ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΕΠΙΚΟΙΝΩΝΙΩΝ, ΗΛΕΚΤΡΟΝΙΚΗΣ ΚΑΙ
ΣΥΣΤΗΜΑΤΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ

Πιλοτική Εφαρμογή Βαθμονόμησης Συστήματος Δορυφορικών Επικοινωνιών

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΗΛΙΑΣ-ΣΤΥΛΙΑΝΟΣ Ζ. ΚΑΡΑΜΠΑΣΗΣ

Επιβλέπων : Θεοδώρα Βαρβαρίγου
Καθηγήτρια Ε.Μ.Π.

Αθήνα, Μάιος 2018

ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΕΠΙΚΟΙΝΩΝΙΩΝ, ΗΛΕΚΤΡΟΝΙΚΗΣ ΚΑΙ
ΣΥΣΤΗΜΑΤΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ

Πιλοτική Εφαρμογή Βαθμονόμησης Συστήματος Δορυφορικών Επικοινωνιών

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΗΛΙΑΣ-ΣΤΥΛΙΑΝΟΣ Ζ. ΚΑΡΑΜΠΑΣΗΣ

Επιβλέπων : Θεοδώρα Βαρβαρίγου
Καθηγήτρια Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 11η Μαΐου 2018.

....	....	....
Θ. Βαρβαρίγου Καθηγήτρια Ε.Μ.Π.	Δ. Ασκούνης Καθηγητής Ε.Μ.Π.	Σ. Παπαβασιλείου Καθηγητής Ε.Μ.Π.

Αθήνα, Μάιος 2018

....

Ηλίας-Στυλιανός Ζ. Καραμπάσης

Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

Περίληψη

Σκοπός αυτής της διπλωματικής εργασίας ήταν η δημιουργία ενός συστήματος το οποίο χρησιμοποιείται στη βαθμονόμηση ενός δορυφορικού transponder. Ο transponder αυτός χρησιμοποιείται στη διαδικασία βαθμονόμησης του δορυφόρου Sentinel-3. Γίνεται εύκολα αντιληπτό ότι το αντικείμενο αυτής της εργασίας συνδέεται τόσο με τον τομέα του cloud αλλά και των σύγχρονων επεκτάσεών του, του fog και του edge computing. Όλα αυτά αναλύονται και συγκρίνονται διεξοδικά στο πρώτο κεφάλαιο. Επίσης, λόγω της φύσης του συστήματος θα ήταν δύσκολο να μην συνδεθεί η εργασία αυτή με τον ραγδαίως αναπτυσσόμενο κλάδο του Internet of Things. Στο δεύτερο κεφάλαιο λοιπόν της διπλωματικής αυτής αναφερόμαστε στο Internet of Things, τις σημαντικές επιδράσεις του cloud και του edge computing σε αυτό και συγκρίνουμε διάφορες γνωστές πλατφόρμες IoT. Στο τρίτο κεφάλαιο περιγράφουμε αναλυτικά τις διάφορες συνιστώσες του συστήματος που υλοποιήθηκε, αναφέροντας και τις τεχνολογίες που χρησιμοποιήθηκαν όπως και ποια από τις προαναφερθέντες πλατφόρμες IoT τελικά επιλέχθηκε. Στη συνέχεια στο τέταρτο κεφάλαιο αναλύουμε την υλοποίηση του συστήματος. Περιγράφονται αναλυτικά η διαδικασία λήψης των μετρήσεων από την έξοδο ενός μετατροπέα ADC, ο υπολογισμός της μέσης τιμής αυτών σε κάθε συχνότητα και της καθυστέρησης ομάδος. Στόχος μας ήταν να δημιουργηθεί ένα σύστημα που θα χρησιμοποιεί σύγχρονες τεχνολογίες και θα είναι εύκολα επεκτάσιμο. Τέλος, στο πέμπτο κεφάλαιο παρουσιάσουμε τα τελικά αποτελέσματα των προσομοιώσεων και αναφέρουμε ενδεικτικά κάποιες μελλοντικές επεκτάσεις του συστήματος.

Λέξεις κλειδιά: Cloud Computing, Fog Computing, Edge Computing, Internet of Things, δορυφορικός transponder, Node-RED, Beaglebone Black, μετατροπέας ADC, καθυστέρησης ομάδος

Abstract

The purpose of this diploma thesis is the creation of a system that is used in the calibration process of a satellite transponder. This transponder is used in the calibration process of Sentinel-3 satellite. It can be easily seen that the subject of this thesis is connected with the scientific field of cloud computing and its contemporary extensions, the fog and edge computing. All these are analyzed and compared in detail in the first chapter. Also, due to the nature of the system, it would be difficult not to link this thesis to the rapidly growing field of the Internet of Things. In the second chapter of this thesis, we refer to the Internet of Things, the important effects of cloud and edge computing on it and we compare several well-known IoT platforms. In the third chapter we describe in detail the various components of the system implemented, mentioning the technologies used as well as which of the aforementioned IoT platforms was finally selected. Then in the fourth chapter we analyze the implementation of the system. We describe in detail the process of taking measurements from the output of an ADC converter, calculating their mean value at each frequency and the group delay. Our goal was to create a system that uses modern technologies and will be easily expandable. Finally, in the fifth chapter we present the final results of the simulations and mention some future extensions of the system.

Keywords: Cloud Computing, Fog Computing, Edge Computing, Internet of Things, Satellite transponder, Node-RED, Beaglebone Black, ADC converter, group delay

<u>Ευχαριστίες</u>

Η παρούσα διπλωματική εργασία εκπονήθηκε στο Εργαστήριο Distributed Knowledge and Media Systems Group του Τομέα Επικοινωνιών, Ηλεκτρονικής και Συστημάτων Πληροφορικής της σχολής Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών του Εθνικού Μετσόβιου Πολυτεχνείου, υπό την επίβλεψη της Καθηγήτριας Θεοδώρας Βαρβαρίγου.

Καταρχάς θα ήθελα να ευχαριστήσω την κα. Θεοδώρα Βαρβαρίγου που μου ανάθεσε και μου επέτρεψε να ασχοληθω με ένα σύγχρονο και ενδιαφέρον θέμα, το οποίο με βοήθησε να εμβαθύνω και να επεκτείνω τις γνώσεις μου.

Επίσης, θα ήθελα να ευχαριστήσω ιδιαιτέρως τους υποψήφιους Διδάκτορες του Ε.Μ.Π., Βρεττό Μουλό και Αχιλλέα Μαρινάκη, για το χρόνο που μου αφιέρωσαν, τη βοήθεια, τις οδηγίες και κατευθύνσεις που μου έδιναν σε όλα τα στάδια εκπόνησης της εργασίας.

Ακόμα, θα ήθελα να ευχαριστήσω τον κ. Βασίλη Κασούρα και τον Δημήτρη Παπαγεωργίου για τη συνεργασία που είχαμε στο πλαίσιο αυτής της διπλωματικής εργασίας.

Τέλος, ένα μεγάλο ευχαριστώ οφείλω σε όλους τους φίλους-συμφοιτητές μου για τα πανέμορφα φοιτητικά χρόνια που περάσαμε μαζί και πάνω απ’ όλα στην οικογένειά μου για την υπομονή και την στήριξή τους στη μέχρι τώρα πορεία μου.

Ηλίας Καραμπάσης
Αθήνα, Ιούνιος 2018

Πίνακας Περιεχομένων

Κεφάλαιο 1: Περιγραφή και σύγκριση των Cloud, Fog και Edge Computing.............12
1.1 Ορισμός και περιγραφή του Cloud Computing……………….........................12
1.2 Πλεονεκτήματα και μειονεκτήματα του Cloud Computing……………………16
1.3 Ορισμός και περιγραφή του Fog Computing………………………………...19
1.4 Σύγκριση μεταξύ Cloud και Fog Computing…………………………………..23
1.5 Ορισμός και περιγραφή του Edge Computing…………………………………26
1.6 Πλεονεκτήματα και προκλήσεις του Edge Computing…………………………28
1.7 Σύγκριση μεταξύ Cloud, Fog και Edge Computing…………………………….30
Κεφάλαιο 2: Internet of Things..33
2.1 Εφαρμογές του Internet of Things (IoT)..33
2.2 Η ενοποίηση της IoT αρχιτεκτονικής με το Cloud και το Edge.........................36
2.3 Πλατφόρμες IoT..39
2.3.1 Eclipse Kura (+Apache Camel)..40
2.3.2 Kaa…………………………………………………………………42
2.3.3 Flogo……………………………………………………………......46
2.3.4 WoTKit Processor……………………………………………………48
2.3.5 Node-RED……………………………………………………………50
2.4 Επιλογή πλατφόρμας IoT………………………………………………………53
Κεφάλαιο 3: Περιγραφή του συστήματος της διπλωματικής..56
3.1 Ανάλυση συστήματος...56
3.2 Δομικά στοιχεία συστήματος..57
3.2.1 Ο δορυφόρος SENTINEL-3………………………………………..57
3.2.2 Υποσύστημα ελέγχου σημάτων……………………………...........60
3.2.3 Synthesizer…………………………………………………………65
3.2.4 Πλακέτα ελέγχου...69
3.3 Πλατφόρμα πληροφοριακού συστήματος..70
Κεφάλαιο 4: Περιγραφή της υλοποίησης..71
4.1 Διαδικτυακή πλατφόρμα πληροφοριακού συστήματος......................................71
4.2 Λήψη μετρήσεων και εκτέλεση υπολογισμών στο Node-Red............................76
Κεφάλαιο 5: Τελικά αποτελέσματα & μελλοντικές επεκτάσεις....................................85
5.1 Τελικά αποτελέσματα...85
5.2 Μελλοντικές επεκτάσεις...86
Κεφάλαιο 6: Βιβλιογραφία..87
Παράρτημα...89

Περιεχόμενα Πινάκων

Πίνακας 1: Σύγκριση χαρακτηριστικών των Cloud και Fog Computing............................24
Πίνακας 2: Σύγκριση διαφόρων χαρακτηριστικών των cloud, fog και edge computing......30
Πίνακας 3: Σύγκριση των cloud, fog και edge computing με βάση υπολογιστικές και υπηρεσίες αποθήκευσης...31
Πίνακας 4: Συγκριτικός πίνακας μεταξύ των πλατφορμών IoT που αναφέρθηκαν στην ενότητα 2.3..55
Πίνακας 5: Περιγραφή των λειτουργιών των 9 pin της υποδοχής P1.................................66

Περιεχόμενα Εικόνων

Εικόνα 1: Εποπτική παρουσίαση ορισμένων εκ των υπηρεσιών που μπορεί να προσφέρει το cloud..12
Εικόνα 2: Τα μοντέλα υπηρεσιών του Cloud Computing..14
Εικόνα 3: Τα μοντέλα ανάπτυξης ενός συστήματος cloud...15
Εικόνα 4: Το fog computing υποστηρίζει ένα cloud-based οικοσύστημα για έξυπνες τελικές συσκευές...20
Εικόνα 5: Εποπτική εικόνα του Edge Computing...27
Εικόνα 6: Διάφοροι τομείς στους οποίους το Internet of Things βρίσκει εφαρμογή............33
Εικόνα 7: Εποπτική παρουσίαση των συστατικών μερών του Kura..................................41
Εικόνα 8: Ο server και τα SDKs του Kaa..43
Εικόνα 9: Το αποτύπωμα του Flogo και των edge εφαρμογών του συγκριτικά με αντίστοιχες πλατφόρμες..46
Εικόνα 10: Παράδειγμα ενός flow από το Web UI του Flogo για την ενεργοποίηση μιας συσκευής..47
Εικόνα 11: Ο βασισμένος σε browser editor ροής δεδομένων του WoTKit Processor........49
Εικόνα 12: Η online σελίδα flows.nodered.org όπου χρήστες μπορούν να αναζητήσουν έτοιμα flows ή κόμβους αλλά και να ανεβάσουν τα δικά τους..51
Εικόνα 13: Ο web-based flow editor του Node-RED..53
Εικόνα 14: Σχηματική αναπαράσταση του συστήματος το οποίο θα τοποθετηθεί στον Ψηλορείτη της Κρήτης...56
Εικόνα 15: Ο δορυφόρος Sentinel-3 και τα διάφορα όργανά του.......................................58
Εικόνα 16: Το όργανο SRAL..59
Εικόνα 17: Μεταδιδόμενο και λαμβανόμενο σήμα...61
Εικόνα 18: Οι όροι Δφ και Δf για τον υπολογισμό της καθυστέρησης ομάδας..................62
Εικόνα 19: Η τοπολογία του ειδικού εξοπλισμού (Built-In Test Equipment - BITE) του συστήματος...62
Εικόνα 20: Σχηματική αναπαράσταση του synthesizer SLSM5...67
Εικόνα 21: Η υπολογιστική πλακέτα Beaglebone Black..69
Εικόνα 22: Το ειδικό Beaglebone Cape που χρησιμοποιήθηκε..69
Εικόνα 23: Η σελίδα εγγραφής νέου χρήστη register.php..71
Εικόνα 24: Η σελίδα εισόδου login.php..72
Εικόνα 25: Η σελίδα εκκίνησης πειραμάτων και απεικόνισης της εξέλιξής τους...............72

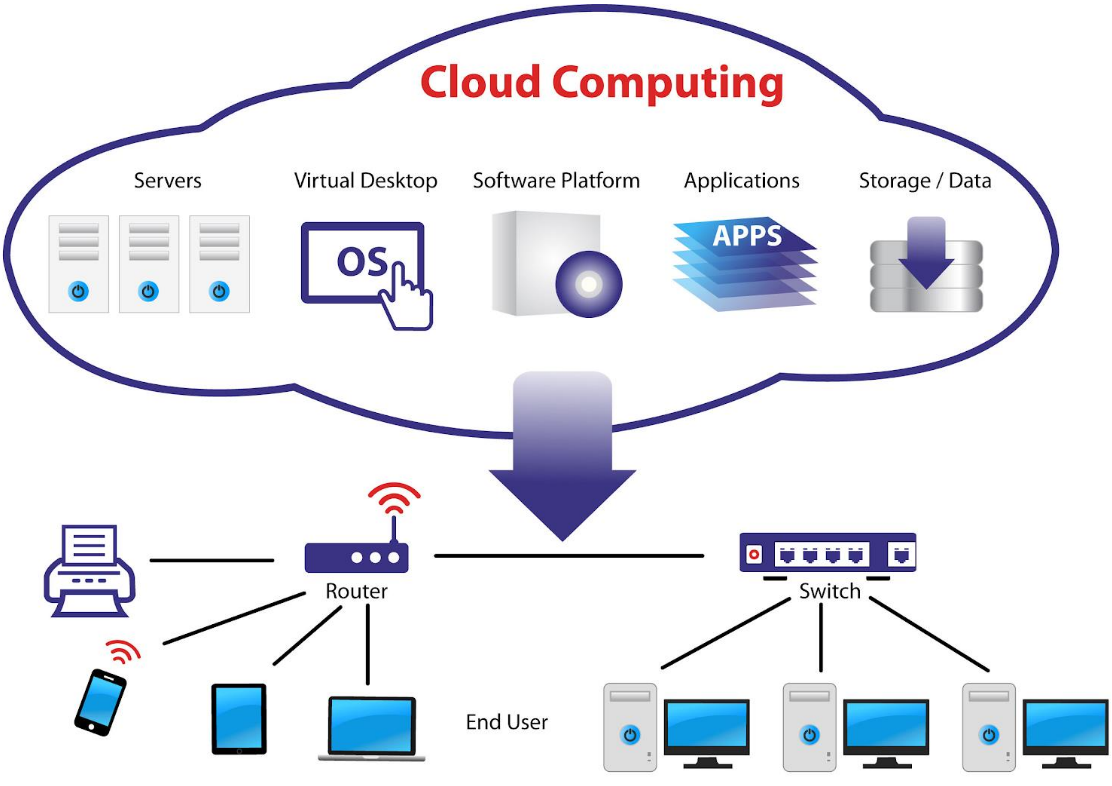
Εικόνα 26: Απεικόνιση της εξέλιξης της εκτέλεσης ενός πειράματος.................................73
Εικόνα 27: Ολοκλήρωση της εκτέλεσης του προηγούμενου πειράματος............................73
Εικόνα 28: Απεικόνιση των leds στην πρόσοψη του εξοπλισμού.......................................74
Εικόνα 29: Η γραφική παράσταση της μέσης τιμής των τάσεων εξόδου του transponder για το προηγούμενο πείραμα...74
Εικόνα 30: Η γραφική παράσταση της καθυστέρησης ομάδος στην έξοδο του transponder για το προηγούμενο πείραμα...75
Εικόνα 31: Η σελίδα απεικόνισης των διαγραμμάτων του τελευταίου πειράματος που εκτελέστηκε...75
Εικόνα 32: Απεικόνιση του websocket εισόδου και εξόδου...76
Εικόνα 33: Αίτηση και απόκριση για την κατάσταση του flow...77
Εικόνα 34: Αίτηση και απόκριση για την κατάσταση των leds...77
Εικόνα 35: Ο κόμβος FSM που δημιουργήθηκε...77
Εικόνα 36: Εποπτική απεικόνιση του συνόλου των κόμβων που αποτελούν το FSM.........78
Εικόνα 37: State 1 – Αρχικοποίηση...78
Εικόνα 38: Μετάβαση σε κατάσταση μετρήσεων...79
Εικόνα 39: Διαδικασία ενεργοποίησης εξοπλισμού για λήψη μετρήσεων και αποστολή στο synthesizer της εντολής συχνότητας...79
Εικόνα 40: Οι κόμβοι που αφορούν την επικοινωνία με το synthesizer..............................80
Εικόνα 41: Το σύνολο των κόμβων που αφορούν την λήψη των μετρήσεων, τη λήψη των απαντήσεων του synthesizer, την απενεργοποίηση των διακοπτών στο τέλος της διαδικασίας, κτλ...81
Εικόνα 42: Οι κόμβοι που αφορούν τον υπολογισμό του μέσου όρου της τάσης................82
Εικόνα 43: Οι κόμβοι που αφορούν τον υπολογισμό του group delay................................83
Εικόνα 44: Οι κόμβοι που αφορούν τις υπόλοιπες καταστάσεις του FSM..........................84
Εικόνα 45: Οι κόμβοι που αφορούν το σημείο αναμονής των flows μέχρι να ολοκληρωθεί η διαδικασία των μετρήσεων...84
Εικόνα 46: Απεικόνιση της μέσης τιμής της τάσης της εξόδου του μετατροπέα ADC σε όλο το πιθανό εύρος συχνοτήτων του πειράματος...85
Εικόνα 47: Απεικόνιση του group delay της τάσης της εξόδου του μετατροπέα ADC σε όλο το πιθανό εύρος συχνοτήτων του πειράματος...86

<u>Κεφάλαιο 1ο</u>
Περιγραφή και σύγκριση των Cloud, Fog και Edge Computing

Στο κεφάλαιο αυτό θα ορίσουμε και θα περιγράψουμε τις έννοιες των cloud, fog και edge computing. Θα αναφερθούμε σε περιπτώσεις χρήσης τους και θα μπούμε στη διαδικασία να συγκρίνουμε αυτά τα τρία αρχιτεκτονικά μοντέλα, αναφέροντας και τυχόν περιορισμούς που μπορεί να επιβάλουν.

