο είναι ορθό αν και έχει διάφορα σημεία στα οποία μπορεί να βελτιωθεί, για τα οποία περιγράφουμε τις λύσεις που δόθηκαν στη συγκεκριμένη υλοποίηση.
- *Επικοινωνία με το σύστημα με τη χρήση ανοιχτών διεπαφών.* Η πρόσβαση στο σύστημα γίνεται με τη χρήση πλήρως ανοιχτών διεπαφών οι οποίες περιγράφονται διεξοδικά από το πρότυπο.
- *Ευέλικτη υλοποίηση με επίπεδα αφαίρεσης.* Λόγω της προσθήκης επιπέδων αφαίρεσης, πολύ εύκολα και με ελάχιστες αλλαγές μπορεί να χρησιμοποιηθεί σε διαφορετικά λειτουργικά συστήματα ή να γίνει χρήση διαφορετικών κόμβων αισθητήρων.

5.2 Ανοιχτά θέματα

Στα πλαίσια της συγκεκριμένης εργασίας, διαπιστώθηκαν αρκετά ανοιχτά θέματα, τα οποία είναι ενδιαφέρον να διερευνηθούν στο μέλλον. Αρχικά, θα μπορούσε να προστεθεί άλλος ένα τρόπος πρόσβασης στην υποδομή, όπως κάνοντας χρήση υπηρεσιών ιστού, οι οποίες κερδίζουν όλο και μεγαλύτερο έδαφος έναντι των κλασικών τρόπων επικοινωνίας.

Επίσης θα είχε ιδιαίτερο ενδιαφέρον η προσθήκη δυνατότητας σκανδάλισης (triggering), ώστε η εφαρμογή να μπορεί να ειδοποιείται ανά τακτά χρονικά διαστήματα για την τιμή κάποιου αισθητήρα ή όταν αυτή βγει εκτός ορίων, τα οποία ή ίδια έχει θέσει. Το HTTP

API υποστηρίζει μία δυνατότητα η οποία αναφέρεται ως «AppCallback» κατά την οποία η εφαρμογή, κατά την HTTP αίτηση, μπορεί να ενημερώνει το σύστημα ότι επιθυμεί κάποιου είδους ενημέρωσης σε συγκεκριμένες καταστάσεις. Δυστυχώς η συγκεκριμένη δυνατότητα δεν περιγράφεται περαιτέρω από το πρότυπο.

Ενδιαφέρουσα θα ήταν η ένταξη στο σύστημα διαφορετικών κόμβων αισθητήρων, ώστε να μπορεί να παρατηρηθεί η ευελιξία του συστήματος και η ικανότητα απρόσκοπτης επικοινωνίας μαζί τους. Πέρα από διαφορετικούς κόμβους, θα μπορούσε να επεκταθεί η δυνατότητα ανακάλυψης αισθητήρων σε διαφορετικά λειτουργικά συστήματα (πέρα από Windows), όπως για παράδειγμα Linux.

Επίσης θα είχε ιδιαίτερο ενδιαφέρον να γίνει μία υλοποίηση του SWE προτύπου και έπειτα να γίνει συνδυασμός με τη δική μας υλοποίηση η οποία θα μπορούσε να αναλάβει την λειτουργικότητα των κατώτερων στρωμάτων του συστήματος.

Τέλος, θα ήταν ενδιαφέρουσα η χρήση της υλοποίησής μας σε κάποιο πραγματικό σύστημα αισθητήρων. Θα μπορούσαν να εξαχθούν πολλά συμπεράσματα ως προς την απόδοση, την ορθότητα και την λειτουργικότητα της συγκεκριμένης λύσης.

ΠΙΝΑΚΑΣ ΟΡΟΛΟΓΙΑΣ

Ξενόγλωσσος όρος	Ελληνικός Όρος
Actuator	Εκκινητήρας
Application Programming Interface	Διεπαφή Προγραμματισμού Εφαρμογής
Byte	Δυφιοσυλλαβή
Byte Stream	Ρεύμα Δυφιοσυλλαβών
Byte Stuffing	Δυφιοσυλλαβική Πλήρωση
Callback	Αναστροφή
Calibration Factor	Παράγοντας Βαθμονόμησης
Checksum	Άθροισμα Ελέγχου
Client-Server	Πελάτης-Εξυπηρετητής
Datablock	Ομάδα Δεδομένων
Fabric	Δικτυοδομή
Flash Memory	Αστραπιαία Μνήμη
Hardware	Υλικό
Hash	Κατακερματισμός
Host	Ξενιστής
Hot Swap	Θερμή Εναλλαγή
Instant Messaging	Στιγμιαία Μηνυματοδοσία
Middleware	Ενδιάμεσο Λογισμικό
Module	Δομοστοιχείο
Native code	Πρωτογενής κώδικας
Plug and Play	Βυσμάτωση και Λειτουργία
Publish-Subscribe	Δημοσιοποίηση-Εγγραφή
Radio Frequency Identification	Ραδιοσυχνотική Αναγνώριση
Sensor Grid	Πλέγμα Αισθητήρων
Sensor Web	Ιστός Αισθητήρων
Serial Peripheral Interface	Σειριακή Περιφερειακή Διεπαφή
Smart Sensor	Ευφυής/Εξυπνος Αισθητήρας
State-ful/less	(Α)καταστασικός
Transducer	Μορφοτροπέας
Triggering	Σκανδάλη
Web Services	Υπηρεσίες Ιστού