1.1 Ορισμός και περιγραφή του Cloud Computing

Το Cloud Computing είναι μια έννοια η οποία ήρθε στο προσκήνιο και άρχισε να απασχολεί τον κλάδο της πληροφορικής κυρίως μέσα στην τελευταία δωδεκαετία. Με βάση τον ορισμό που χρησιμοποιείται από το NIST (National Institute of Standards and Technology) [1], όταν αναφερόμαστε στο cloud computing μιλάμε ουσιαστικά για ένα μοντέλο που επιτρέπει πανταχού παρούσα, ευέλικτη και κατ' απαίτηση δικτυακή πρόσβαση σε έναν κοινόχρηστο και παραμετροποιήσιμο σύνολο υπολογιστικών πόρων, όπως δίκτυα, εξυπηρετητές, εφαρμογές και υπηρεσίες, οι οποίοι παρέχονται και απελευθερώνονται ταχύτατα, με ελάχιστο κόστος διαχείρισης και ελάχιστη αλληλεπίδραση με τον πάροχο της υπηρεσίας. Ουσιαστικά, το cloud computing είναι η παροχή υπολογιστικών υπηρεσιών μέσω του διαδικτύου.



Εικόνα 1: Εποπτική παρουσίαση ορισμένων εκ των υπηρεσιών που μπορεί να προσφέρει το cloud

Έχοντας δώσει έναν ορισμό για το μοντέλο του cloud computing, μπορούμε να περιγράψουμε εν συντομία τα κυριότερα χαρακτηριστικά του, τα οποία το έχουν καταστήσει τόσο δημοφιλές. Με βάση και πάλι το NIST, έχουμε τα εξής [1]:

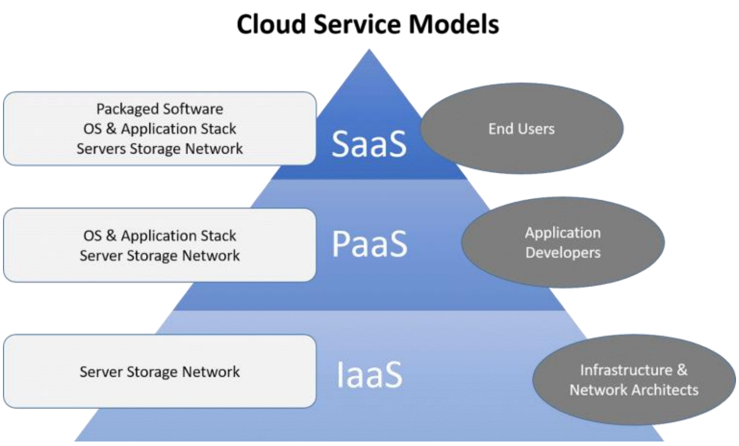
- **Κατ' απαίτηση υπολογιστικό μοντέλο:** Ο χρήστης μπορεί μόνος του να επιλέξει τους πόρους τους οποίους χρειάζεται και να τους προσαρμόζει ανάλογα με τις απαιτήσεις που έχει κάθε φόρα όντας τελείως αποδεσμευμένος από τον πάροχο των υπηρεσιών.
- **Ευρεία δικτυακή πρόσβαση:** Όλες οι υπηρεσίες είναι διαθέσιμες μέσω του δικτύου και είναι προσβάσιμες από τυποποιημένους μηχανισμούς που προωθούν την χρήση τόσο ελαφριών όσο και πιο βαριών πλατφορμών (π.χ. κινητά τηλέφωνα, ηλεκτρονικούς υπολογιστές, σταθμούς εργασίας κ.λ.π.).
- **Συγκέντρωση πόρων:** Όλοι οι πόροι του παρόχου είναι συγκεντρωμένοι ώστε να εξυπηρετούν πολλαπλούς χρήστες, με διαφορετικά φυσικά και εικονικά μέσα να ανατίθενται και να ανακατανέμονται δυναμικά, ανάλογα με τις ανάγκες του χρήστη. Υπάρχει μια αίσθηση ανεξαρτησίας από την τοποθεσία με την έννοια ότι ο πελάτης γενικώς δεν έχει κανένα έλεγχο ή γνώση σχετικά με την ακριβή τοποθεσία των παρεχόμενων υπηρεσιών, αλλά μπορεί να είναι σε θέση να καθορίσει την τοποθεσία σε ένα υψηλότερο επίπεδο αφαίρεσης (πχ τη χώρα, την πολιτεία ή το datacenter). Παραδείγματα πόρων περιλαμβάνουν τον αποθηκευτικό χώρο, τις υπολογιστικές δυνατότητες, την μνήμη και το εύρος της δικτυακής σύνδεσης.
- **Ελαστικότητα και Ευχρηστία:** Ίσως από τα σημαντικότερα χαρακτηριστικά του cloud computing. Οι πόροι και οι υπηρεσίες μπορούν άμεσα να δεσμεύονται και να αποδεσμεύονται, ακόμα και αυτόματα σε μερικές περιπτώσεις, ανάλογα με την εκάστοτε ζήτηση. Στον χρήστη οι πόροι και οι υπηρεσίες αυτές συχνά εμφανίζονται ότι μπορούν να διατεθούν σε οποιαδήποτε ποσότητα την οποιαδήποτε στιγμή χρειαστεί.
- **Μετρούμενη υπηρεσία:** Τα cloud συστήματα ελέγχουν και βελτιστοποιούν αυτόματα την χρήση πόρων αξιοποιώντας κάποια μετρική σε κάποιο επίπεδο αφαίρεσης που να ταιριάζει στον τύπο της υπηρεσίας. Η χρήση των πόρων μπορεί να παρακολουθηθεί, να ελεγχθεί και να αναφερθεί, παρέχοντας διαφάνεια τόσο στον πάροχο όσο και στον καταναλωτή της χρησιμοποιούμενης υπηρεσίας.

Ανάλογα με τον συνδυασμό υπηρεσιών που προσφέρουν, τα συστήματα cloud χωρίζονται σε διαφορετικά μοντέλα υπηρεσιών. Υπάρχουν τρία μοντέλα υπηρεσιών τα οποία είναι καθολικά αποδεκτά [2]:

- **Λογισμικό ως υπηρεσία (Software as a Service - SaaS):** Πρόκειται για ένα πλήρως λειτουργικό περιβάλλον με εφαρμογές, διαχείριση και διεπιφάνεια χρήστη. Στο μοντέλο SaaS, η εφαρμογή παρέχεται στον πελάτη μέσω μιας διεπιφάνειας (συνήθως μέσω ενός προγράμματος περιήγησης - browser) και η ευθύνη του πελάτη αρχίζει και τελειώνει με την είσοδο του σε αυτήν την διεπιφάνεια και την διαχείριση των δεδομένων του, καθώς και με οποιαδήποτε άλλη αλληλεπίδραση χρήστη. Για οτιδήποτε από την εφαρμογή μέχρι την υποδομή, είναι υπεύθυνος ο πάροχος. Από την πλευρά του πελάτη δεν χρειάζεται κάποια επένδυση σε υλικό ή κάποια ιδιαίτερη τεχνογνωσία. Παραδείγματα παρόχων SaaS: Abiquo, Accelops, Akamai, AppDynamics, MeghaWare, CloudSwitch, Cumulux, Eloqua, Oracle On Demand, Marketo, SAP BusinessByDesign, etc.

- **Πλατφόρμα ως υπηρεσία (Platform as a Service - PaaS):** Εδώ μια ολόκληρη πλατφόρμα παρέχεται ως υπηρεσία, πάνω στην οποία μπορεί να χτιστεί κάποια άλλη εφαρμογή. Παρέχονται εικονικές μηχανές, λειτουργικά συστήματα, εφαρμογές, υπηρεσίες, πλαίσια ανάπτυξης και δομές ελέγχου. Ο πελάτης μπορεί να αναπτύξει τις εφαρμογές του στο Cloud ή να χρησιμοποιήσει εφαρμογές που προγραμματίστηκαν χρησιμοποιώντας γλώσσες και εργαλεία που υποστηρίζονται από τον πάροχο υπηρεσιών PaaS. Ο πάροχος των υπηρεσιών διαχειρίζεται την υποδομή νέφους, τα λειτουργικά συστήματα και το απαιτούμενο λογισμικό. Ο πελάτης είναι υπεύθυνος για την εγκατάσταση και την διαχείριση της εφαρμογής που αναπτύσσει. Παραδείγματα παρόχων PaaS: Amazon Web Services, AppScale, Salesforce.com, Google, Windows Azure, GridGain, Openstack, ThinkGrid, κλπ.
- **Υποδομή ως υπηρεσία (Infrastructure as a Service - IaaS):** Παρέχονται εικονικές μηχανές, εικονικός αποθηκευτικός χώρος, εξυπηρετητές, εικονική υποδομή και άλλα υλικά στοιχεία ως πόροι που μπορούν οι πελάτες να προμηθευτούν. Ο πάροχος υπηρεσιών IaaS χειρίζεται όλη την υποδομή, ενώ ο πελάτης είναι υπεύθυνος για όλες τις πτυχές της ανάπτυξης. Αυτές μπορεί να περιλαμβάνουν το λειτουργικό σύστημα, εφαρμογές και αλληλεπιδράσεις χρηστών με το σύστημα. Παραδείγματα παρόχων IaaS : Amazon Elastic Compute Cloud (EC2), AT&T, bluelock, ca technologies, cloud scaling, DataPipe, Eucalyptus Systems, GoGrid, HP, logicworks, NaviSite, OpSource, RackSpace Cloud, Terremark, Verizon, etc.

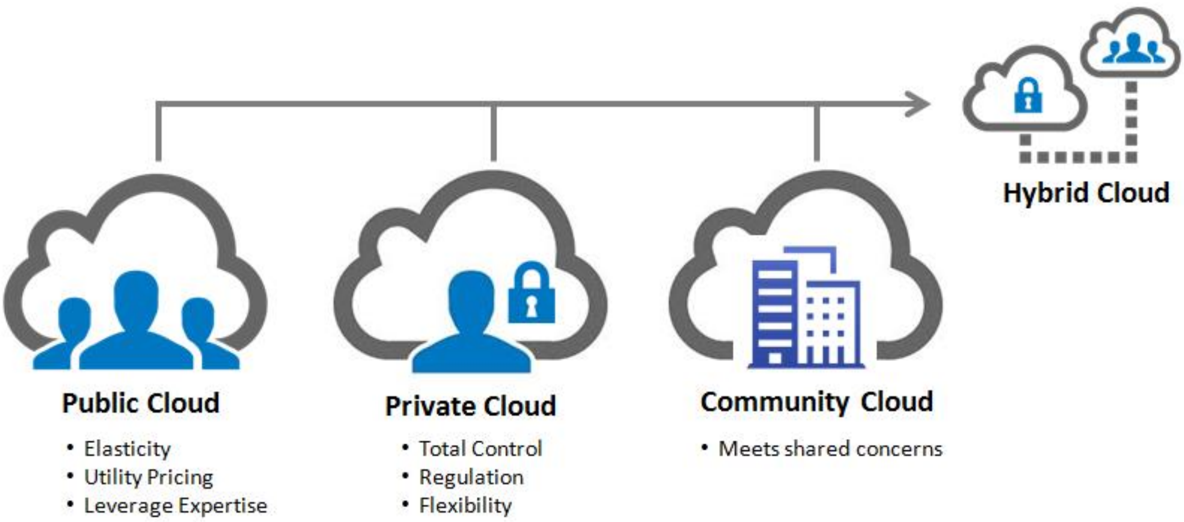
Τα τρία διαφορετικά μοντέλα υπηρεσιών σαν σύνολο, είναι γνωστά ως το SPI μοντέλο του Cloud. Έχουν αναφερθεί πολλά άλλα μοντέλα υπηρεσιών, όπως Αποθηκευτικός χώρος ως υπηρεσία (Storage as a Service - StaaS), Ταυτότητα ως υπηρεσία (Identity as a Service - IdaaS) κ.α. Παρόλα αυτά, οι υπηρεσίες SPI υπερκαλύπτουν όλες τις άλλες πιθανότητες.



Εικόνα 2: Τα μοντέλα υπηρεσιών του Cloud Computing

Υπάρχουν διάφορα μοντέλα ανάπτυξης ενός cloud συστήματος και το εκάστοτε μοντέλο καθορίζεται με βάση την ιδιοκτησία, το μέγεθος και την προσβασιμότητα αυτού. Έτσι έχουμε τα παρακάτω τέσσερα μοντέλα [1]:

- **Ιδιωτικό cloud (Private cloud):** Η υποδομή του cloud παρέχεται για αποκλειστική χρήση από έναν μοναδικό οργανισμό που αποτελείται από πολλαπλούς χρήστες (για παράδειγμα μονάδες επιχείρησης). Στοχεύουν να κυρίως στο να λύσουν το θέμα της ασφάλειας και προσφέρουν μεγαλύτερο έλεγχο. Μπορεί να κατέχει, να διαχειρίζεται και να λειτουργεί το υπολογιστικό νέφος, ένας οργανισμός, ένα τρίτο πρόσωπο ή ένας συνδυασμός αυτών, ενώ μπορεί να βρίσκεται εντός ή εκτός των εγκαταστάσεων τους.
    - *On-premise Private Cloud:* φιλοξενείται μέσα στο ίδιο το κέντρο δεδομένων του πελάτη. Το μοντέλο αυτό παρέχει πιο τυποποιημένη μορφή διαχείρισης και προστασίας αλλά είναι περιορισμένο όσον αναφορά την επέκταση. Επιπλέον πρέπει να διατίθεται κεφάλαιο για την συντήρηση των υλικών πόρων. Βρίσκει καλύτερη εφαρμογή σε περιπτώσεις όπου χρειάζεται πλήρης έλεγχος στην παραμετροποίηση της δομής του νέφους και στην ασφάλεια.
    - *Externally hosted Private Cloud:* φιλοξενείται εξωτερικά του δικτύου του πελάτη μέσω ενός παρόχου νέφους, ο οποίος εγγυάται αποκλειστικότητα στο νέφος που παρέχει και ασφάλεια. Είναι ιδανικό για πελάτες που δεν θέλουν ή δεν μπορούν να διαθέσουν κεφάλαιο και χώρο για τη συντήρηση των υλικοτεχνικών δομών ενός cloud.
- **Δημόσιο cloud (Public cloud):** Η υποδομή του cloud παρέχεται για δημόσια χρήση. Μπορεί να κατέχει, να διαχειρίζεται και να λειτουργεί το cloud μία επιχείρηση, ένας ακαδημαϊκός ή κυβερνητικός οργανισμός ή ένας συνδυασμός αυτών. Βρίσκεται εντός των εγκαταστάσεων του παρόχου του cloud. Παρέχουν υπηρεσίες χαμηλού κόστους με μοντέλο Pay-As-You-Go. Όλοι οι πελάτες μοιράζονται τους ίδιους πόρους με περιορισμένες επιλογές προσαρμογής, περιορισμένη προστασία ασφαλείας και διαθεσιμότητα. Ένα βασικό πλεονέκτημα του public cloud είναι ότι είναι πολύ μεγαλύτερο από ένα ιδιωτικό cloud και μπορεί να προσφέρει πολύ περισσότερες δυνατότητες επέκτασης στους πελάτες του.



Εικόνα 3: Τα μοντέλα ανάπτυξης ενός συστήματος cloud

- **Cloud κοινότητας (Community cloud):** Η υποδομή του cloud παρέχεται για αποκλειστική χρήση από μία συγκεκριμένη κοινότητα χρηστών, από οργανισμούς που έχουν κοινά ενδιαφέροντα (για παράδειγμα κοινή αποστολή, πολιτική απαιτήσεων ασφαλείας και ενδιαφέροντα συμβατότητας). Μπορεί να κατέχει, να διαχειρίζεται και να λειτουργεί το Cloud, ένας ή περισσότεροι από τους οργανισμούς της κοινότητας, ένα τρίτο πρόσωπο ή ένας συνδυασμός αυτών, ενώ μπορεί να βρίσκεται εντός ή εκτός των εγκαταστάσεων τους.
- **Υβριδικό cloud (Hybrid cloud):** Η υποδομή του cloud είναι μια σύνθεση δύο ή περισσοτέρων ξεχωριστών υποδομών νέφους (ιδιωτικό, κοινότητας ή δημόσιο), που παραμένουν μοναδικές οντότητες, αλλά είναι δεμένες (δεσμευμένες) μαζί μέσω πρότυπης ή ιδιοκτησιακής τεχνολογίας που επιτρέπει την φορητότητα δεδομένων και εφαρμογών.

1.2 Πλεονεκτήματα και μειονεκτήματα του Cloud computing

Αρχικά, ας μιλήσουμε για μερικά από τα βασικότερα πλεονεκτήματα που προσφέρει το cloud computing [3]:

1. **Ευελιξία**
   Οι υπηρεσίες βασισμένες στο cloud είναι ιδανικές για επιχειρήσεις οι οποίες έχουν αναπτυσσόμενες ή κυμαινόμενες απαιτήσεις. Αν οι ανάγκες μιας επιχείρησης αυξάνονται, είναι εύκολο να αυξήσει την χωρητικότητα που δεσμεύει στο cloud στους απομακρυσμένους εξυπηρετητές του παρόχου των cloud υπηρεσιών. Αντιστοίχως, μπορεί με την ίδια ευελιξία να μειώσει τις υποδομές που δεσμεύει όταν δεν τις είναι πλέον χρήσιμες. Αυτού του είδους η "ευκολία" μπορεί να δώσει στις επιχειρήσεις που υιοθετούν το cloud computing ένα σημαντικό πλεονέκτημα έναντι των ανταγωνιστών.
2. **Αξιοπιστία**
   Η κλίμακα των δικτύων cloud και η δυνατότητά τους να παρέχουν εξισορρόπηση υπερβάλλοντος φορτίου που μπορεί να προκύψει και αντιμετώπιση αστοχίας του συστήματος, τα καθιστά αξιόπιστα. Υπάρχει γενικά μεγαλύτερη αξιοπιστία από την ύπαρξη τοπικών υλικοτεχνικών δομών, αφού ο πελάτης έχει πρόσβαση σε έναν τεράστιο αριθμό πόρων των οποίο μπορεί να αξιοποιήσει και σε περίπτωση αποτυχίας κάποιου εξυπηρετητή τα δεδομένα και οι υπηρεσίες του μπορούν άμεσα και με καθόλου κόστος να μεταφερθούν στον επόμενο.
3. **Ανάκτηση δεδομένων σε περίπτωση καταστροφής**
   Όλες οι επιχειρήσεις ανεξαρτήτως μεγέθους θα πρέπει να επενδύουν σε ένα ισχυρό σύστημα ανάκτησης δεδομένων σε περιπτώσεις καταστροφών. Με το cloud η κάλυψη αυτής της σημαντικής ανάγκης γίνεται ευκολότερη για περισσότερους οργανισμούς/επιχειρήσεις. Σύμφωνα μάλιστα με το Aberdeen Group, οι μικρές επιχειρήσεις είναι δύο φορές πιθανότερο από τις μεγαλύτερες να έχουν υλοποιήσει υπηρεσίες backup και ανάκτησης δεδομένων βασισμένες στο cloud, εξοικονομώντας έτσι χρόνο, αποφεύγοντας μεγάλες επενδύσεις εκ των προτέρων και στηριζόμενες στην τεχνογνωσία των μεγάλων παρόχων υπηρεσιών cloud.

4. **Εύκολη διαχείριση**
   Δεν χρειάζεται καμία τεχνογνωσία ή η εγκατάσταση επιπλέον λογισμικού ή υλικοτεχνικής υποδομής από πλευράς του πελάτη. Η διαχείριση των πόρων απλοποιείται μέσω κεντρικών διαχειριστικών μονάδων, ενώ η δέσμευση και αποδέσμευση πόρων γίνεται δυναμικά.
5. **Αυτόματες αναβαθμίσεις λογισμικού**
   Οι πάροχοι υπηρεσιών cloud έχουν υπό την εποπτεία τους τους servers που χρησιμοποιούν οι επιχειρήσεις-πελάτες τους και είναι υπεύθυνοι να εγκαθιστούν τακτικά όλες τις αναβαθμίσεις λογισμικού, συμπεριλαμβανομένων και των αναβαθμίσεων ασφαλείας φυσικά. Έτσι, μια επιχείρηση δεν χρειάζεται να "σπαταλά" χρόνο για τη συντήρηση των συστημάτων της, έχοντας έτσι περισσότερο χρόνο να ασχοληθεί με την ανάπτυξή της.
6. **Μηδενικό κεφαλαιουχικό κόστος**
   Το cloud computing εκμηδενίζει το υψηλό κόστος του υλικού εξοπλισμού (hardware), αφού η πληρωμή γίνεται με τη μορφή συνδρομής ή με το λεγόμενο μοντέλο pay-as-you-go, δηλαδή με πληρωμή με βάση τα τρέχοντα έσοδα της επιχείρησης. Έτσι, διευκολύνεται η ροή του κεφαλαίου μιας επιχείρησης. Επιπλέον, η εγκατάσταση και η διαχείριση των υποδομών στο cloud είναι πολύ ευκολότερη από το να διατηρούσε μια επιχείρηση τις αντίστοιχες υποδομές τοπικά.
7. **Αυξημένη/Βελτιωμένη συνεργασία**
   Όταν μια ομάδα μπορεί να έχει πρόσβαση, να επεξεργάζεται και να διαμοιράζει τα έγγραφά της οποτεδήποτε και απ' οπουδήποτε, τότε τα μέλη της μπορούν να κάνουν περισσότερα μαζί και να εκτελέσουν το έργο τους καλύτερα. Η ροή εργασιών βασισμένη στο cloud και οι εφαρμογές διαμοιρασμού αρχείων βοηθούν τα μέλη της ομάδας να κάνουν ανανεώσεις στο περιεχόμενο σε πραγματικό χρόνο και τους δίνουν πλήρη εποπτεία στις συνεργασίες τους.
8. **Εργασία από παντού**
   Το cloud computing παρέχει τη δυνατότητα να μπορεί να δουλέψει κανείς απ' οπουδήποτε, εφόσον έχει στη διάθεσή του μια σύνδεση στο internet. Επιπλέον, το γεγονός ότι οι περισσότερες σοβαρές υπηρεσίες cloud προσφέρουν και mobile εφαρμογές, δεν περιορίζει κάποιον σε ό,τι αφορά τη συσκευή που έχει στα χέρια του. Έτσι, οι επιχειρήσεις μπορούν να προσφέρουν στους εργαζομένους τους ευέλικτες συνθήκες εργασίας, έτσι ώστε εκείνοι να απολαμβάνουν την ισορροπία μεταξύ εργασίας και προσωπικής ζωής που επιθυμούν.
9. **Διαχείριση/Ελεγχος εγγράφων**
   Δίχως το cloud, τα μέλη μιας ομάδας έπρεπε να στέλνουν τα έγγραφά τους μπρος-πίσω μέσω email, για να τα επεξεργαστεί ένα άτομο τη φορά. Έτσι, αργά ή γρήγορα μπορεί να κατέληγαν σε έγγραφα με αλληλοσυγκρουόμενο περιεχόμενο, μορφοποίηση ή τίτλους. Και καθώς ακόμα και οι μικρότερες επιχειρήσεις γίνονται περισσότερο παγκόσμιες, το περιθώριο για να δημιουργηθεί κάποιο μπέρδεμα αυξάνει. Αντίθετα, με τη χρησιμοποίηση του cloud όλα τα αρχεία αποθηκεύονται κεντρικά και όλοι βλέπουν την ίδια έκδοση ταυτόχρονα. Αυτό συμβάλλει στην καλύτερη και ταχύτερη συνεργασία της ομάδας, σε ένα καλύτερο τελικό αποτέλεσμα και κατ' επέκταση σε μεγαλύτερο εταιρικό κέρδος.

10. **Περισσότερο φιλικό προς το περιβάλλον**
    Το cloud computing βοηθά μια επιχείρηση να λειτουργεί περισσότερο φιλικά ως προς το περιβάλλον. Καθώς οι ανάγκες μιας επιχείρησης αυξομειώνονται, μεταβάλλονται ανάλογα και οι υπολογιστικοί πόροι στο cloud τους οποίους χρησιμοποιεί. Έτσι, μια επιχείρηση καταναλώνει πάντα μόνο όση ενέργεια πραγματικά χρειάζεται, μειώνοντας έτσι το ενεργειακό της αποτύπωμα.

Φυσικά παρά τα πολλά πλεονεκτήματα που προσφέρει το cloud computing, δεν αποτελεί πανάκεια. Μερικά από τα σημαντικότερα μειονεκτήματά του αναφέρονται παρακάτω [2]:

1. **Προσωρινή παύση λειτουργίας**
   Αυτό ίσως να είναι ένα από τα χειρότερα μειονεκτήματα του cloud computing. Κανένας πάροχος υπηρεσιών cloud, ακόμα και οι καλύτεροι, δεν μπορούν να εγγυηθούν ότι δεν θα υπάρξει ποτέ προσωρινή διακοπή λειτουργίας της υπηρεσίας τους. Επίσης, τα συστήματα cloud βασίζονται στο Internet και άρα η πρόσβαση σε αυτά εξαρτάται πλήρως από τη δυνατότητα σύνδεσης στο Internet. Και φυσικά, όπως και κάθε λογής hardware, οι πλατφόρμες cloud μπορεί να αποτύχουν για οποιονδήποτε λόγο παρά τη συχνή συντήρησή τους από τους παρόχους.
2. **Ασφάλεια και ιδιωτικότητα**
   Η ασφάλεια είναι το μεγαλύτερο θέμα ανησυχίας όταν πρόκειται για το cloud. Με το να συνεργάζεται με εξωτερικούς φορείς ο πελάτης δίνει ευαίσθητα δεδομένα είτε προς επεξεργασία είτε προς φύλαξη στο νέφος. Δεδομένα τα οποία πλέον είναι υπεύθυνος ο πάροχος να διαχειριστεί και να προφυλάξει.
3. **Ζητήματα νομικής φύσεως**
   Λόγω της κατανεμημένης φύσης των συστημάτων cloud, είναι πρακτικά δύσκολος ο εντοπισμός της φυσικής τοποθεσίας, στην οποία βρίσκονται τα δεδομένα ενός χρήστη ή όπου γίνεται η εκτέλεση των εφαρμογών του. Αναλόγως με την χώρα στην οποία βρίσκεται η φυσική υποδομή ενός cloud, μπορεί να ισχύει και διαφορετική νομοθεσία σε ό,τι αφορά την ιδιωτικότητα των δεδομένων. Αυτό έχει ως αποτέλεσμα, να μην γνωρίζει τελικά ο χρήστης, σε ποια νομοθεσία υπόκεινται τα δεδομένα και οι δραστηριότητές του και τελικά να υφίσταται παραβίαση της ιδιωτικότητάς του, χωρίς να το γνωρίζει.
4. **Δύσκολη αλλαγή παρόχου**
   Η αλλαγή παρόχου είναι ιδιαίτερα δύσκολη κυρίως λόγω της δυσκολίας της μεταφοράς μεγάλου όγκου δεδομένων από τον ένα πάροχο στον άλλο. Οπότε ουσιαστικά ο πελάτης δεσμεύεται για συνεργασία μόνο με έναν πάροχο.
5. **Μειωμένος έλεγχος**
   Αφού όλη η υλικοτεχνική δομή του νέφους και η διαχείρισή του γίνεται από τον πάροχο ο έλεγχος του πελάτη περιορίζεται μόνο στις εφαρμογές που τρέχουν πάνω από αυτό το επίπεδο. Οπότε οποιαδήποτε κύρια διαχειριστική ενέργεια, όπως πρόσβαση στον πυρήνα του εξυπηρετητή, αναβάθμιση, διαχείριση του firmware δεν επιτρέπονται.

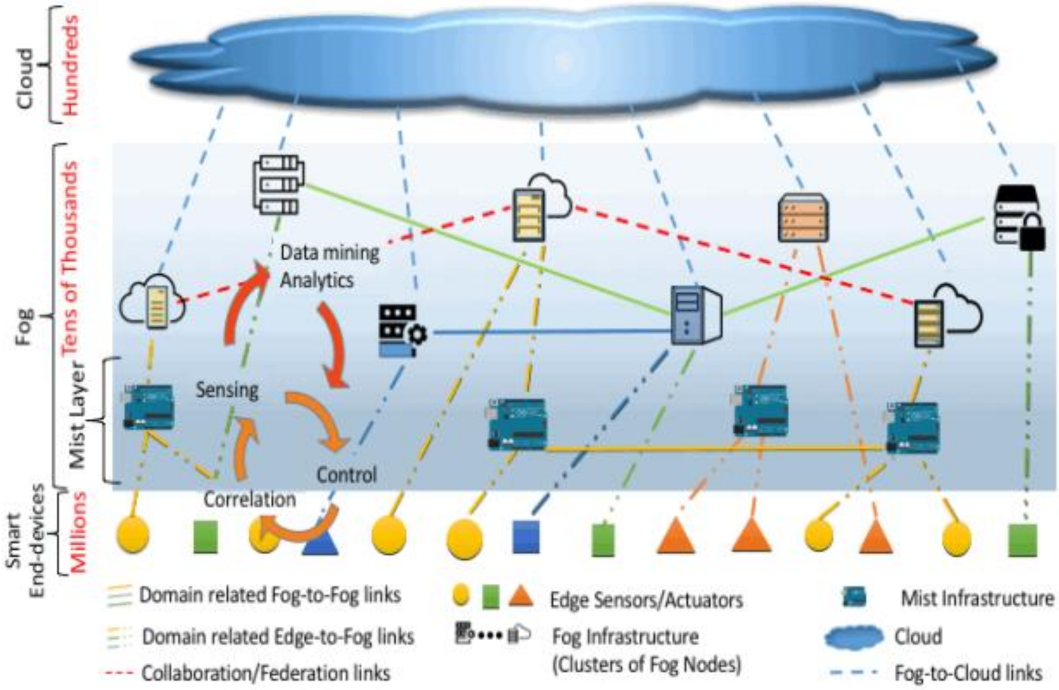
6. **Δύσκολη η μεταφορά μεγάλου όγκου δεδομένων**
   Υπάρχει γενικά αδυναμία να ικανοποιήσει συστήματα και εφαρμογές που απαιτούν τη μεταφορά μεγάλων ποσοτήτων δεδομένων, λόγω της έμφυτης αδυναμίας της συνδεσιμότητας WAN που απαιτεί το cloud για να λειτουργήσει.
7. **Περιορισμένη προσαρμοστικότητα**
   Δεδομένου ότι μια εφαρμογή cloud απευθύνεται σε πολλούς διαφορετικούς χρήστες (είτε πρόκειται για επιχειρήσεις, είτε για ιδιώτες), είναι δύσκολο να έχει μεγάλη προσαρμοστικότητα στις απαιτήσεις και επιθυμίες του εκάστοτε χρήστη. Αντίθετα, μια εφαρμογή που κατασκευάζεται εντός των εγκαταστάσεων μιας επιχείρησης μπορεί να είναι κατασκευασμένη με τρόπο προσωποποιημένο, που να καλύπτει όλες τις ανάγκες και επιθυμίες της εκάστοτε επιχείρησης (ή του εκάστοτε χρήστη).

1.3 Ορισμός και περιγραφή του Fog Computing

Με τον όρο fog computing, επίσης γνωστό και ως fog networking ή fogging, περιγράφουμε μια αποκεντρωμένη υπολογιστική υποδομή στην οποία τα δεδομένα, οι υπολογισμοί, ο αποθηκευτικός χώρος και οι εφαρμογές κατανέμονται στο πιο λογικό και αποδοτικό μέρος ανάμεσα στην πηγή των δεδομένω και το cloud. Το fog computing επεκτείνει το cloud computing και τις υπηρεσίες cloud στις άκρες, στα σύνορα του δικτύου, φέρνοντας τα πλεονεκτήματα και τη δύναμη του cloud πιο κοντά στο μέρος όπου δημιουργούνται τα δεδομένα. [4]

Σύμφωνα με το ινστιτούτο NIST [5], το fog computing είναι ένα οριζόντιο φυσικό ή εικονικό σύνολο πόρων που βρίσκεται μεταξύ των έξυπνων τελικών συσκευών και των παραδοσιακών datacenter ή του cloud. Αυτό το σύνολο υποστηρίζει απομονωμένες καθέτως, ευαίσθητες ως προς την καθυστέρηση εφαρμογές παρέχοντας πανταχού παρόντες, κλιμακώσιμους, επιπεδοποιημένους, ομοσπονδιοποιημένους και κατανεμημένους υπολογιστικούς και αποθηκευτικούς πόρους και δικτυακή συνδεσιμότητα. Η εικόνα της επόμενης σελίδας απεικονίζει το fog computing στο ευρύτερο περιβάλλον ενός cloud-based οικοσυστήματος που υπηρετεί έξυπνες τελικές συσκευές. Είναι σημαντικό να σημειώσουμε ότι το fog computing δεν θεωρείται ως υποχρεωτικό επίπεδο ενός τέτοιου οικοσυστήματος.

Οι fog κόμβοι είναι ενδιάμεσα υπολογιστικά στοιχεία του δικτύου πρόσβασης των έξυπνων τελικών συσκευών, τα οποία βρίσκονται ανάμεσα στο cloud και τις έξυπνες τελικές συσκευές. Μπορεί να είναι είτε φυσικά είτε εικονικά στοιχεία και είναι στενά συνδεδεμένα με τις τελικές συσκευές ή τα δίκτυα πρόσβασης. Οι fog κόμβοι τυπικά παρέχουν κάποιου είδους διαχείρισης δεδομένων και υπηρεσίας επικοινωνίας μεταξύ του περιφερειακού επιπέδου όπου βρίσκονται οι έξυπνες τελικές συσκευές και του cloud. Τέλος, ειδικά οι εικονικοί fog κόμβοι, που αναφέρονται επίσης και ως cloudlets, μπορούν να ομοσπονδιοποιηθούν για να παρέχουν οριζόντια επέκταση της λειτουργικότητας σε διασκορπισμένες γεωγραφικές τοποθεσίες.



Εικόνα 4: Το fog computing υποστηρίζει ένα cloud-based οικοσύστημα για έξυπνες τελικές συσκευές

Παρακάτω παραθέτουμε τα χαρακτηριστικά του fog computing και πάλι όπως περιγράφονται από το ινστιτούτο NIST [5]:

- **Σχετική με το περιβάλλον επίγνωση τοποθεσίας και χαμηλή καθυστέρηση:** Το fog computing προέρχεται από αρχικές προτάσεις ειδικών που υποστήριζαν ακραίες συσκευές με πλούσιες υπηρεσίες στα σύνορα ενός δικτύου, συμπεριλαμβανομένων εφαρμογών με απαιτήσεις για χαμηλή καθυστέρηση (πχ gaming, video streaming, επαυξημένη πραγματικότητα, κα). Επειδή οι fog κόμβοι τείνουν να βρίσκονται πολύ κοντά στις ακραίες IoT συσκευές, η στατιστική ανάλυση και η απόκριση στα δεδομένα που παράγονται από αυτές είναι πολύ γρηγορότερη συγκριτικά με ένα κεντρικοποιημένο σύστημα cloud.
- **Γεωγραφική κατανομή:** Σε έντονη αντίθεση με το πιο κεντρικοποιημένο cloud, οι υπηρεσίες και οι εφαρμογές όπου στοχεύει το fog computing απαιτούν ευρέως κατανεμημένη ανάπτυξη. Για παράδειγμα, το fog computing θα διαδραματίσει ενεργό ρόλο στο να προσφέρει υπηρεσίες streaming υψηλής ποιότητας σε κινούμενα οχήματα, μέσω από proxies και access points κατά μήκος αυτοκινητοδρόμων. Δίκτυα αισθητήρων μεγάλης κλίμακας για την παρακολούθηση του περιβάλλοντος και τα Smart Grid είναι άλλα παραδείγματα από εγγενώς κατανεμημένα συστήματα που απαιτούν κατανεμημένους υπολογισμούς και αποθηκευτικούς πόρους.
- **Πολύ μεγάλος αριθμών κόμβων** σαν συνέπεια της ευρείας γεωγραφικής κατανομής, όπως αποδεικνύεται στα δίκτυα αισθητήρων γενικά και ειδικότερα στα Smart Grids.
- **Υποστήριξη για κινητικότητα:** Είναι σημαντικό για πολλές εφαρμογές fog computing να επικοινωνούν απευθείας με κινητές συσκευές και γι’ αυτό να υποστηρίζουν τεχνικές κινητικότητας, όπως το πρωτόκολλο LISP, το οποίο

αποσυνδέει την ταυτότητα του host από την ταυτότητα τοποθεσίας και απαιτεί ένα κατανεμημένο σύστημα καταλόγων.

- **Αλληλεπίδραση πραγματικού χρόνου:** Σημαντικές εφαρμογές fog computing εμπεριέχουν αλληλεπίδραση πραγματικού χρόνου παρά διαδικασίες επεξεργασίας συνόλων (batch processing).
- **Κυριαρχία της ασύρματης πρόσβασης:** Αν και το fog computing μπορεί να χρησιμοποιείται σε περιβάλλοντα ενσύρματης σύνδεσης, ο μεγάλος αριθμό από ασύρματους αισθητήρες στο IoT απαιτεί κατανεμημένους υπολογισμούς και στατιστικές αναλύσεις. Γι' αυτό το λόγο το fog computing ταιριάζει πολύ καλά με τα ασύρματα δίκτυα πρόσβασης IoT.
- **Ετερογένεια:** Οι fog κόμβοι εμφανίζονται σε πολλές διαφορετικές μορφές και αναπτύσσονται σε μεγάλης ποικιλίας περιβάλλοντα. Επίσης, οι συσκευές απ' όπου συλλέγουν δεδομένα μπορεί να διαφέρουν σε μορφή και στη δυνατότητα δικτυακής επικοινωνίας.
- **Διαλειτουργικότητα και ομοσπονδιοποίηση:** Η αδιάλειπτη υποστήριξη συγκεκριμένων υπηρεσιών (ένα καλό παράδειγμα είναι οι υπηρεσίες streaming πραγματικού χρόνου) απαιτούν τη συνεργασία διαφορετικών παρόχων. Γι' αυτό τα συστατικά στοιχεία ενός fog computing συστήματος πρέπει να μπορούν να υποστηρίζουν τη διαλειτουργικότητα και οι υπηρεσίες πρέπει να είναι ομοσπονδιοποιημένες στους διάφορους τομείς (domains).
- **Υποστήριξη για στατιστική ανάλυση πραγματικού χρόνου και συνεργασία με το cloud:** Το fog computing καλείται να παίξει καθοριστικό ρόλο στην λήψη και την επεξεργασία των δεδομένων κοντά στην πηγή τους καθώς αυτά παράγονται. Αν και οι fog κόμβοι παρέχουν τοπικοποίηση, επιτρέποντας γι' αυτό χαμηλή καθυστέρηση και επίγνωση του περιβάλλοντος, το Cloud παρέχουν καθολική κεντρικοποίηση. Πολλές εφαρμογές απαιτούν τόσο την τοπικοποίηση του fog computing όσο και την καθολικοποίηση του cloud, ειδικά για στατιστική ανάλυση και Big Data. Το fog computing ταιριάζει χαρακτηριστικά καλά σε πραγματικού χρόνου, streaming στατιστική ανάλυση σε αντίθεση με την στατιστική ανάλυση ενός Big Data συνόλου που κανονικά πραγματοποιείται κεντρικά σε ένα data center.