ΣΥΝΤΜΗΣΕΙΣ – ΑΡΚΤΙΚΟΛΕΞΑ – ΑΚΡΩΝΥΜΙΑ

API	Application Programming Interface
CAN	Controller area network
EEPROM	Electrically Erasable Programmable Read-Only Memory
HTTP	HyperText Transfer Protocol
IEEE	Institute of Electrical and Electronics Engineers
J2ME	Java 2 Micro Edition
JAR	Java ARchive
LED	Light-Emitting Diode
MAC	Media Access Control
MIDlet	Mobile Information Device profile application
NASA	National Aeronautics and Space Administration
NCAP	Network Capable Application Processor
NIST	U.S. National Institute of Standards and Technology
O&M	Observations & Measurements
OGC	Open Geospatial Consortium
OSWA	Open Sensor Web Architecture
RAM	Random-Access Memory
RFID	Radio Frequency IDentification
SAS	Sensor Alert Service
SensorML	Sensor Model Language
SI	Système International
SOA	Service Oriented Architecture
SOAP	Simple Object Access Protocol
SOS	Sensor Observation Service
SPI	Serial Peripheral Interface
SPS	Sensor Planning Service
SQL	Structured Query Language
STIM/TIM	(Smart) Transducer Interface Module
STWS	Smart Transducer Web Services
SunSPOT	Sun's Small Programmable Object Technology
SWE	Sensor Web Enablement
TBC	Transducer Bus Controller

TBIM	Transducer Bus Interface Modules
TEDS	Transducer Electronic Data Sheet
TII	Transducer Independent Interface
TLV	Type-Length-Value
TML	Transducer Markup Language
USB	Universal Serial Bus
UUID	Universal Unique IDentification
WNS	Web Notification Services
WSDL	Web Services Description Language
WSRF	Web Services Resource Framework
XML	eXtensible Markup Language
XMPP	Extensible Messaging and Presence Protocol