Το fog computing είναι μια επέκταση του παραδοσιακού μοντέλου του cloud computing, όπου υλοποιήσεις της αρχιτεκτονικής μπορούν να βρίσκονται σε πολλαπλά επίπεδα μιας τοπολογίας δικτύου. Όπως και με το cloud, οι ακόλουθοι τύποι μοντέλων υπηρεσιών μπορούν να αναπτυχθούν:

- **Λογισμικό ως υπηρεσία (Software as a Service - SaaS):** Η δυνατότητας που παρέχεται στον πελάτη της υπηρεσίας fog είναι να χρησιμοποιεί τις εφαρμογές του παρόχου υπηρεσιών fog, οι οποίες τρέχουν σε ένα cluster από ομοσπονδιοποιημένους fog κόμβους που διαχειρίζεται ο πάροχος. Αυτού του τύπου η υπηρεσία είναι παρόμοια με το μοντέλο Software as a Service του cloud computing και συνεπάγεται ότι η τελική συσκευή αποκτά πρόσβαση στις εφαρμογές του fog κόμβου μέσω μιας ελαφριάς διεπαφής πελάτη ή μιας διεπαφής προγράμματος. Ο τελικός χρήστης δεν διαχειρίζεται ή ελέγχει την υποκείμενη υποδομή του fog κόμβου η οποία περιλαμβάνει το δίκτυο, τους servers, τα λειτουργικά συστήματα, τον αποθηκευτικό χώρο ή ακόμα και ξεχωριστές δυνατότητες εφαρμογής, με την πιθανή εξαίρεση κάποιων περιορισμένων ρυθμίσεων παραμετροποίησης της εφαρμογής που είναι συγκεκριμένα για τον εκάστοτε χρήστη.

- **Πλατφόρμα ως υπηρεσία (Platfom as a Service - Paas):** Η δυνατότητα που παρέχεται στον πελάτη της fog υπηρεσίας είναι παρόμοια με το μοντέλο Platform as a Service του cloud computing και επιτρέπει την ανάπτυξη πάνω σε πλατφόρμες των ομοσπονδιοποιημένων fog κόμβων. Έτσι δημιουργείται ένα cluster από εφαρμογές που δημιουργεί ο πελάτης ή από εφαρμογές που αποκτά ο πελάτης και έχουν δημιουργηθεί χρησιμοποιώντας γλώσσες προγραμματισμού, βιβλιοθήκες, υπηρεσίες και εργαλεία που υποστηρίζει ο πάροχος υπηρεσιών fog. Ο πελάτης δεν διαχειρίζεται ή ελέγχει την υποκείμενη fog πλατφόρμα και υποδομή η οποία περιλαμβάνει το δίκτυο, τους servers, τα λειτουργικά συστήματα και τον αποθηκευτικό χώρο, αλλά έχει τον έλεγχο πάνω στις εφαρμογές που αναπτύσσονται και πιθανώς σε ρυθμίσεις παραμετροποίησης για το περιβάλλον που φιλοξενεί αυτές τις εφαρμογές.
- **Υποδομή ως υπηρεσία (Infrastructure as a Service - IaaS):** Η δυνατότητα που παρέχεται στον πελάτη της fog υπηρεσίας είναι να παρέχει επεξεργαστικούς, αποθηκευτικούς, δικτυακούς και άλλους θεμελιώσεις υπολογιστικούς πόρους αξιοποιώντας την υποδομή των fog κόμβων οι οποίοι δημιουργούν ένα ομοσπονδιοποιημένο cluster. Όπως και στο μοντέλο Infrastructure as a Service του cloud computing, ο πελάτης μπορεί να αναπτύσσει και να τρέχει λογισμικό, το οποίο μπορεί να περιλαμβάνει λειτουργικά συστήματα και εφαρμογές. Ο πελάτης δεν διαχειρίζεται ή ελέγχει την υποκείμενη υποδομή του cluster των fog κόμβων, αλλά έχει τον έλεγχο πάνω στα λειτουργικά συστήματα, στον αποθηκευτικό χώρο και τις εφαρμογές που αναπτύσσονται πάνω στην υποδομή. Επίσης, έχει πιθανώς περιορισμένο έλεγχο στο να επιλέγει συστατικά στοιχεία του δικτύου (πχ τείχη προστασίας - firewalls).

Αφού το fog computing αναγνωρίζεται και ορίζεται ως μια επέκταση του παραδοσιακού μοντέλου του cloud computing, υποστηρίζονται επίσης τα ακόλουθα μοντέλα ανάπτυξης:

- **Ιδιωτικός fog κόμβος (Private fog node):** Είναι ένας κόμβος που παρέχεται για αποκλειστική χρήση από έναν μοναδικό οργανισμό που αποτελείται από πολλαπλούς πελάτες (πχ επιχειρησιακές μονάδες). Μπορεί να τον κατέχει, να τον διαχειρίζεται και να τον λειτουργεί ο οργανισμός ή κάποιος τρίτος ή συνδυασμός αυτών και μπορεί να βρίσκεται εντός ή εκτός των χώρων του οργανισμού.
- **Fog κόμβος κοινότητας (Community fog node):** Είναι ένας κόμβος που παρέχεται για αποκλειστική χρήση από μια συγκεκριμένη κοινότητα πελατών από οργανισμούς που μοιράζονται κοινά ενδιαφέροντα (πχ αποστολής, απαιτήσεις ασφάλειας, πολιτικές και σκέψεις συμμόρφωσης). Μπορεί να τον κατέχεται, να τον διαχειρίζεται και να τον λειτουργεί ένας ή περισσότεροι οργανισμοί μέσα στην κοινότητα, κάποιος τρίτος ή συνδυασμός αυτών και μπορεί να βρίσκεται εντός ή εκτός των χώρων της κοινότητας.

- **Δημόσιος fog κόμβος (Public fog node):** Είναι ένας fog κόμβος ο οποίος παρέχεται για ανοιχτή χρήση από το γενικό κοινό. Μπορεί να τον κατέχει, να τον διαχειρίζεται και να τον λειτουργεί μια επιχείρηση, ένας ακαδημαϊκός ή κυβερνητικός οργανισμός ή κάποιος συνδυασμός αυτών. Βρίσκεται στους χώρους του παρόχου fog υπηρεσιών.
- **Υβριδικός fog κόμβος (Hybrid fog node):** Είναι ένας σύνθετος fog κόμβος ο οποίος αποτελεί σύνθεση δύο ή περισσότερων fog κόμβων (ιδιωτικών, κοινότητας ή δημόσιων) που παραμένουν μοναδικές οντότητες αλλά είναι συνδεδεμένοι μεταξύ τους με προτυποποιημένη ή ιδιόκτητη τεχνολογία, η οποία επιτρέπει την φορητότητα δεδομένων και εφαρμογών.

1.4 Σύγκριση μεταξύ Cloud και Fog Computing

Από όσα αναφέρθηκαν παραπάνω μπορεί κανείς να καταλάβει ότι από τεχνικής άποψης το fog computing και το cloud computing μοιάζουν αρκετά υπό την έννοια ότι και τα δύο αποτελούνται από εικονικά συστήματα που παρέχουν την ευελιξία της κατ' απαίτησης παροχής υπολογιστικών, αποθηκευτικών και δικτυακών πόρων. Ωστόσο συγκριτικά με το cloud τα συστήματα του fog computing υλοποιούνται πιο κοντά στις τελικές συσκεύες ή στους τελικούς χρήστες. Βέβαια οι edge συσκευές και οι αισθητήρες ενώ βρίσκονται εκεί που τα δεδομένα παράγονται και συλλέγονται, δεν έχουν τους υπολογιστικούς και αποθηκευτικούς πόρους για να πραγματοποιήσουν προηγμένη ανάλυση και εργασίες machine learning. Από την άλλη οι cloud servers έχουν τη δύναμη να πραγματοποιήσουν αυτά, αλλά συχνά βρίσκονται πολύ μακριά για να επεξεργαστούν τα δεδομένα και να απαντήσουν την κατάλληλη στιγμή. Επιπλέον, το να έχεις όλα τα ακραία στοιχεία ενός συστήματος να συνδέονται και να στέλνουν ανεπεξέργαστα δεδομένα στο cloud μέσω του Internet μπορεί να έχει ζητήματα με την ιδιωτικότητα, την ασφάλεια και τη νομιμότητα, ειδικά όταν αφορά ευαίσθητα δεδομένα που υπόκεινται σε διάφορους κανονισμούς σε διαφορετικές χώρες. [4]

Οι σημαντικότεροι περιορισμοί περιορισμοί του cloud computing περιλαμβάνουν την απαίτηση για υψηλής ταχύτητας αξιόπιστη σύνδεση στο Internet και μερικές φορές πολλαπλή πρόσβαση (multi-homing) για την αποφυγή διακοπής λειτουργίας, υψηλή καθυστέρηση, ακαθόριστη ασφάλεια, κα. Επίσης, λόγω της ετερογενούς και χαλαρά ελεγχόμενης φύσης του Internet, υπάρχουν αρκετά ζητήματα που σχετίζονται με την ποιότητα της υπηρεσίας (Quality of Service - QoS). Οι αναδυόμενες τάσεις στη δικτύωση όπως τα μεγάλα κατανεμημένα δίκτυα αισθητήρων συνδεδεμένα μέσω Internet, το Internet of Things, τα κινητά δίκτυα δεδομένων καθώς και οι εφαρμογές streaming πραγματικού χρόνου έχουν χαρακτηριστικά που δεν μπορούν να ικανοποιηθούν από το cloud computing.

Αντίθετα, το fog computing μπορεί να παρέχει καλύτερη ποιότητα υπηρεσίας από άποψη καθυστέρησης, κατανάλωσης ισχύος, μειωμένου όγκου δεδομένων που αποστέλλονται μέσω Internet, κτλ. Το βασικό χαρακτηριστικό του fog computing είναι η δυνατότητά του να υποστηρίζει εφαρμογές που απαιτούν χαμηλή καθυστέρηση, επίγνωση τοποθεσίας και κινητικότητα. Αυτή η δυνατότητα οφείλεται στο γεγονός ότι τα συστήματα fog computing αναπτύσσονται κοντά στους τελικούς χρήστες με έναν ευρέως κατανεμημένο τρόπο. [6]

Ο στόχος του fog computing είναι να βελτιώσει την αποδοτικότητα και να μειώσει την ποσότητα των δεδομένων που μεταφέρονται στο cloud για επεξεργασία, ανάλυση και αποθήκευση. Πέρα από την αύξηση της αποδοτικότητας αυτό, όπως είπαμε, μπορεί να συμβαίνει και για λόγους ασφάλειας και συμμόρφωσης με κανονιστικές ρυθμίσεις. Σε ένα fog περιβάλλον, η επεξεργασία λαμβάνει χώρα σε ένα κέντρο δεδομένων πάνω σε μια έξυπνη συσκευή ή σε ένα έξυπνο router ή gateway, μειώνοντας έτσι την ποσότητα των δεδομένων που αποστέλλονται στο cloud. Είναι σημαντικό να σημειώσουμε ότι το fog computing συμπληρώνει και δεν αντικαθιστά το cloud computing. Το fog computing επιτρέπει βραχυπρόθεσμη στατιστική ανάλυση κοντά στην πηγή των δεδομένων και το cloud πραγματοποιεί μακροπρόθεσμη στατιστική ανάλυση, η οποία απαιτεί μεγάλη ποσότητα πόρων. [4]

Στον πίνακα 1 μπορεί κανείς να δει ότι τα χαρακτηριστικά του cloud computing έχουν σοβαρούς περιορισμούς σε ό,τι αφορά την ποιότητα υπηρεσίας που απαιτείται από εφαρμογές πραγματικού χρόνου που χρειάζονται σχεδόν άμεση δράση από τον server. Στη συνέχεια δίνεται αναλυτική εξήγηση στη σύγκριση των δύο τεχνολογιών από την άποψη της επίδρασής τους.

Χαρακτηριστικό	**Cloud Computing**	**Fog Computing**
Καθυστέρηση	Υψηλή	Χαμηλή
Διακύμανση Καθυστέρησης (Delay Jitter)	Υψηλή	Πολύ Χαμηλή
Τοποθεσία των server κόμβων	Μέσα στο Internet	Στο άκρο του τοπικού δικτύου
Απόσταση μεταξύ client και server	Πολλαπλά βήματα (hops)	Ένα βήμα (hop)
Ασφάλεια	Ακαθόριστη	Μπορεί να καθοριστεί
Επίθεση στα δεδομένα κατά τη διαδρομή	Υψηλή πιθανότητα	Πολύ χαμηλή πιθανότητα
Επίγνωση τοποθεσίας	Όχι	Ναι
Γεωγραφική κατανομή	Κεντρικοποιημένη	Κατανεμημένη
Πλήθος των server κόμβων	Μικρό	Πολύ μεγάλο
Υποστήριξη κινητικότητας	Περιορισμένη	Υποστηρίζεται κανονικά

Πίνακας 1: Σύγκριση χαρακτηριστικών των Cloud και Fog Computing

Το Internet of Things οραματίζεται να βάλει μια διεύθυνση IP σε κάθε συσκευή συνδέοντάς την στο Internet. Τα δίκτυα των οχημάτων, μια ειδική κατηγορία των αναδυόμενων κινητών δικτύων, είναι ένα σημαντικό μέρος του IoT όπως το οραματίζονται οι ειδικοί. Στα δίκτυα των οχημάτων η επικοινωνία θα λαμβάνει χώρα μεταξύ των οχημάτων, μεταξύ των οχημάτων και των σημείων πρόσβασης και μεταξύ των σημείων πρόσβασης για τη μεταφορά σημαντικών πληροφοριών μεταξύ αυτών των αντικειμένων. Μερικά από τα σημαντικά χαρακτηριστικά αυτής της επικοινωνίας περιλαμβάνουν επίγνωση τοποθεσίας, ασύρματη σύνδεση, απαιτήσεις γρηγορότερης επεξεργασίας, χαμηλή καθυστέρηση, αλληλεπίδρασης πραγματικού χρόνου και κινητικότητα. Για να μπορέσουν να καλυφθούν αυτές οι απαιτήσεις πρέπει να υπάρχει ένας μεγάλος αριθμός από κόμβους επεξεργασίας που να βρίσκονται πολύ κοντά στα οχήματα, οι οποίοι να μπορούν να επικοινωνούν με μικρή καθυστέρηση με τα οχήματα ασύρματα. Γι’ αυτό το fog computing είναι πιο κατάλληλο μοντέλο επικοινωνίας συγκριτικά με το cloud computing όπου η επεξεργασία των πληροφοριών λαμβάνει χώρα βαθιά μέσα στο Internet.

Οι αναδυόμενες υψηλής ποιότητας εφαρμογές πολυμέσων συμπεριλαμβανομένων κατανεμημένων διαδραστικών παιχνιδιών, βίντεο κατ’ απαίτηση και streaming απαιτούν μεγάλους ρυθμούς μεταφοράς δεδομένων με χαμηλή καθυστέρηση, χαμηλή διακύμανση καθυστέρησης και απώλεια πακέτων. Για να επιτευχθεί αυτό είναι απαραίτητο να επεξεργάζονται αυτές οι εφαρμογές κοντά στους τελικούς χρήστες. Από τη στιγμή που τα κέντρα δεδομένων του cloud είναι γενικά τοποθετημένα μέσα στο Internet είναι δύσκολο να διαχειριστούν αυτούς τους παράγοντες. Γι’ αυτό το fog computing είναι η πρακτική λύση γι’ αυτού του είδους της ευαίσθητες σε απόδοση εφαρμογές.

Τα ασύρματα δίκτυα αισθητήρων έχουν αναπτυχθεί ευρέως σε πολλές εφαρμογές σχετικές με το περιβάλλον. Αυτά τα δίκτυα γενικά χαρακτηρίζονται από χαμηλή ισχύ, χαμηλό bandwidth και κόμβους χαμηλών επεξεργαστικών δυνατοτήτων κατανεμημένους κατά μήκος ευρέων γεωγραφικών περιοχών. Αυτά τα δίκτυα πρέπει να υποστηρίζονται από χαμηλής καθυστέρησης, ευρέως κατανεμημένα συστήματα με επίγνωση τοποθεσίας για την επεξεργασία και τη διανομή των δεδομένων. Αυτά είναι τυπικά χαρακτηριστικά περισσότερο του fog παρά του cloud computing.

Η ασφάλεια και η ακεραιότητα των δεδομένων είναι δύο σημαντικά χαρακτηριστικά που απαιτούνται από πολλές εφαρμογές Internet. Όσο μεγαλύτερη διαδρομή διανύουν τα δεδομένα τόσο περισσότερο ευάλωτα γίνονται σε επιθέσεις ακόμα και κρυπτογραφημένα. Γι’ αυτό είναι πάντα επιθυμητό να υπάρχουν λίγα βήματα ανάμεσα στους clients και τους servers. Έτσι, το fog computing προτιμάται σε τέτοιες περιπτώσεις έναντι του cloud computing, μιας και σε συνδυασμό με όλα τα υπόλοιπα πλεονεκτήματα του cloud παρέχει επίσης πολύ μικρότερη απόσταση ανάμεσα στους clients και τους servers.

Ακόμα και η διαθεσιμότητα των cloud συστημάτων που βρίσκονται στο Internet μπορεί να δεχθεί επίθεση χρησιμοποιώντας διάφορες μεθόδους επίθεσης Άρνησης Υπηρεσίας (Denial of Service - DoS Attack). Οι επιθέσεις DoS δεν χρειάζεται να γίνουν απευθείας στα τελικά συστήματα. Ακόμα και επιθέσεις σε ενδιάμεσες συσκευές όπως τα routers μπορεί να είναι το ίδιο θανάσιμες. Γι’ αυτό υπάρχουν πολλές ευκαιρίες για τους hackers για να στοχεύσουν cloud συστήματα. Από την άλλη πλευρά οι κόμβοι του fog computing είναι ιδιαίτερα κατανεμημένοι κοντά στα άκρα των δικτύων των χρηστών και έτσι για να πέσει θύμα επίθεσης η διαθεσιμότητα αυτών των συστημάτων, απαιτείται να γίνεται μια εκτεταμένη επίθεση σε όλα τα συστήματα που βρίσκονται κοντά σε έναν client. Αυτό χρειάζεται εκτεταμένους πόρους από την πλευρά των επιτιθεμένων επίσης. Ακόμα δεν υπάρχουν πολλές ενδιάμεσες συσκευές που μπορεί να γίνουν στόχοι επίθεσης καθώς οι υπολογιστικοί κόμβοι βρίσκονται κοντά στους τελικούς χρήστες. Γι' αυτό μπορούμε να πούμε ότι τα συστήματα fog computing είναι λιγότερο ευάλωτα σε DoS επιθέσεις συγκριτικά με τα cloud συστήματα. [6]

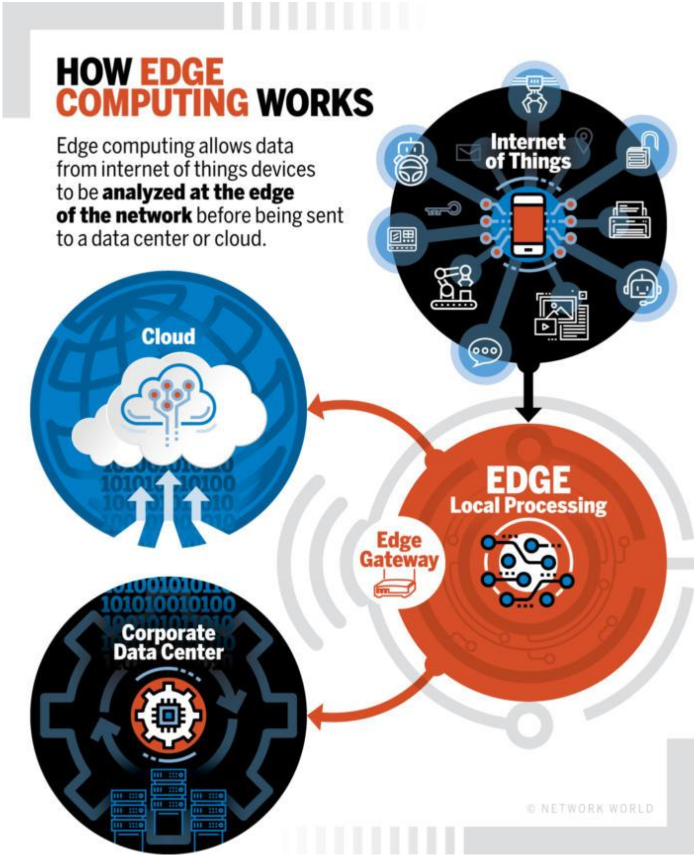
Από τα παραπάνω φαίνεται ότι το fog computing ανταποκρίνεται καλύτερα στις ανάγκες των χρηστών και στα νέες αναδυόμενες υπολογιστικές και δικτυακές εφαρμογές απ' ότι τα παραδοσιακά συστήματα cloud. Ακόμα το fog computing είναι πιο ανθεκτικό και ασφαλές από το cloud computing στο πρόσωπο της αλλαγής των αναγκών και των αναδυόμενων τάσεων.

Από την άλλη πλευρά πρέπει να αναφερθεί ότι το cloud computing έχει και αυτό τα πλεονεκτήματά του, όπως αναφέρθηκαν αναλυτικά και σε προηγούμενη ενότητα. Από τη στιγμή που το fog computing απαιτεί εκτεταμένη γεωγραφικά κατανεμημένη υλοποίηση οι μεμονωμένοι κόμβοι δεν μπορούν να έχουν μεγάλες ποσότητες πόρων για οικονομικούς λόγους. Ωστόσο το υψηλής ποιότητας, ακριβό business computing όπως οι εργασίες batch processing απαιτούν μεγάλη ποσότητα πόρων, ενώ δεν είναι πολύ ευαίσθητες σε θέματα καθυστέρησης. Αυτού του είδους οι εργασίες μπορούν να εκτελεστούν καλύτερα χρησιμοποιώντας συστήματα cloud computing παρά fog κόμβους. Γι' αυτό το fog computing δεν θα μπορέσει ποτέ να αντικαταστήσει το cloud computing και να γίνεται το μοναδικό υπολογιστικό μοντέλο του μέλλοντος. Έτσι, μπορούμε με ασφάλεια να πούμε ότι το cloud και το fog computing θα συνυπάρχουν υπηρετώντας διαφορετικές κοινότητες και ενδιαφέροντα και συμπληρώνοντας το ένα το άλλο όπου χρειάζεται.

1.5 Ορισμός και περιγραφή του Edge Computing

Το edge computing είναι ένα αρχιτεκτονικό υπολογιστικό μοντέλο στο οποίο η επεξεργασία των δεδομένων των clients γίνεται στην περιφέρεια του δικτύου, όσο πιο κοντά γίνεται στην πηγή των δεδομένων. Ο όρος "edge" προέρχεται από τα δικτυακά διαγράμματα. Τυπικά "edge" σε ένα δικτυακό διάγραμμα είναι το σημείο στο οποίο η κίνηση εισέρχεται ή εξέρχεται του δικτύου. "Edge" είναι ακόμα το σημείο στο οποίο το υποκείμενο πρωτόκολλο για την μεταφορά των δεδομένων μπορεί να αλλάξει. Για παράδειγμα, ένας έξυπνος αισθητήρας μπορεί να χρησιμοποιεί ένα πρωτόκολλο χαμηλής καθυστέρησης όπως το MQTT για να μεταδώσει τα δεδομένα σε έναν broker μηνυμάτων που βρίσκεται στα άκρα του δικτύου και ο broker χρησιμοποιεί το πρωτόκολλο HTTP για να μεταδώσει χρήσιμα δεδομένα από τον αισθητήρα σε έναν απομακρυσμένα server στο Internet.

Η κίνηση προς το edge computing καθοδηγείται από το mobile computing, το μειούμενο κόστος των συστατικών μερών των υπολογιστών και τον ραγδαία αυξανόμενο αριθμό των διασυνδεδεμένων συσκευών στο Internet of Things. Ανάλογα με την υλοποίηση, σε μια edge computing αρχιτεκτονική η επεξεργασία δεδομένων που είναι ευαίσθητα στο χρόνο και την καθυστέρηση μπορεί να γίνεται στο σημείο της δημιουργίας τους από μια έξυπνη συσκευή ή να στέλνονται σε έναν ενδιάμεσο server που να βρίσκεται σε κοντινή γεωγραφική απόσταση από τον client. Δεδομένα με λιγότερες απαιτήσεις σε θέματα χρόνου στέλνονται στο cloud για ιστορική ανάλυση, ανάλυση big data και μακροχρόνια αποθήκευση. Το edge computing λοιπόν συλλέγει τα δεδομένα έτσι ώστε κάποια να επεξεργάζονται τοπικά, μειώνοντας την κίνηση προς ένα κεντρικό αποθετήριο.



Εικόνα 5: Εποπτική εικόνα του Edge Computing

Η μετάδοση τεράστιων ποσοτήτων από ανεπεξέργαστα δεδομένα πάνω από το δίκτυο επιβαρύνει σε μεγάλο βαθμό τους δικτυακούς πόρους. Σε μερικές περιπτώσεις είναι πολύ πιο αποδοτικό να επεξεργάζεσαι τα δεδομένα κοντά στην πηγή τους και να στέλνεις μόνο τα δεδομένα που έχουν αξία πάνω από το δίκτυο σε ένα απομακρυσμένα κέντρο δεδομένων. Για παράδειγμα, αντί να μεταδίδει διαρκώς δεδομένα για το επίπεδο λαδιού στη μηχανή ενός αυτοκινήτου, ένας αισθητήρας αυτοκίνησης μπορεί απλά να στέλνει περιοδικά μια σύνοψη των δεδομένων σε έναν απομακρυσμένο server. [7]

Πολλοί χρησιμοποιούν τους όρους fog computing και edge computing εναλλακτικά, καθώς και οι δύο περιλαμβάνουν το να φέρει κανείς την ευφυΐα και την επεξεργαστική διαδικασία πιο κοντά εκεί όπου παράγονται τα δεδομένα. Ωστόσο, η βασική διαφορά ανάμεσα στα δύο είναι το σημείο όπου τοποθετείται η ευφυΐα και η υπολογιστική ισχύς. Σε ένα fog περιβάλλον η ευφυΐα βρίσκεται στο τοπικό δίκτυο. Τα δεδομένα μεταδίδονται από τις ακραίες συσκευές σε ένα gateway όπου στη συνέχεια μεταδίδονται σε πηγές για επεξεργασία και επαναποστολή. Στο edge computing η ευφυΐα και η ισχύς ενός edge gateway ή συσκευής βρίσκεται σε συσκευές όπως ένα αυτόματος προγραμματιζόμενος ελεγκτής. [4]

Τα συστήματα edge computing είναι ιδανικά σε ποικιλία περιπτώσεων. Μία από αυτές είναι όταν συσκευές IoT έχουν κακής ποιότητας συνδεσιμότητα και δεν είναι αποδοτικό γι’ αυτές να είναι συνεχώς συνδεδεμένες σε ένα κεντρικό cloud. Άλλες περιπτώσεις έχουν να κάνουν με την επεξεργασία πληροφορίας, η οποία είναι ευαίσθητη σε καθυστέρηση. Το edge computing μειώνει την καθυστέρηση επειδή τα δεδομένα δεν χρειάζεται να ταξιδέψουν πάνω από το δίκτυο για να φτάσουν σε ένα data center ή cloud server για να επεξεργαστούν. Αυτό είναι ιδανικό για περιπτώσεις όπου καθυστερήσεις της τάξης των millisecond δεν είναι βιώσιμες, όπως στα οικονομικά συστήματα ή στον κατασκευαστικό κλάδο. [8]

Το edge computing προσδίδει ξεχωριστή αξία τόσο σε καταναλωτικές όσο και σε επιχειρησιακές περιπτώσεις χρήσης του IoT. Μπορεί να βοηθήσει να μειωθεί το κόστος συνδεσιμότητας στέλνοντας μόνο την πληροφορία που έχει αξία αντί ανεπεξέργαστων ροών δεδομένων αισθητήρων, το οποίο είναι ιδιαίτερα χρήσιμο σε συσκευές που συνδέονται μέσω LTE και κινητών δικτύων γενικότερα, όπως έξυπνους μετρητές ή ανιχνευτές επενδυτικών αγαθών (asset trackers). Ακόμα όταν έχει κανείς να κάνει με μια τεράστια ποσότητα δεδομένων που παράγονται από αισθητήρες σε μια βιομηχανική εγκατάσταση ή σε μια επιχείρηση εξόρυξης για παράδειγμα, το να έχει τη δυνατότητα να αναλύσει και να φιλτράρει τα δεδομένα πριν την αποστολή τους μπορεί να οδηγήσει σε μεγάλη εξοικονόμηση δικτυακών και υπολογιστικών πόρων. [9]

1.6 Πλεονεκτήματα και προκλήσεις του Edge Computing

Καθώς πολυάριθμες επιχειρησιακές υπηρεσίες έχουν αντικαταστήσει το cloud με το edge computing λόγω των προηγμένων υπολογιστικών υπηρεσιών του, η οικονομία κλίμακας έχει βελτιωθεί σημαντικά, παρέχοντας τεράστια οφέλη στις επιχειρησιακές υπηρεσίες που παρέχουν την υποδομή όσο και στις επιχειρήσεις που την χρησιμοποιούν. Ακολουθούν ορισμένα από τα σημαντικότερα οφέλη υπηρεσιών του edge computing [10]:

- Με το edge computing είναι ταχύτερη η διαδικασία ανάλυσης των δεδομένων, το οποίο βελτιώνει συνολικά την απόδοση των real-time εφαρμογών.
- Το κόστος υλοποίησης ενός data center μειώνεται σημαντικά επιλέγοντας κόμβους ως edge servers.
- Ελαττώνεται ο φόρτος στις συνδέσεις στο backhaul μέρος του δικτύου, καθώς μειώνεται συνολικά η δικτυακή κίνηση.
- Εξαλείφει την ύπαρξη μοναδικών σημείων αστοχίας και υιοθετεί την κατανεμημένη λογική στους υπολογισμούς.
- Αυξάνει την εικονικοποίηση και την κλιμακωσιμότητα στο δίκτυο
- Βελτιώνει την ποιότητα υπηρεσίας ελαχιστοποιώντας την απόσταση μεταφοράς των δεδομένων.
- Πετυχαίνει αξιοπιστία εγκαθιστώντας εφαρμογές σε κοντινή απόσταση από τις τελικές συσκευές.
- Εγγενώς περιλαμβάνει λιγότερο πολύπλοκες και πιο εύκολες στη διαχείριση συσκευές hardware.

Ωστόσο, προκειμένου τα παραπάνω πλεονεκτήματα να ισχύουν, πρέπει να αντιμετωπιστούν ορισμένες προκλήσεις που παρουσιάζει η σχεδίαση της αρχιτεκτονικής ενός συστήματος edge computing. Ακολουθούν οι σημαντικότερες από αυτές μαζί με προτάσεις για το πώς αυτές μπορούν να αντιμετωπιστούν [10]:

- Η επιλογή μιας συσκευής edge computing είναι κρίσιμη σε διάφορα σενάρια δικτύων. Για παράδειγμα, στα δίκτυα τύπου VANET (Vehicular Adhoc NETwoks) η συσκευή edge computing μπορεί να είναι ένα όχημα ή ένας ειδικά αφιερωμένος edge server. Αν τα οχήματα επιλεχθούν ως edge συσκευές, οι υπολογισμοί γίνονται κατανεμημένοι ωστόσο το κόστος της υλοποίησης θα είναι μεγάλο. Από την άλλη, αν το δίκτυο έχει έναν ειδικά αφιερωμένο edge server, μπορεί να αντιμετωπίσει προκλήσεις στη διαχείριση των αυξανόμενων απαιτήσεων των τελικών συσκευών. Γι' αυτό για να έχει ένα αποδοτικό σύστημα edge computing, η εφαρμογή θα πρέπει να ενσωματώσει ένα αποδοτικό σχήμα διαχείρισης πόρων το οποίο θα πρέπει να έχει την ευχέρεια να διαχειρίζεται τόσο edge servers όσο και τις συνδεδεμένες συσκευές.
- Ο διαμοιρασμός του υπολογιστικού φόρτου στις edge συσκευές είναι μία ακόμα πρόκληση. Σε ένα δυναμικό δίκτυο οι υπολογισμοί που αφορούν τους διάφορους edge κόμβους χρειάζεται να διαμοιράζονται με έναν κατανεμημένο τρόπο. Χωρίς ένα κατανεμημένο σχήμα, ο φόρτος εργασίας γίνεται μεροληπτικός, το οποίο εν τέλει αυξάνει το φορτίο σε ορισμένα συστήματα και εξαντλεί την μπαταρία τους. Απαιτείται προσεκτική εφαρμογή πολιτικών συνδυασμένη με αποδοτική οργάνωση των υπολογισμών και της διαχείρισης προκειμένου να υπάρχει ένα ενεργειακά αποδοτικό σύστημα κατανομής του φόρτου εργασίας.
- Η αυτοματοποιημένη κατανομή των εργασιών μεταξύ του cloud και του edge είναι απαιτητική. Λόγω συγκεκριμένων τεχνολογικών περιορισμών από την άποψη της υπολογιστικής ισχύος και του αποθηκευτικού χώρου, το edge computing δεν αποκλείει εξολοκλήρου τις υπηρεσίες του cloud computing, καθώς μερικοί υπολογισμοί γίνονται ακόμα σε cloud servers για την αύξηση της αξιοπιστίας του συστήματος. Χρειάζεται να ενσωματωθεί στο edge computing ένας αξιόπιστος μηχανισμός προγραμματισμού των εργασιών, ο οποίος θα κατανέμει κατάλληλα τις εργασίας στους edge και τους cloud servers χωρίς να επηρεάζεται η απόδοση του συστήματος.
- Η μείωση των καθυστερήσεων λόγω της επικοινωνίας για να επιτευχθεί η ζητούμενη ποιότητα υπηρεσίας στο edge computing είναι επίσης απαιτητική. Χωρίς καμία προτυποποίηση και πρωτόκολλα των δικτύων, τα συστήματα edge computing μπορεί να υποφέρουν από προβλήματα σχετικά με το δίκτυο, όπως συμφόρηση της δικτυακής κίνησης και άρνηση υπηρεσίας. Χρειάζεται να σχεδιαστεί ένα αποδοτικό πρωτόκολλα και τα κατάλληλα standards για τα συστήματα edge computing για τη διασφάλιση της ομαλής λειτουργίας χωρίς καθυστερήσεις στο δίκτυο.
- Η διαχείριση της κινητικότητας στο edge computing αποτελεί μία ακόμα πρόκληση. Οι συσκευές που χρησιμοποιούνται στα δίκτυα υψηλής κινητικότητας όπως τα MANETs και τα VANETs αντιμετωπίζουν συχνά αποσύνδεση στην επικοινωνία. Αυτό έχει ως αποτέλεσμα η επεξεργασία των δεδομένων και η λήψη των αποφάσεων να μπορεί να επηρεαστεί και να καθυστερήσει σημαντικά. Θα πρέπει να

ενσωματωθεί στις συσκευές edge computing ένα αξιόπιστο σχήμα συνεργασίας για να χειρίζονται αποδοτικά τέτοια θέματα στην κινητικότητα.

- Η διασφάλιση της ασφάλειας και της ιδιωτικότητας σε ένα σύστημα edge computing μπορεί να είναι αρκετά απαιτητική. Καθώς οι υπολογισμοί μεταφέρονται στα άκρα του δικτύου, οι πληροφορίες γίνονται ευάλωτες σε διάφορες απειλές ασφαλείας και επιθέσεις. Χρειάζεται να ενσωματωθούν στο σύστημα αποδοτικά σχήματα ταυτοποίησης και συστήματα διαχείρισης της εμπιστοσύνης για να χειρίζονται θέματα ασφαλείας και να αντικρούουν πιθανές κακόβουλες εισβολές/επιθέσεις.

1.7 Σύγκριση μεταξύ Cloud, Fog και Edge Computing

Στην τελευταία ενότητα αυτού του κεφαλαίου θα συγκρίνουμε με βάση κάποια χαρακτηριστικά τους τις τεχνολογίες του cloud, του fog και του edge computing. Στον πίνακα 2 της επόμενης σελίδας παρουσιάζονται συγκριτικά διάφορα υπολογιστικά χαρακτηριστικά των τριών αυτών τεχνολογιών. Απ' ότι φαίνεται το edge computing υπερισχύει έναντι του cloud, ενώ παρουσιάζει σχεδόν ίδια συμπεριφορά με το fog computing, υπερισχύοντας σε δύο μόνο χαρακτηριστικά.

Χαρακτηριστικό	**Cloud Computing**	**Fog Computing**	**Edge Computing**
Καθυστέρηση	Υψηλή	Χαμηλή	Χαμηλή
Χρησιμοποίηση εύρους ζώνης	Υψηλή	Χαμηλή	Πολύ χαμηλή
Χρόνος απόκρισης	Υψηλός	Χαμηλός	Χαμηλός
Αποθηκευτικός Χώρος	Υψηλός	Χαμηλός	Χαμηλός
Επιβάρυνση server	Πολύ υψηλή	Χαμηλή	Πολύ χαμηλή
Κατανάλωση ενέργειας	Υψηλή	Χαμηλή	Χαμηλή
Συμφόρηση δικτύου	Πολύ υψηλή	Χαμηλή	Χαμηλή
Κλιμακωσιμότητα	Μέτρια	Υψηλή	Υψηλή
Ποιότητα Υπηρεσίας (QoS) και Ποιότητα Εμπειρίας (QoE)	Μέτρια	Υψηλή	Υψηλή

Πίνακας 2: Σύγκριση διαφόρων χαρακτηριστικών των cloud, fog και edge computing

Μια λεπτομερής συγκριτική ανάλυση ανάμεσα στις τρεις παραπάνω αρχιτεκτονικές σε ό,τι αφορά υπολογιστικές υπηρεσίες και υπηρεσίες αποθήκευσης φαίνεται στον πίνακα 3 παρακάτω:

Χαρακτηριστικό	**Edge Computing**	**Fog Computing**	**Cloud Computing**
Υπολογιστικές Υπηρεσίες	Χρόνος απόκρισης σε χιλιοστά του δευτερολέπτου	Χρόνος απόκρισης από δευτερόλεπτα μέχρι λεπτά ανάλογα την εφαρμογή	Χρόνος απόκρισης σε λεπτά
Υπηρεσίες Αποθήκευσης	Προσωρινή αποθήκευση, δεν υποστηρίζει μεγάλη συλλογή δεδομένων	Τα δεδομένα μπορούν να αποθηκευτούν από ώρες μέχρι κάποιες μέρες	Μόνιμη αποθήκευσης, υποστηρίζει τεράστια συλλογή δεδομένων

Πίνακας 3: Σύγκριση των cloud, fog και edge computing με βάση υπολογιστικές και υπηρεσίες αποθήκευσης

Στο edge computing, ο χρόνος απόκρισης των υπολογιστικών υπηρεσιών είναι σε χιλιοστά του δευτερολέπτου και υποστηρίζει σχήματα εφαρμογής ως υπηρεσίας (Application as a Service - AaaS). Το edge computing μπορεί να εκτελέσει αποδοτικά ανάλυση δεδομένων, διάφορες προγνωστικές αναλύσεις και εικονικοποίηση στους edge servers. Στηριζόμενο στη χαμηλή του καθυστέρηση το edge computing επιτρέπει πανταχού παρούσες υπολογιστικές διαδικασίες σε έξυπνες εφαρμογές, όπου ο χρήστης μπορεί να έχει διάδραση με το σύστημα σε πραγματικό χρόνο και να έχει μια καλύτερη ποιότητα εμπειρίας. Σε ό,τι αφορά τις υπηρεσίες αποθήκευσης, το edge computing τις υποστηρίζει τοπικά και κρατάει τα δεδομένα στον server μόνο για ένα μεταβατικό χρονικό διάστημα, μιας και χρησιμοποιεί τους αποθηκευτικούς πόρους περισσότερο για cache χρήση παρά για αποθήκευση. Μιας και η χωρητικότητα αποθήκευσης στο edge computing είναι περιορισμένη, οι υπηρεσίες αποθήκευσης του edge computing δεν μπορούν να διαχειριστούν μεγάλες επιχειρησιακές εφαρμογές που χειρίζονται τεράστιες ποσότητες από δεδομένα.