ΑΝΑΦΟΡΕΣ

- [1] F. Randy, "Understanding Smart Sensors", 2nd edition, Artech House, 2000
- [2] J. Mark and P. Hufnagel, The IEEE 1451.4 Standard for Smart Transducers, IEEE 1451.4 Standard Working Group, 2004
- [3] J.L. Hill, System Architecture for Wireless Sensor Networks, master's thesis, Dept. Computer Science, University of California, Berkeley, 2003
- [4] J. Wiczer, Smart Interfaces for Sensors, Sensor Synergy, Proceeding Sensor Expo 2001, Chicago, Jun. 2001
- [5] R.N. Johnson, Proposed IEEE Standard P1451.0, Telemonitor, Inc., Columbia, Maryland, Jun. 2003
- [6] Lee, K., "IEEE 1451: A Standard in Support of Smart Transducer Networking", IEEE Instrumentation and Measurement Technology Conference Baltimore, MD USA, May 1-4, 2000
- [7] V. Tsetos, N. Silvestros and S. Hadjiefthymiades, Collaborative Sensing over Smart Sensors, 2nd Student Workshop on Wireless Sensor Networks at Athens, Greece, Oct. 2009
- [8] An Overview of IEEE 1451.4 Transducer Electronic Data Sheets (TEDS), National Instruments
- [9] NIST IEEE-P1451 Draft Standard Home Page; <http://ieee1451.nist.gov/> [Προσπελάστηκε 31/8/2010]
- [10] Darold Wobschall, IEEE 1451 -- A UNIVERSAL TRANSDUCER PROTOCOL STANDARD
- [11] Jeff Burch, Proposed IEEE 1451.0 to P1451.5 Interface, Agilent Technologies
- [12] A. Mainwaring, J. Polastre, R. Szewczyk, D. Culler and J. Anderson, Wireless sensor networks for habitat monitoring, ACM International Workshop on Wireless Sensor Networks and Applications (WSNA'02), Atlanta, Sep. 2002
- [13] L. Schwiebert, S.K.S. Gupta and J. Weinmann, Research challenges in wireless networks of biomedical sensors, Wayne State University, Arizona State University, USA 2001
- [14] M. Bahrepour, N. Meratnia and P. Havinga, Automatic Fire Detection: a survey from wireless sensor network perspective, Pervasive Systems Group, University of Twente
- [15] L. Yu, N. Wang and X. Meng, Real-time Forest Fire Detection with Wireless Sensor Networks, 2005
- [16] ALERT Systems Organization HomePage; <http://www.alertsystems.org/> [Προσπελάστηκε 3/9/2010]
- [17] The ZebraNet Wildlife Tracker; <http://www.princeton.edu/~mrm/zebranet.html> [Προσπελάστηκε 3/9/2010]
- [18] P. Sikka, P. Corke and L. Overs, Wireless sensor devices for animal tracking and control, CSIRO ICT Centre, 29th Annual IEEE International Conference on Local Computer Networks (LCN'04), Australia, 2004
- [19] N. Shaylor, D.N. Simon and W.R. Bush, A Java Virtual Machine Architecture for Very Small Devices, Sun Microsystems Research Laboratories, Mountain View, USA
- [20] J. Iqbal and F. Moughal, Wireless Sensor Network Setup, master's thesis, Dept. Computer Systems Engineering, Halmstad University, Jun. 2010
- [21] K. Aberer, M. Hauswirth and A. Salehi, Zero-programming Sensor Network Deployment, Ecole Polytechnique Federale de Lausanne (EPFL), Lausanne, Switzerland, Digital Enterprise Research Institute (DERI), National University of Ireland, Galway
- [22] OGC (Open Geospatial Consortium) Website; <http://www.opengeospatial.org/> [Προσπελάστηκε 10/9/2010]
- [23] H. Jin and W. Jiang, "Handbook of Research on Developments and Trends in Wireless Sensor Networks: From Principle to Practice", Hershey, New York
- [24] J. Wiczer, A Unifying Standard for Interfacing Transducers to Networks – IEEE-1451.0, Sensor Synergy, Inc., Presented at ISA Expo 2005: Automation + Control
- [25] Java and USB | Java.net; <http://today.java.net/article/2006/07/05/java-and-usb> [Προσπελάστηκε 17/9/2010]
- [26] jUSB - Java USB API for Windows; <http://www.steelbrothers.ch/jusb/> [Προσπελάστηκε 17/9/2010]
- [27] IEEE Std 1451.0, IEEE Standard for a Smart Transducer Interface for Sensors and Actuators—Common Functions, Communication Protocols, and Transducer Electronic Data Sheet (TEDS) Formats, Sep. 2007
- [28] R.L. Oostdyk, C.T. Mata and J.M. Perotti, A Kennedy Space Center Implementation of IEEE 1451 Networked Smart Sensors and Lessons Learned, Kennedy Space Center, NASA
- [29] K. Lee, A Synopsis of the IEEE P1451- Standards for Smart Transducer Communication, National Institute of Standards and Technology
- [30] H.M.G. Ramos and O. Postolache, A Virtual Instrument to Test Smart Transducer Interface Modules (STIMs), IEEE Transactions on Instrumentation and Measurement, vol. 53, no. 4, Aug. 2004, pp. 1232-1239
- [31] N. Kularatna, An Environmental Air Pollution Monitoring System Based on the IEEE 1451 Standard for Low Cost Requirements, IEEE Sensors Journal, vol. 8, no. 4, Apr. 2008, pp. 415-422