Οι υπολογιστικές υπηρεσίες του fog computing μοιράζονται τα ίδια πλεονεκτήματα με το edge computing. Καθώς οι υπολογιστικές δυνατότητες μετακινούνται σε επίπεδο LAN παρέχονται επίσης κάποιες επιπλέον υπηρεσίες όπως συνεργασία ως υπηρεσία (Cooperation as a Service - CaaS) και δίκτυο ως υπηρεσία (Network as a Service - NaaS) στους τελικούς χρήστες. Η υπηρεσία αποθήκευσης στο fog computing παρέχει αποθηκευτικό χώρο για μικρό χρονικό διάστημα. Καθώς οι fog κόμβοι βρίσκονται στα άκρα του δικτύου LAN, υπάρχει μεγαλύτερος αποθηκευτικός χώρος απ' ότι στο edge computing. Ανάλογα με την παραμετροποίηση αποθήκευσης του fog computing server τα δεδομένα μπορεί να παραμένουν σε αυτόν από κάποιες ώρες μέχρι μέρες.

Οι υπολογιστικές υπηρεσίες στο cloud computing μπορούν να κατηγοριοποιηθούν ευρέως ως IaaS, PaaS και SaaS όπως έχουμε δει. Οι υπολογιστικές δυνατότητες είναι πολύ υψηλές λόγω της παρουσίας κεντρικοποιημένων κέντρων δεδομένων. Παρόλο που το cloud computing παρέχει τεράστια υπολογιστική ισχύ, η απόσταση ανάμεσα στον server και τον τελικό χρήστη καθιστά την διαδικασία ευάλωτη σε υψηλή καθυστέρηση, το οποίο δεν ταιριάζει με εφαρμογές επεξεργασίας πραγματικού χρόνου και IoT. Η υπηρεσία αποθήκευσης στο cloud computing είναι τόσο μεγάλη που επιτρέπει την αποθήκευση των δεδομένων σε μόνιμη βάση.

Κεφάλαιο 2ο
Internet of Things

Στο κεφάλαιο αυτό θα αναφερθούμε αρχικά στο Internet of Things και τις εφαρμογές που βρίσκει στις μέρες μας. Στη συνέχεια θα μιλήσουμε για το πώς συσχετίζεται η αρχιτεκτονική του IoT με το edge και το cloud. Στο τελευταίο κομμάτι του κεφαλαίου θα περιγράψουμε αναλυτικά κάποιες δημοφιλείς open source πλατφόρμες IoT παρουσιάζοντας και διάφορα κριτήρια τα οποία μπορεί να χρησιμοποιήσει κανείς για να επιλέξει την πιο κατάλληλη πλατφόρμα για την εφαρμογή του.

2.1 Εφαρμογές του Internet of Things (IoT)

Στις μέρες μας όλο και περισσότερο γίνεται συζήτηση και όχι μόνο στους κόλπους της πληροφορικής σχετικά με το Internet of Things, τις πολυάριθμες εφαρμογές που βρίσκει στη ζωή μας και πώς αυτές διαμορφώνουν ένα νέο τοπίο στην κοινωνία του σήμερα και ακόμα περισσότερο του αύριο. Ξεκινώντας με έναν ορισμό μπορούμε να πούμε ότι **Internet of Things (IoT) ή Διαδίκτυο των Πραγμάτων** είναι το δίκτυο που αποτελείται από βιομηχανικές μηχανές μέχρι wearable συσκευές που χρησιμοποιούν ενσωματωμένους αισθητήρες για τη συλλογή και μετάδοση δεδομένων ή την ανάληψη κάποιας δράσης βάσει αυτών. Οι συσκευές αυτές είναι συνδεδεμένες στο ίντερνετ, συλλέγουν, ανταλλάσσουν πληροφορίες και επικοινωνούν μεταξύ τους. Ιστορικά ο όρος *Internet of Things* επινοήθηκε στα τέλη της δεκαετίας του 90΄ και πιο συγκεκριμένα το 1999 από τον επιχειρηματία Kevin Ashton, έναν από τους ιδρυτές του auto-id center στο MIT. Ο Kevin Ashton αποτελεί μέλος της ομάδας που ανακάλυψε τον τρόπο σύνδεσης των συσκευών με το διαδίκτυο. Κατά την παρουσίαση του επιτεύγματος αυτού αναφέρθηκε για πρώτη φορά ο όρος *Internet of Things* και από τότε καθιερώθηκε η χρήση του. [11]

Εικόνα 6: Διάφοροι τομείς στους οποίους το Internet of Things βρίσκει εφαρμογή.

Η γρήγορη διάδοση του IoT οφείλεται στο ότι όλο και περισσότερες συσκευές διαθέτουν σύστημα Wi-Fi ή Bluetooth καθώς και προηγμένους αισθητήρες. Για παράδειγμα, σε ένα σπίτι με “έξυπνες” συσκευές όπως το ψυγείο, η καφετιέρα, η κουζίνα και ο θερμοστάτης μέσω της σύνδεσης τους στο διαδίκτυο μπορούν να ελεγχθούν από απόσταση χρησιμοποιώντας το smartphone ή τον ηλεκτρονικό υπολογιστή.

Το Internet of Things θα φέρει σημαντικές αλλαγές στον τρόπο διοίκησης και λειτουργίας των επιχειρήσεων και οργανισμών. Ενδεικτικά, σε μελέτη που πραγματοποιήθηκε από την εταιρία ερευνών Gartner προβλέπεται ότι το IoT θα φέρει κέρδη αξίας 2 τρισεκατομμυρίων δολαρίων μέχρι το 2020 και ο αριθμός των συνδεδεμένων συσκευών θα φτάσει τον αριθμό των 26 δισεκατομμυρίων! Ο όγκος των πληροφοριών που θα προκύψουν εκτιμάται πως θα είναι 14 φορές μεγαλύτερος από τον όγκο των δεδομένων που παράγονται σε ετήσια βάση.

Το IoT αποτελεί μια τεχνολογία που θα αλλάξει τις ισορροπίες στο μέλλον, θα φέρει δραματικές αλλαγές στις αγορές, τις υπηρεσίες υγείας και στη βιομηχανία. Τα δεδομένα που θα συγκεντρώνονται θα μπορούν να χρησιμοποιηθούν για τη βελτίωση της απόδοσης, τον εντοπισμό και την πρόβλεψη των αναγκών των ανθρώπων και των οργανισμών πριν αυτές εκδηλωθούν. Χαρακτηριστική είναι η περίπτωση της πόλης του Όσλο, όπου με τέτοιου είδους έξυπνες λύσεις επιτεύχθηκε η μείωση του ενεργειακού κόστους κατά 62%![13]

Το Internet of Things αποτελεί κάτι περισσότερο από μια διευκόλυνση για τους καταναλωτές, δεδομένου ότι δημιουργεί νέες πηγές πληροφοριών, νέα επιχειρηματικά μοντέλα, νέες υπηρεσίες και νέα καινοτόμα προϊόντα σε πολλούς κλάδους. Ενδεικτικά κάποιοι κλάδοι που θα επηρεαστούν άμεσα είναι :

- **Υγειονομική περίθαλψη & υπηρεσίες υγείας**

Πολλοί άνθρωποι παγκοσμίως ήδη χρησιμοποιούν smartwatches ή άλλες έξυπνες συσκευές για να παρακολουθούν την κατάσταση της υγείας τους. Εκτιμάται πως στο άμεσο μέλλον στα νοσοκομεία η παρακολούθηση των ασθενών θα γίνεται μέσω μόνιτορ τα οποία θα είναι συνδεδεμένα σε ένα δίκτυο IoT. Με αυτόν τον τρόπο θα βελτιωθούν οι υπηρεσίες υγείας και υγειονομικής περίθαλψης, θα να εξοικονομηθεί χρόνος και θα μειωθεί δραματικά το κόστος περίθαλψης των ασθενών.

- **Μεταφορές**

Κάποτε τα αυτοκινούμενα αυτοκίνητα φάνταζαν σενάριο επιστημονικής φαντασίας. Τα τελευταία χρόνια εταιρίες όπως η Tesla Motors, η BMW και η Volvo έχουν δημιουργήσει αυτοκίνητα τα οποία κινούνται αυτόνομα ή με την επίβλεψη του ανθρώπου. Αξιοποιώντας τεχνολογίες IoT τα αυτοκίνητα είναι συνδεδεμένα στο διαδίκτυο συλλέγοντας πληροφορίες για τη διαδρομή, την κίνηση στους δρόμους, τις κλιματολογικές συνθήκες και την κατάσταση του οδοστρώματος. Στη συνέχεια αναλύουν και επεξεργάζονται τις πληροφορίες και έτσι μπορούν να αποφασίσουν ποια είναι η κατάλληλη ταχύτητα και βέλτιστη διαδρομή που πρέπει να ακολουθηθεί. Το 2016 η εταιρία μεταφορών Uber πρόσθεσε στον αυτοκινητιστικό της στόλο τα πρώτα αυτοκινούμενα αυτοκίνητα τα οποία θα προσφέρουν υπηρεσίες μεταφορών οι οποίες θα κοστίζουν λιγότερο.

- **Λιανικό Εμπόριο**

Τόσο οι καταναλωτές όσο και τα καταστήματα μπορούν να επωφεληθούν από τις λειτουργίες IoT στο εμπόριο. Οι επιχειρήσεις θα μπορούν να παρακολουθούν τα αποθέματα τους αυτοματοποιημένα και θα ειδοποιούνται σε πραγματικό χρόνο για την πορεία λειτουργιών ή επιχειρησιακών διαδικασιών. Θα αυξηθεί η ροή πληροφοριών που θα έχει στη διάθεση της η επιχείρηση με τις εφαρμογές mobile wallet να έρχονται και να προστίθονται με την σειρά τους στα εργαλεία της εταιρίας που βασίζονται στην τεχνολογία IoT. Επίσης, μέσω χρήσης αισθητήρων και των συστημάτων παρακολούθησης θα γνωρίζει ανά πάσα ώρα και στιγμή πόσοι καταναλωτές βρίσκονται στο κατάστημα αλλά και τι ενέργειες πραγματοποιούν. Βασισμένη στο IoT έχει αναπτυχθεί τεχνολογία η οποία ανιχνεύει και ταυτόχρονα αναλύει τις ανθρώπινες εκφράσεις. Έτσι η επιχείρηση μπορεί να εξάγει πληροφορίες σχετικά με τις προτιμήσεις των καταναλωτών βασισμένες στις εκφράσεις και τις συνήθειες τους. Εκμεταλλευόμενοι αυτή την τεχνολογία η διαφήμιση και η προώθηση των προϊόντων θα γίνεται πιο αποτελεσματικά και άμεσα. Από τη μεριά των καταναλωτών οι αγορές θα γίνουν γρηγορότερες αφού οι νέοι τρόποι πληρωμής που θα ενταχθούν θα κάνουν τις πληρωμές αυτοματοποιημένες. Επιπρόσθετα, μέσω των πληροφοριών που θα έχουν συλλέξει οι εταιρίες θα μπορούν να προσφέρουν εξατομικευμένα προϊόντα και υπηρεσίες αυξάνοντας κατακόρυφα την ικανοποίηση των πελατών τους.

Συνοψίζοντας, μερικές ακόμα πτυχές της κοινωνίας μας όπου το Internet of Things ενδέχεται να γνωρίσει ταχεία άνθιση είναι οι “έξυπνες” πόλεις (smart cities), τα “έξυπνα” σπίτια (smart homes), τα wearables, η γεωργία, οι βιομηχανικοί αυτοματισμοί και η διαχείριση ενέργειας.

Η μεγαλύτερη πρόκληση που θα κληθούν να αντιμετωπίσουν οι εταιρίες είναι η ασφάλεια. Σκεφτείτε ένα τεράστιο δίκτυο συνδεδεμένων συσκευών οι οποίες ανταλλάσσουν, επεξεργάζονται και συλλέγουν πληροφορίες. Όλος αυτός ο όγκος θα πρέπει να αρχειοθετηθεί και να ταξινομηθεί με ασφάλεια. Για παράδειγμα, αν έχουμε αισθητήρες για να παρακολουθούμε την υγεία ενός ασθενή, θα πρέπει να διασφαλίσουμε ότι αυτά τα δεδομένα θα μείνουν ασφαλή και δεν θα πέσουν στα χέρια λάθος ανθρώπων. Εξίσου σημαντική αποτελεί η διαδικασία εύρεσης αξιόπιστων και ενεργειακά αποδοτικών τρόπων αποθήκευσης και ανάλυσης των δεδομένων που θα παράγουν ταυτόχρονα δισεκατομμύρια συσκευές.

2.2 Η ενοποίηση της IoT αρχιτεκτονικής με το Cloud και το Edge

Τόσο το cloud όσο και το edge computing παίζουν πλέον σημαντικό ρόλο στον τομέα του Internet of Things ανάλογα την περίπτωση. Σε αυτή την ενότητα θα συνοψίσουμε τους σημαντικότερους λόγους για τους οποίους συμβαίνει αυτό.

Το Internet of Things χαρακτηρίζεται από συσκευές κατανεμημένες σε μεγάλος εύρος, οι οποίες έχουν περιορισμένες επεξεργαστικές δυνατότητες και σχετικά περιορισμένο αποθηκευτικό χώρο. Οι συσκευές αυτές συχνά αντιμετωπίζουν ζητήματα που σχετίζονται με την απόδοση, την αξιοπιστία, την ιδιωτικότητα και την ασφάλεια. [14]

Το cloud computing συνιστά ένα τεράστιο δίκτυο με σχεδόν απεριόριστες αποθηκευτικές δυνατότητες και υπολογιστική ισχύ. Παρέχει ένα ευέλικτο και ισχυρό περιβάλλον το οποίο επιτρέπει τη δυναμική ενσωμάτωση δεδομένων από ποικίλες πηγές. Έτσι, λοιπόν, μπορεί να επιλύσει μερικώς τα περισσότερα από τα παραπάνω ζητήματα. Πράγματι το IoT και το cloud είναι δύο τεχνολογίες οι οποίες συνδυάζονται για να αλλάξουν το τρέχον και το μελλοντικό περιβάλλον των διασυνδεδεμένων υπηρεσιών.

Με την ενοποίηση του cloud computing και του IoT δημιουργείται μια πλατφόρμα, το Internet of Things βασισμένο στο Cloud (The Cloud-based Internet of Things), η οποία επιτρέπει την έξυπνη χρήση εφαρμογών, πληροφοριών και υποδομών με έναν οικονομικά αποδοτικό τρόπο. Αυτό συμβαίνει γιατί ενώ το IoT και το cloud computing διαφέρουν το ένα από το άλλο, τα χαρακτηριστικά τους είναι σχεδόν συμπληρωματικά.

Ορισμένα από τα σημαντικότερα οφέλη της ενοποίησης του cloud computing και του IoT αναφέρονται παρακάτω:

1. **Επικοινωνία**

   Ο διαμοιρασμός εφαρμογών και δεδομένων είναι δύο σημαντικά χαρακτηριστικά του Cloud-based Internet of Things. Αναρίθμητες εφαρμογές μπορούν να μεταδοθούν μέσω του IoT, ενώ η αυτοματοποίηση μπορεί να χρησιμοποιηθεί για να διευκολύνει την χαμηλού κόστους διασπορά και συλλογή δεδομένων. Το cloud είναι μια αποδοτική και οικονομική λύση, η οποία μπορεί να χρησιμοποιηθεί για τη σύνδεση, τη διαχείριση και τον εντοπισμό των συστατικών στοιχείων του IoT χρησιμοποιώντας ενσωματωμένες εφαρμογές και προσαρμοσμένα portal.

2. **Αποθηκευτικός χώρος**

   Το IoT περιλαμβάνει δισεκατομμύρια συσκευών, συγκεντρώνοντας έναν τεράστιο αριθμό από πηγές πληροφοριών, οι οποίες δημιουργούν έναν τεράστιο όγκο από ημιδομημένα ή και μερικές φορές καθόλου δομημένα δεδομένα. Το cloud θεωρείται μια από τις πιο οικονομικά αποδοτικές και ταιριαστές λύσεις σε ό,τι αφορά τη διαχείριση τεράστιων ποσοτήτων δεδομένων που παράγονται από το IoT. Επιπλέον, δημιουργεί νέες ευκαιρίες για ενσωμάτωση και ομαδοποίηση δεδομένων αλλά και διαμοιρασμό με άλλους οργανισμούς.

3. **Επεξεργαστικές δυνατότητες**

   Όπως είπαμε, οι συσκευές του IoT χαρακτηρίζονται από περιορισμένες επεξεργαστικές δυνατότητες, οι οποίες αποτρέπουν συχνά την επιτόπου και πολύπλοκη επεξεργασία δεδομένων. Αντ' αυτού τα δεδομένα συλλέγονται και μεταφέρονται σε κόμβους που έχουν μεγαλύτερες επεξεργαστικές δυνατότητες, όπου υλοποιούνται η ομαδοποίηση και η επεξεργασία τους. Το cloud παρέχει απεριόριστες εικονικές επεξεργαστικές δυνατότητες και ένα μοντέλο χρήσης τους κατ' απαίτηση ανάλογα με τις εκάστοτε ανάγκες. Αλγόριθμοι πρόβλεψης και αποφάσεις με βάση τα δεδομένα μπορούν να ενσωματωθούν έτσι στο IoT, για την αύξηση των κερδών και την μείωση των κινδύνων με χαμηλότερο κόστος.

4. Νέες δυνατότητες

Το IoT χαρακτηρίζεται από την ετερογένεια των συσκευών του, των πρωτοκόλλων και των τεχνολογιών και γι' αυτό η αξιοπιστία, η κλιμακοσιμότητα, η διαλειτουργικότητα, η ασφάλεια, η διαθεσιμότητα και η αποδοτικότητα μπορεί να είναι δύσκολο να επιτευχθούν. Ενσωματώνοντας το IoT στο cloud επιλύει τα περισσότερα από αυτά τα ζητήματα, δημιουργώντας ταυτότοχρονα και νέα άλλα χαρακτηριστικά όπως η ευκολία στη χρήση και στην πρόσβαση με χαμηλά κόστη ανάπτυξης.

5. Νέα μοντέλα

Η ενσωμάτωση του IoT στο cloud καθιστά δυνατά νέα σενάρια για τα έξυπνα αντικείμενα, τις έξυπνες εφαρμογές και υπηρεσίες. Ορισμένα από αυτά είναι:

a. *SaaS (Sensing as a Service):* επιτρέπει την πρόσβαση σε δεδομένα αισθητήρων,
b. *EaaS (Ethernet as a Service):* προσφέρει ευρεία συνδεσιμότητα για τον έλεγχο απομακρυσμένων συσκευών,
c. *SAaaS (Sensing and Actuation as a Service):* παρέχει λογικό έλεγχο αυτόματα,
d. *IPMaaS (Identity and Policy Management as a Service):* παρέχει πρόσβαση στη διαχείριση πολιτικών και ταυτότητας,
e. *DBaaS (Database as a Service):* παρέχει ευρεία δυνατότητα διαχείριση βάσεων δεδομένων,
f. *SenaaS (Sensor as a Service):* παρέχει διαχείριση για απομακρυσμένους αισθητήρες,
g. *DaaS (Data as a Service):* παρέχει ευρεία πρόσβαση σε οποιονδήποτε τύπο δεδομένων.

Περιγράψαμε παραπάνω μερικούς από τους λόγους που επιβεβαιώνουν τη σημασία του cloud computing στον τομέα του Internet of Things στις μέρες μας. Ας δούμε τώρα και τους λόγους για τους οποίους σιγά σιγά γίνεται ολοένα και σημαντικότερο το edge computing στον τομέα αυτό [15]:

1. **Γρηγορότερη λήψη αποφάσεων**

   Τοποθετώντας την αναλυτική επεξεργασία στις πηγές δεδομένων, μπορεί κανείς να δράσει συγκεκριμένα σε μια ευρεία ποικιλία γεγονότων. Σε περιπτώσεις όπου ακόμα και μια καθυστέρηση λίγων λεπτών μπορεί να αποβεί επιζήμια, η άμεση απόκριση είναι καθοριστικής σημασίας. Οι στατιστικές αναλύσεις σε πραγματικό χρόνο στις πηγές των δεδομένων επιτρέπουν την απόκριση σε στενά χρονικά παράθυρα.

2. **Αντιμετώπιση χωρικών περιορισμών**

   Πολλές πηγές δεδομένων έχουν περιορισμούς σε χώρο, οι οποίοι μπορεί να καθιστούν αδύνατο το στήσιμο ενός server κανονικού μεγέθους. Ανάλογα με τις επεξεργαστικές απαιτήσεις μπορεί να είναι αδύνατο να τρέχουν servers σε ένα αυτοκίνητο ή σε ιατρικές συσκευές. Χρειάζεται ένα σύστημα το οποίο μπορεί να τρέξει αποδοτικά έχοντας μικρές ανάγκες σε hardware, όπως οι σημερινοί minicomputers.

3. **Αντιμετώπιση περιορισμών στο εύρος ζώνης (bandwidth)**
   Ορισμένες εφαρμογές του IoT, όπως οι πετρελαιοπηγές και τα συνδεδεμένα αυτοκίνητα, παράγουν σημαντικές ποσότητες δεδομένων που ξεπερνούν το εύρος ζώνης που παρέχουν τα σημερινά δίκτυα του internet. Αυτό σημαίνει ότι το να μεταφέρεις όλα τα δεδομένα σε ένα κεντρικό σημείο για ανάλυση δεν είναι πρακτικό. Με την βοήθεια του edge computing μπορεί κανείς να μειώσει τις απαιτήσεις σε bandwidth δίχως να στηρίζεται πλήρως στην μεταφορά όλων των δεδομένων σε ένα κεντρικό cluster για στατιστική ανάλυση. Μπορούν επίσης να εφαρμοστούν τακτικές όπως η σύνοψη, η υποδειγματοληψία ή η συμπίεση δεδομένων πριν αυτά μεταδοθούν σε ένα κεντρικό cluster στατιστικής ανάλυσης.
4. **Επιλεκτική επεξεργασία**
   Τα δεδομένα που συλλέγονται από τις πηγές στο edge είναι συχνά τεράστια σε όγκο αλλά δεν είναι όλα το ίδιο πολύτιμα. Με τη βοήθεια του edge computing, μπορεί κάποιος να απομονώσει τα σημαντικά δεδομένα για περαιτέρω ανάλυση, όπως και να μειώσει τις συνολικές ανάγκες σε αποθηκευτικό όγκο και σε μετάδοση δεδομένων. Για παράδειγμα, τα πιο ουσιώδη δεδομένα σε ένα test drive ενός αυτοκινούμενου οχήματος είναι αυτά που συλλέγονται για όση ώρα χρειάζεται να επεμβαίνει ο οδηγός του. Αυτά αναπαριστούν μια ανωμαλία η οποία πρέπει να αναλυθεί περισσότερο. Αντίθετα, τα υπόλοιπα δεδομένα που παράγονται όσο το όχημα κινείται δίχως προβλήματα δεν είναι τόσο χρήσιμα και άρα δεν χρειάζεται να αποσταλούν.
5. **Αντιμετώπιση περιορισμών από κανονιστικά νομικά πλαίσια**
   Αν κάποιος αντιμετωπίζει κανονισμούς ή άλλες αυστηρές πολιτικές σχετικά με το πού πρέπει να αποθηκεύονται τα δεδομένα, τότε αποκτά νόημα το να έχει αναλυτικές δυνατότητες στις πηγές δεδομένων. Έτσι, μπορεί να αξιοποιήσει την υπολογιστική ισχύ στο edge για ανωνυμοποίηση των προσωπικών πληροφοριών ταυτοποίησης, έτσι ώστε αυτές να μπορούν να μεταφερθούν σε μια άλλη τοποθεσία δίχως να παραβιάζονται τα ρυθμιστικά πλαίσια για τα προσωπικά δεδομένα.
6. **Αποφυγή υπερφόρτωσης του συστήματος από δεδομένα**
   Σε μερικές περιπτώσεις, όπως πχ φυσικές καταστροφές, μπορεί οι πηγές δεδομένων να παράξουν έναν τόσο μεγάλο όγκο δεδομένων που να υπερφορτώσει ολόκληρο το σύστημα IoT ενός οργανισμού. Υλοποιώντας στρατηγικές edge computing μπορεί κάποιος να περιορίσει την επίδραση μιας τέτοιας “καταιγίδας δεδομένων”, αφού γίνεται τοπική ανάλυση πριν αποσταλούν στο υπόλοιπο δίκτυο IoT.

2.3 Πλατφόρμες IoT

Στη συνέχεια θα πούμε λίγα λόγια σχετικά με τις πλατφόρμες που χρησιμοποιούνται στη βιομηχανία του IoT για την ανάπτυξη του λογισμικού, τη διασύνδεση, την ενσωμάτωση και τον έλεγχο των διαφόρων συσκευών που αποτελούν μέρος ενός project IoT. Σκοπός μας είναι να έχουμε μια πλατφόρμα που υποστηρίζει το dataflow programming, έτσι ώστε να είναι ευκολότερος ο προγραμματισμός και η διαχείριση. Με τον όρο dataflow programming περιγράφεται το στυλ προγραμματισμού που μοντελοποιεί ένα πρόγραμμα ως ένα συνδεδεμένο γράφο των δεδομένων που ρέουν μεταξύ των διαφόρων διαδικασιών και οντοτήτων. [16] Θα περιγράψουμε ορισμένες αρκετά δημοφιλείς πλατφόρμες ανοιχτού κώδικα (open source), όπως το Kura (+Apache Camel), το Kaa, το Flogo, το WoTKit και το Node-RED.

Η τεχνολογία του IoT καθιστά εύκολο το να συνδέει κανείς διαφόρων ειδών συσκευές στο δίκτυο και να αναπτύσσει εφαρμογές για να ελέγχει και να διαχειρίζεται αυτές τις συσκευές. Όλη η πολυπλοκότητα της παροχής συνδεσιμότητας, υπηρεσιών και του cloud για αυτές τις συσκευές είναι καθήκον μιας πλατφόρμας IoT.

Μια πλατφόρμα IoT διασφαλίζει την απρόσκοπτη ενοποίηση με διαφορετικό υλικό εξοπλισμό χρησιμοποιώντας μεγάλη ποικιλία από δημοφιλή πρωτόκολλα επικοινωνίας, εφαρμόζοντας διαφορετικούς τύπους τοπολογιών (άμεσης σύνδεσης ή μέσω πύλης (gateway)) και χρησιμοποιώντας SDKs όπου χρειάζεται. Χρησιμοποιώντας τις north-bound διεπαφές ενοποίησης που παρέχει μια πλατφόρμα, ο χρήστης μπορεί να μεταδώσει τα IoT δεδομένα που συλλέγονται σε συστήματα στατιστικής ανάλυσης, οπτικοποίησης δεδομένων ή αποθήκευσης δεδομένων, όπως επίσης και να μεταδώσει δεδομένα σε συνδεδεμένες συσκευές (παραμετροποίηση, ειδοποιήσεις) ή μεταξύ τους (έλεγχοι, γεγονότα) χρησιμοποιώντας διαφορετικά είδη εφαρμογών χρήστη.

Μια πλατφόρμα IoT αναφέρεται συχνά και ως *IoT middleware,* όρος που υπογραμμίζει το λειτουργικό της ρόλο ως ενδιάμεσου μεταξύ των επιπέδων του hardware και των εφαρμογών. Οι καλύτερες πλατφόρμες IoT είναι ικανές να ενοποιηθούν με σχεδόν οποιαδήποτε συνδεδεμένη συσκευή και να ταιριάξουν με τις εφαρμογές που χρησιμοποιούνται από αυτή. Αυτή η ανεξαρτησία από υποκείμενο hardware και από το υπερκείμενο software επιτρέπει σε μια πλατφόρμα IoT να υλοποιήσει χαρακτηριστικά του IoT σε οποιουδήποτε είδους συνδεδεμένη συσκευή με τον ίδιο ξεκάθαρο τρόπο. [17]

Σε ό,τι αφορά το IoT integration, χρησιμοποιούνται συχνά οι λεγόμενες *process engines*, όπως αυτές που αναφέραμε παραπάνω. Αυτές ενοποιούν και συνδέουν συσκευές hardware, αισθητήρες, online υπηρεσίες, APIs, κα. Πέραν του απλού μοντέλου ETL (Extract-Transform-Load) ενός dataflow, οι process engines προσφέρουν και άλλες δυνατότητες επί των δεδομένων, όπως χειρισμός λαθών, επαναδρομολόγηση, αναμονή, συνέχιση. Επιτρέπουν τόσο σύγχρονη όσο και ασύγχρονη επικοινωνία και προσφέρουν επεξεργασία σε πραγματικό χρόνο. Επιπλέον, υποστηρίζουν τη συνδεσιμότητα με δημοφιλείς τεχνολογίες και πρωτόκολλα IoT (MQTT, CoaP, REST, κα.), ενώ πολύ σημαντικό χαρακτηριστικό τους είναι ότι μπορούν να εγκατασταθούν σε edge συσκευές. Τέλος, προσφέρουν ένα web UI (user interface - διεπαφή χρήστη) για visual coding, testing και debugging. [18]

Μετά από αυτή τη γενική εισαγωγή για τις πλατφόρμες IoT παρακάτω περιγράφουμε τα κυριότερα χαρακτηριστικά των πλατφορμών τις οποίες αναφέραμε στην αρχή αυτής της ενότητας.

2.3.1 Eclipse Kura (+Apache Camel)

Το Kura αποτελεί περισσότερο ένα IoT Gateway, παρά ολοκληρωμένο IoT framework γι' αυτό θα αναφερθούμε σε αυτό σε συνδυασμό με το Apache Camel, το οποίο είναι μια ισχυρή πλατφόρμα ολοκλήρωσης βασισμένη σε γνωστά Enterprise Integration Patterns (Επιχειρησιακά Πρότυπα Ολοκλήρωσης).

Πρόκειται για ένα IoT project της Eclipse το οποίο παρέχει μια πλατφόρμα για τη δημιουργία IoT gateway. Επιτρέπει την απομακρυσμένη διαχείριση τέτοιων gateway και παρέχει μεγάλο εύρος από APIs επιτρέποντας στον χρήστη να γράψει και να αναπτύξει τη δική του εφαρμογή IoT.

Το Kura τρέχει πάνω από ένα Java Virtual Machine (JVM) και χρησιμοποιεί την τεχνολογία OSGi, ένα δυναμικών στοιχείων σύστημα για την Java, για να απλοποιήσει την διαδικασία γραφής επαναχρησιμοποιήσιμων κομματιών λογισμικού. Τα APIs του Kura προσφέρουν εύκολη πρόσβαση στο υποκείμενο hardware, όπως σειριακές θύρες, GPS, watchdogs, θύρες USB, GPIOs, κα. Παρέχει επίσης OSGi bundles (σύνολο από Java κλάσεις και άλλους πόρους) για να απλοποιήσει τη διαχείριση της παραμετροποίσης δικτύων, της επικοινωνίας με IoT servers και την απομακρυσμένη διαχείριση του gateway. [19]

Το Kura περιέχει τις ακόλουθες υπηρεσίες:

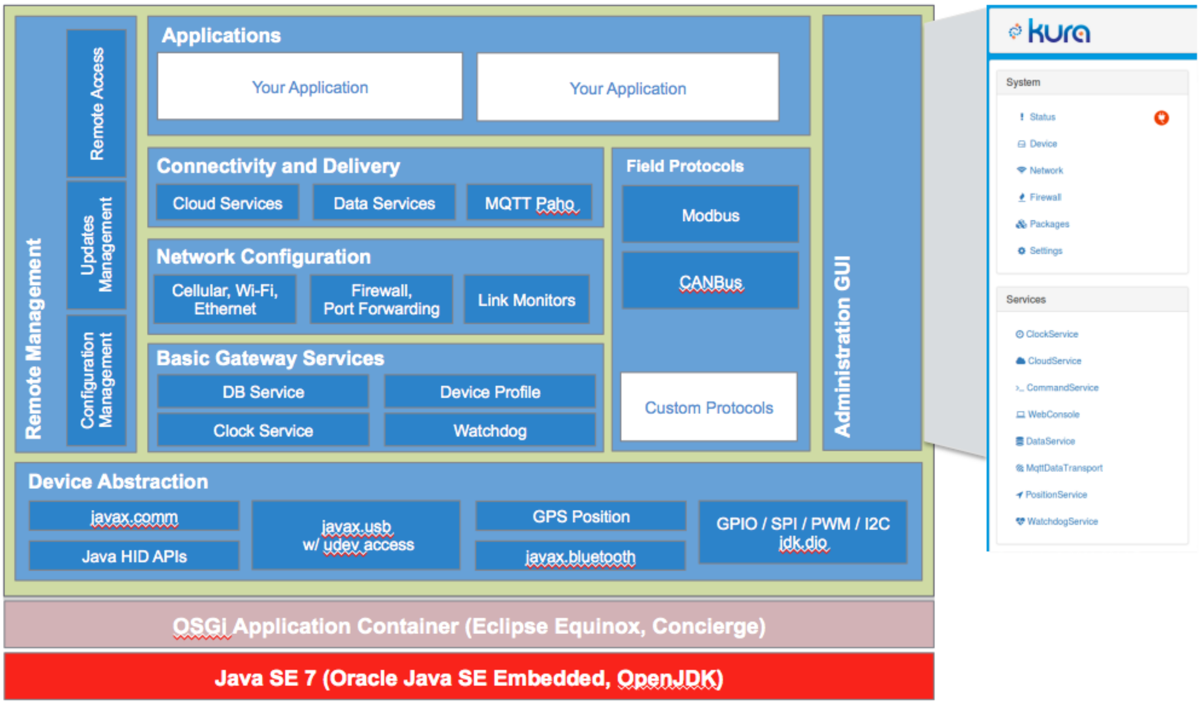
- **a.** Υπηρεσίες Εισόδου/Εξόδου (I/O)
  - **i.** Πρόσβαση σειριακής θύρας μέσω javax.comm 2.0 API ή σύνδεσης OSGi I/O
  - **ii.** Πρόσβαση θύρας USB μέσω javax.usb ή HID API
  - **iii.** Πρόσβαση Bluetooth μέσω javax.bluetooth ή σύνδεσης OSGi I/O
  - **iv.** Υπηρεσία τοποθεσίας για πληροφορίες GPS από NMEA stream
  - **v.** Υπηρεσία ρολογιού για το συγχρονισμό του συστήματος ρολογιού
  - **vi.** Kura API για πρόσβαση σε GPIO/PWM/I2C/SPI
- **b.** Υπηρεσίες δεδομένων
  - **i.** Λειτουργία αποθήκευσης και προώθησης δεδομένων τηλεμετρίας που συλλέγει το gateway και δημοσιεύονται στους απομακρυσμένους servers.
  - **ii.** Σύστημα δημοσιεύσεων βάσει πολιτικής, που αποκρύπτει από τον προγραμματιστή την πολυπλοκότητα του στρώματος δικτύου και του πρωτοκόλλου δημοσιεύσεων. Η προκαθορισμένη βιβλιοθήκη μηνυμάτων που χρησιμοποιείται παρέχεται από την εφαρμογή Eclipse Paho και τον πελάτη MQTT που χρησιμοποιεί.
- **c.** Υπηρεσίες Cloud
- **d.** Υπηρεσίες Παραμετροποίησης
- **e.** Υπηρεσία Απομακρυσμένης Διαχείρισης
  - **i.** Η υπηρεσία αυτή στηρίζεται στις υπηρεσίες Cloud και Παραμετροποίησης
- **f.** Υπηρεσία Δικτύωσης

i. Δυνατότητα παραμετροποίησης των διεπαφών δικτύου που είναι διαθέσιμα στο gateway όπως Ethernet και Wifi

g. Υπηρεσία Watchdog

h. Διεπαφή διαχείρισης μέσω Web

  i. Παρέχεται μια κονσόλα διαχείρισης του Gateway μέσω ενός web περιβάλλοντος



Εικόνα 7: Εποπτική παρουσίαση των συστατικών μερών του Kura

Σχετικά με το Apache Camel, μπορούμε να αναφέρουμε ότι επιτρέπει στο χρήστη να δημιουργεί τα Enterprise Integration Patterns για να υλοποιήσει κανόνες δρομολόγησης και παρέμβασης είτε μέσω μιας DSL (Domain Specific Language) γλώσσας βασισμένης στην Java, μέσω Xml αρχείων διαχείρισης βασισμένων σε Spring ή Blueprint ή μέσω της DSL γλώσσας Scala. Αυτό σημαίνει ότι ο χρήστης λαμβάνει έξυπνη ολοκλήρωση των κανόνων δρομολόγησης στο IDE του (Integrated Development Environment - Ολοκληρωμένο Περιβάλλον Ανάπτυξης) είτε χρησιμοποιεί Java, Scala ή XML editor.

Το Apache Camel χρησιμοποιεί URIs έτσι ώστε να μπορεί εύκολα να δουλέψει απευθείας με κάθε είδους μοντέλο μεταφοράς ή μηνυμάτων όπως HTTP, ActiveMQ, JMS, JBI, SCA, MINA ή CXF. Το Apache Camel είναι μια μικρή βιβλιοθήκη η οποία έχει ελάχιστες εξαρτήσεις ώστε να ενσωματώνεται εύκολα σε οποιαδήποτε Java εφαρμογή. [20]

Το Kura παρέχει δύο τρόπους ενοποίησης με το Camel, ως εφαρμογής και ως υπηρεσίας cloud. Η πρώτη επιτρέπει στο χρήστη να χειρίζεται το Camel για να παρέχει δεδομένα και να λαμβάνει εντολές από οποιαδήποτε παράδειγμα υπηρεσίας cloud που τρέχει στο Kura. Για παράδειγμα την προκαθορισμένη υπηρεσία cloud που υποστηρίζεται από το πρωτόκολλο MQTT. Η δεύτερη προσέγγιση επιτρέπει στο χρήστη να δημιουργήσει μια custom υλοποίηση υπηρεσίας cloud και να δρομολογήσει δεδομένα που έρχονται από άλλες εφαρμογές του Kura με τις διαδρομές που παρέχονται στο περιβάλλον του Camel. [21]

Κλείνοντας την ανάλυσή μας για το Kura, να αναφέρουμε ότι αποτελεί μια "ώριμη" πλατφόρμα μιας και ξεκίνησε να κυκλοφορεί τον Δεκέμβρη του 2013 και μπορεί να τρέξει πάνω ποικιλία από πλατφόρμες είτε τοπικά είτε στο cloud είτε σε edge συσκευές είτε ως container. Ωστόσο ένα βασικό του μειονέκτημα πέραν του ότι είναι σχετικά βαρύ και αργό σαν πρόγραμμα είναι ότι είναι αρκετά πολύπλοκο στο να το εγκαταστήσεις και να τρέξεις μια πρώτη IoT ροή δεδομένων αλλά και ότι χρειάζεται να γράφεις πηγαίο κώδικα για ό,τι χρειάζεται να φτιάξεις.