- [32] T. Cummins, An IEEE 1451 Standard Transducer Interface Chip with 12-b ADC, Two 12-b DAC's, 10-kB Flash EEPROM, and 8-b Microcontroller, IEEE Journal of Solid-State Circuits, vol. 33, no. 12, Dec. 1998
- [33] D. Wobschall and Esensors Inc., An Implementation of IEEE 1451 NCAP for Internet Access of Serial Port-Based Sensors, Simd02 - Sensors for Industry Conference, Houston, Texas, USA, 19-21 Nov. 2002
- [34] V. Kochan, K. Lee, R. Kochan and A. Sachenko, Approach to Improvement the Network Capable Application Processor Compatible with IEEE 1451 Standard, IEEE International Workshop on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, Lviv, Ukraine, 8-10 Sep. 2003
- [35] L. Camara, O. Ruiz and J. Samitier, Complete IEEE-1451 Node, STIM and NCAP, Implemented for a CAN Network, Department of Electronics, University of Barcelona
- [36] L. Camara, O. Ruiz, A. Herms, J. Samitier and J. Bosch, Automatic generation of intelligent instruments for IEEE 1451, Department of Electronics, University of Barcelona
- [37] W. Liu, Design of TEDS Writer, Reader and Testing System for Transducer Interface Modules based on the IEEE 1451 Standard, Ph.D. thesis, Department of Electrical Engineering, State University of New York, Buffalo, 2006
- [38] D. Wobschall, W. Liu and W.S. Poh, An IEEE 1451 TEDS Compiler and Serial Network Compliance Tester, Esensors Inc. and University at Buffalo, Sensor Standards Harmonization -- Sensors Expo Chicago, Jun. 2006
- [39] K.C. Lee, M.H. Kim, S. Lee and H.H. Lee, IEEE 1451 based Smart Module for In-Vehicle Networking Systems of Intelligent Vehicles
- [40] K. Wooshik, L. Suyoung, A. Jinsoo, N. Jiyoung and K. Namhyun, Integration of IEEE 1451 and HL7 Exchanging Information for Patients' Sensor Data, Korea, 2009
- [41] K. Jaehwan, H. Jeongil and K. Wooshick, A Study on the Implementation of IEEE1451 for e-Health, Korea, IFMBE Proceedings Vol. 14/6
- [42] IEEE Std 1451.2, IEEE Standard for a Smart Transducer Interface for Sensors and Actuators-Transducer to Microprocessor Communication Protocols and Transducer Electronic Data Sheet (TEDS) Formats, Sep. 1997
- [43] E.Y. Song and K.B. Lee, Sensor Network based on IEEE 1451.0 and IEEE p1451.2-RS232, National Institute of Standards and Technology, I2MTC 2008 - IEEE International Instrumentation and Measurement Technology Conference Victoria, Vancouver Island, Canada, May 12-15, 2008
- [44] IEEE 1451.1-1999, IEEE Standard for a Smart Transducer Interface for Sensors and Actuators—Network Capable Application Processor (NCAP) Information Model, Jun. 1999
- [45] E.Y. Song and K.B. Lee, STWS: A Unified Web Service for IEEE 1451 Smart Transducers, IEEE Transactions on Instrumentation and Measurement, vol. 57, no. 8, Aug. 2008, pp. 1749-1756
- [46] E. Song and K. Lee, Smart Transducer Web Services Based on the IEEE 1451.0 Standard, Instrumentation and Measurement Technology Conference - IMTC 2007 Warsaw, Poland, May 1-3, 2007
- [47] E. Song and K. Lee, An Implementation of Smart Transducer Web Services for IEEE 1451-based Sensor Systems, SAS 2007 - IEEE Sensors and Applications Symposium San Diego, California, USA, Feb. 6-8, 2007
- [48] E. Song and K. Lee, Integration of IEEE 1451 Smart Transducers and OGC-SWE Using STWS, SAS 2009 - IEEE Sensors Applications Symposium New Orleans, LA, USA, Feb. 17-19, 2009
- [49] Sun SPOT Manager; <http://www.sunspotworld.com/SPOTManager/> [Προσπέραστηκε 29/9/2010]
- [50] How to install The Java Communications API in a Windows Environment; <http://circuitnegma.wordpress.com/2007/02/07/how-to-install-the-java-communications-api-in-a-windows-environment/> [Προσπέραστηκε 29/9/2010]
- [51] Java USB API for Windows - User Installation; <http://www.steelbrothers.ch/jusb/api/usb/windows/related-docs/chp7-enduserinstallation.pdf> [Προσπέραστηκε 29/9/2010]
- [52] Apache Tomcat; <http://tomcat.apache.org/> [Προσπέραστηκε 29/9/2010]
- [53] Sensor Web Enablement; http://geostandards.geonovum.nl/index.php/5_Sensor_Web_Enablement [Προσπέραστηκε 29/9/2010]
- [54] S. Jirka, A. Bröring and C. Stasch, Discovery Mechanisms for the Sensor Web, Westfälische Wilhelms-Universität, Institute for Geoinformatics, Münster, Germany, Sensors 2009, Apr. 16
- [55] P. Hu, R. Robinson and J. Indulka, Sensor Standards: Overview and Experiences, University of Queensland, Australia
- [56] M. Botts, G. Percivall, C. Reed and J. Davidson, OGC Sensor Web Enablement: Overview and High Level Architecture. 2008
- [57] TinyOS Home Page; <http://www.tinyos.net/> [Προσπέραστηκε 3/10/2010]
- [58] The Contiki Operating System; <http://www.sics.se/contiki/> [Προσπέραστηκε 3/10/2010]

[59] SunSPOTWorld; <http://www.sunspotworld.com/> [Προσπέραστηκε 3/10/2010]