2.3.2 Kaa

Το Kaa είναι μια middeware πλατφόρμα πολλαπλού σκοπού για το Internet of Things η οποία επιτρέπει τη δημιουργία ολοκληρωμένων από άκρο σ' άκρο λύσεων IoT, συνδεδεμένων εφαρμογών και έξυπνων προϊόντων. Προσφέρει μία ανοιχτή, πλούσια σε χαρακτηριστικά εργαλειοθήκη για την ανάπτυξη προϊόντων IoT και γι' αυτό μειώνει δραματικά τα σχετικά κόστη, τα ρίσκα και τον χρόνο μέχρι την είσοδο του προϊόντος στην αγορά.

Το Kaa επιτρέπει μια σειρά σημαντικών IoT δυνατοτήτων για τις συσκευές όπως:

- Διαχείριση ενός πολύ μεγάλου αριθμού συσκευών
- Εγκαθίδρυση διαλειτουργικότητας μεταξύ των συσκευών
- Alpha και Beta testing των υπηρεσιών
- Απομακρυσμένη παραμετροποίηση των συσκευών
- Παρακολούθηση των συσκευών σε πραγματικό χρόνο
- Ασύρματη διανομή αναβαθμίσεων των προγραμμάτων λογισμικού
- Δημιουργία υπηρεσιών cloud για έξυπνα προϊόντα
- Συλλογή και ανάλυση δεδομένων από αισθητήρες
- Ανάλυση συμπεριφοράς χρηστών και παράδοση στοχευμένων ειδοποίησεων

Το Kaa καθιστά δυνατή τη διαχείριση δεδομένων από συνδεδεμένα αντικείμενα και back-end υποδομές παρέχοντας το server και τα SDKs για τα ακραία στοιχεία του συστήματος. Τα SDKs ενσωματώνονται στις συνδεδεμένες συσκευές και υλοποιούν σε πραγματικό χρόνο ανταλλαγή δεδομένων με τον server και προς τις δύο κατευθύνσεις. Τα SDKs του Kaa μπορούν να ενοποιηθούν σχεδόν με κάθε τύπο συνδεμένης συσκευής ή μικροτσίπ.

Εικόνα 8: Ο server και τα SDKs του Kaa

Ο server του Kaa παρέχει όλη την back-end λειτουργικότητα που χρειάζεται για να λειτουργήσουν ακόμα και μεγάλης κλίμακας, κρίσιμες IoT εφαρμογές. Χειρίζεται όλη την επικοινωνία σε όλο το εύρος των συνδεδεμένων αντικειμένων συμπεριλαμβανομένης της συνέπειας των δεδομένων, της ασφάλειας, της διαλειτουργικότητας των συσκευών και της συνδεσιμότητας χωρίς σφάλματα.

Ο server του Kaa παρέχει καλά εγκαθιδρυμένες διεπαφές για την ενοποίηση με συστήματα στατιστικής ανάλυσης και διαχείρισης δεδομένων όπως επίσης και με υπηρεσίες σχετικές με το προϊόν. Λειτουργεί ως το θεμέλιο του back-end συστήματος της εφαρμογής, το οποίο μπορεί να επεκταθεί και να προσαρμοστεί κατάλληλα για να ικανοποιεί τις συγκεκριμένες απαιτήσεις ενός προϊόντος. [22]

Η πλατφόρμα του Kaa παρέχει [23] μια σειρά από σημαντικά χαρακτηριστικά όπως:

- **Ενεργητική εξισορρόπηση φόρτου:** Υλοποιεί μια σειρά από στρατηγικές εξισορρόπησης φόρτου και της εφαρμόζει αυτόματα κατά το χρόνο εκτέλεσης για να επιτύχει περισσότερο ή λιγότερο ίσο φορτίο για κάθε κόμβο στο cluster του Kaa. Ακόμη, αυτό το χαρακτηριστικό εξασφαλίζει ότι τα ακραία στοιχεία ανακατευθύνονται αυτόματα σε άλλους κόμβους αν ο τρέχων κόμβος τους σταματήσει να λειτουργεί.
- **Βιβλιοθήκη κοινών τύπων:** Πρόκειται για μια αποθήκη των διαγραμμάτων των τύπων δεδομένων που χρησιμοποιούνται για όλες τις μονάδες του Kaa. Καθώς δημιουργούνται περισσότερα είδη διαγραμμάτων και εκδόσεις, αποθηκεύονται στη βιβλιοθήκη κοινών τύπων για μελλοντική χρήση.
- **Διαχείριση παραμετροποίησης:** Επιτρέπει στο χρήστη να δημιουργεί και να διανέμει δεδομένα παραμετροποίησης στα ακραία στοιχεία του συστήματος.
- **Συλλογή δεδομένων:** Επιτρέπει στο χρήστη να συλλέγει καταλόγους δεδομένων (logs) από τα ακραία στοιχεία του συστήματος, να τα αποθηκεύει σε συγκεκριμένα συστήματα ανάλυσης δεδομένων ή να τα αναλύει άμεσα.
- **Καταγραφή ακραίων στοιχείων:** Επιτρέπει την ασφαλή ταυτοποίηση και καταγραφή των ακραίων στοιχείων ενός συστήματος μέσα σε ένα Kaa cluster.
- **Ομάδες ακραίων στοιχείων:** Επιτρέπει την ομαδοποίηση ακραίων στοιχείων του συστήματος και στη συνέχεια την εφαρμογή άλλων χαρακτηριστικών του Kaa σε ολόκληρες τις ομάδες.

- **Ιδιοκτησία ακραίων στοιχείων:** Επιτρέπει το συσχετισμό ιδιοκτητών με συγκεκριμένα ακραία στοιχεία του συστήματος.
- **Προφίλ ακραίων στοιχείων:** Επιτρέπει την παροχή συγκεκριμένων ιδιοτήτων στα ακραία στοιχεία ενός συστήματος, οι οποίες στη συνέχεια μπορεί να χρησιμοποιηθούν για φιλτράρισμα και ομαδοποίηση των στοιχείων αυτών.
- **Γεγονότα:** Επιτρέπει τη δημιουργία προσαρμοσμένων γεγονότων και την αποστολή τους από ένα ακραίο στοιχείο του συστήματος σε ένα άλλο.
- **Ειδοποιήσεις:** Επιτρέπει τη μεταφορά μηνυμάτων και κλήσεων για δράση σε ακραία στοιχεία του συστήματος.

Κλείνοντας την περιγραφή μας για την πλατφόρμα Kaa θα αναφερθούμε σε ορισμένες δυνατότητες που παρέχει αυτή [24]:

- **Hardware:** Η πλατφόρμα του Kaa είναι ανεξάρτητη του hardware. Μπορεί να κανείς να ενοποιήσει με αυτή σχεδόν οποιαδήποτε συσκευή, ξεκινώντας από αυτές που υποστηρίζονται από πλήρως εξοπλισμένα λειτουργικά συστήματα και φτάνοντας μέχρι μικροελεγκτές με περιορισμένους πόρους με λιγότερη από 10KB μνήμη RAM διαθέσιμη. Το Kaa μπορεί ήδη να τρέξει πάνω από πολλά γνωστές πλατφόρμες, όπως windows, linux, android, raspberry pi, beaglebone, intel edison, κα.
- **Μοντέλα δεδομένων:** Το Kaa μπορεί να χειριστεί τόσο δομημένα όσο και αδόμητα δεδομένα, αλλά ευνοεί την πρώτη κατηγορία, μιας και τα καλά δομημένα δεδομένα παρέχουν εξαιρετικό υλικό για χειρισμό και στατιστική ανάλυση. Το Kaa διασφαλίζει ότι ο χρήστης μπορεί να διαχερίζεται και να λειτουργεί ακόμα και συσκευές βασισμένες σε διαφορετική τεχνολογία, εφόσον μοιράζονται το ίδιο σύνολο από διαγράμματα δεδομένων. Αυτά τα διαγράμματα δεδομένων είναι συμβατά με την τεχνολογία Apache Avro, η οποία επιτρέπει η σειριοποίηση δεδομένων να είναι πολύ αποδοτική. Το Kaa παρέχει επίσης και ένα GUI για διαγράμματα δεδομένων.
- **SDKs:** Τα SDKs του Kaa για τα ακραία στοιχεία ενός συστήματος είναι διαθέσιμα σε Java, C και C++ και είναι σχεδιασμένα ώστε να ενσωματώνονται στην εφαρμογή πελάτη. Χειρίζονται την επικοινωνία πελάτη-εξυπηρετητή, την ταυτοποίηση, την κρυπτογράφηση και άλλες λειτουργίες που παρέχονται από την πλατφόρμα. Κάθε SDK που παράγεται ταιριάζει με το συγκεκριμένο στιγμιότυπο του server. Επίσης, τα SDKs έχουν γνώση της υλοποίησης του πρωτοκόλλου επιπέδου μεταφοράς που χρησιμοποιείται, όπως ακόμα και πώς να εγκαθιδρύσουν επικοινωνία με τον server στον οποίο δημιουργήθηκαν τα SDKs αυτά. Είναι εύκολο μεταφέρσιμα σχεδόν σε οποιοδήποτε hardware ή λειτουργικό σύστημα.
- **Συνδεσιμότητα:** Το Kaa είναι ανεξάρτητο των πρωτοκόλλων μεταφοράς. Επιτρέπει τη δημιουργία εφαρμογών που δημιουργούν πάνω από οποιονδήποτε τρόπο δικτυακής σύνδεσης είτε συνεχούς είτε διακοπτόμενης. Μπορεί κανείς να χρησιμοποιήσει τις υπάρχουσες υλοποιήσεις των πρωτοκόλλων μεταφοράς που παρέχει το Kaa ή να δημιουργήσει προσαρμοσμένες και να τις φορτώσει στο σύστημα. Το Kaa θα διαφημίσει αυτόματα τις παραμέτρους της σύνδεσης στα ακραία στοιχεία του συστήματος και θα εξισορροπήσει το φόρτο μεταξύ των

servers του cluster. Προχωρημένοι χρήστες της πλατφόρμας θα βρουν χρήσιμο το ότι το Kaa μπορεί να αναθέσει ευέλικτα πρωτόκολλα επικοινωνίας στις υπηρεσίες της πλατφόρμας. Για παράδειγμα, οι ειδοποιήσεις μπορεί να μεταφέρονται μέσω UDP ή SMS ενώ οι παραμετροποιήσεις και τα δεδομένα των προφίλ μπορεί να συγχρονίζονται μέσω ενός AES κρυπτογραφημένου πρωτοκόλλου βασιζόμενου στο TCP.

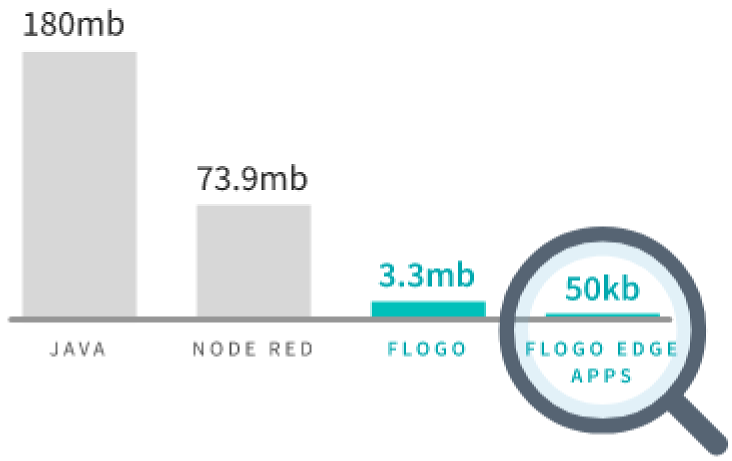
- **Gateways και routers IoT:** Το Kaa μπορεί να χρησιμοποιηθεί για να ενεργοποιήσει προηγμένα χαρακτηριστικά IoT σε gateways και Wi-fi routers. Με τις λειτουργίες του Kaa αυτές οι συσκευές μπορούν να λειτουργήσουν ως μια οικονομική αλλά με πολλαπλές λειτουργίες γέφυρα μεταξύ των ακραίων στοιχείων του συστήματος και του IoT cloud. Τα gateways και τα routers που είναι εξοπλισμένα με το Kaa μπορούν απλοποιήσουν τη συλλογή δεδομένων από αισθητήρες ή συσκευές που χρησιμοποιούν για τη συνδεσιμότητα πρωτόκολλα τύπου PAN, όπως Bluetooth, ZigBee ή Z-Wave. Για μεγάλους όγκους δεδομένων πραγματικού χρόνου ή για μεγάλης κλίμακας εφαρμογές, το Kaa επιτρέπει να γίνεται προκαταρκτική επεξεργασία και ομαδοποίηση των δεδομένω στο IoT gateway πριν στείλει τα δεδομένα στον Kaa server ή στο cloud. Έτσι, μειώνεται έως και δέκα φορές η κίνηση στο δίκτυο εξοικονομώντας πόρους.
- **Kaa server:** Ο server του Kaa προσφέρει μια σειρά από μηχανισμούς για εσωτερική προσαρμογή και ενοποίηση με εξωτερικά συστήματα: διεπαφές REST για ενοποίηση με υπηρεσίες σχετικές με το προϊόν, log appenders για ενοποίηση με συστήματα στατιστικής ανάλυσης, συστήματα οπτικοποίησης και διαχείρισης δεδομένων, κα. Ο server του Kaa εγκαθίσταται συνήθως ως ένα cluster από κόμβους, το οποίο είναι εύκολα επεκτάσιμο εφόσον χρειαστεί μεγαλύτερη απόδοση. Το cluster του Kaa χρησιμοποιεί το Apache Zookeeper για το συντονισμό των servers, την επιλογή των κύριων κόμβων και την αντιμετώπιση αποτυχιών.
- **Επεξεργασία δεδομένων:** Η συλλογή και η ανάλυση δεδομένων από συνδεδεμένες συσκευές είναι μία από τις πιο συχνές IoT περιπτώσεις χρήσεις. Η αποστολή του Kaa είναι να κάνει αυτές τις εργασίες εντελώς ξεκάθαρες ακόμα και σε τεράστια κλίμακα. Βασισμένος στο log σχεδιάγραμμα που ο χρήστης σχεδιάζει για την εφαρμογή, ο server του Kaa κατασκευάζει δεσμούς αντικειμένων και APIs για το προσαρμοσμένο με βάση την εφαρμογή SDK. Στη συνέχεια ο χρήστης χρησιμοποιεί αυτά τα APIs για να δηλώσει στο Kaa να μεταφέρει αρχεία log στο back-end σύστημα επεξεργασίας ή αποθήκευσης δεδομένων. Ένα ακραίο στοιχείο Kaa αποθηκεύει προσωρινά αρχεία log και τα στέλνει στο server μόλις δοθεί ένα σήμα για ανέβασμα. Τόσο η υλοποίηση της προσωρινής αποθήκευσης δεδομένων όσο και τα σήματα για ανέβασμα είναι προσαρμόσιμα. Στη μεριά του server, το Kaa λειτουργεί σαν "ελβετικός σουγιάς", υποστηρίζοντας μια πλατφόρμα από log appenders που ο χρήστης χρησιμοποιεί για να φορτώσει δεδομένα στη βάσης δεδομένων της αρεσκείας του, να στείλει αυτά τα δεδομένα για stream processing ή να τα διαθέσει σε προσαρμοσμένες μονάδες επεξεργασίας δεδομένων μέσω REST, Flume, κα.

2.3.3 Flogo

Το Flogo είναι μία πολύ ελαφριά πλατφόρμα ολοκλήρωσης η οποία στηρίζεται στη γλώσσα προγραμματισμού Go. Εντάσσεται και αυτό στην κατηγορία των ανοιχτού κώδικα πλατφορμών και είναι αρκετά εύκολο να επεκταθεί για τις εκάστοτε περιπτώσεις χρήσεις κάποιου χρήστη. Χρησιμοποιείται για την ανάπτυξη edge εφαρμογών IoT ή μικροϋπηρεσιών για το cloud ή για αρχιτεκτονικές δίχως server. Γι' αυτό μπορεί να λειτουργήσει συμπληρωματικά σε άλλες πλατφόρμες ολοκλήρωσης και σε IoT πλατφόρμες για το cloud.

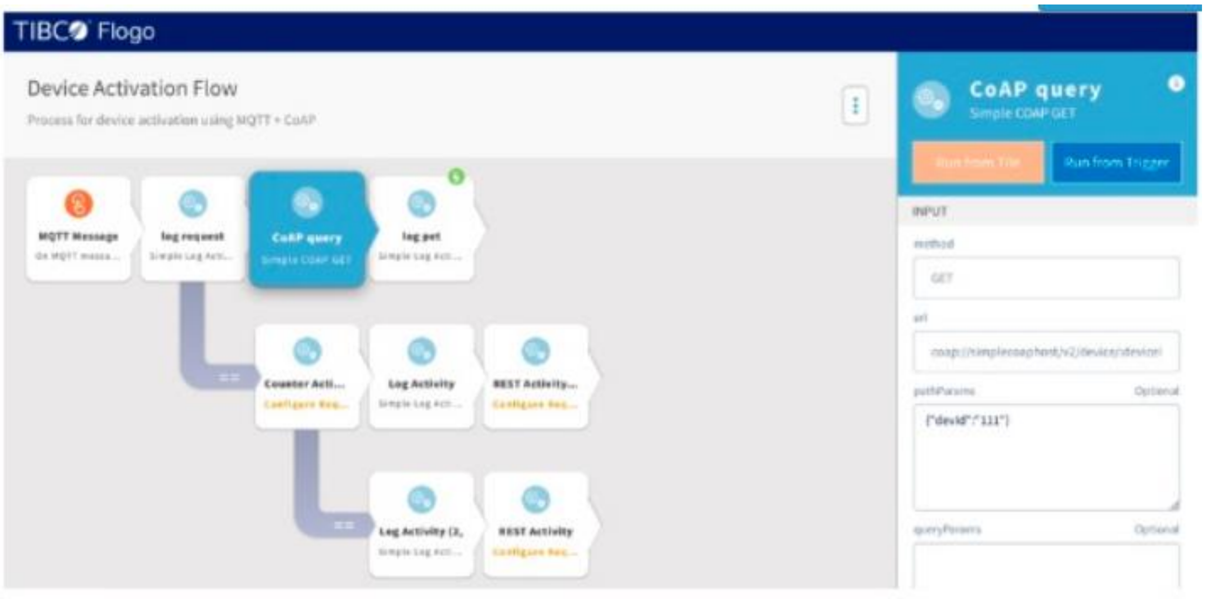
Ορισμένα από τα βασικά χαρακτηριστικά αυτής της σχετικά νέας πλατφόρμας IoT, μιας και κυκλοφόρησε στα τέλη του 2016, αναφέρονται παρακάτω [25]:

a. Είναι αρκετά εύκολη η διαδικασία εγκατάστασής της και δημιουργίας μιας πρώτης IoT ροής ολοκλήρωσης.
b. Έχει πολύ ελαφρύ αποτύπωμα για την εγκατάστασή της σε edge συσκευές με ένα μοντέλο μηδενικών εξαρτήσεων. Οι εφαρμογές της έχουν πολύ μικρό αποτύπωμα σε ό,τι αφορά το σκληρό δίσκο και τη μνήμη και πολύ γρήγορο χρόνο εκκίνησης.
c. Μπορεί να τρέξει πάνω από ποικίλες πλατφόρμες όπως edge συσκευές, edge gateways, τόσο σε τοπικές εγκαταστάσεις όσο και στο cloud, ως container, κτλ. Οι ροές που δημιουργούνται μπορούν να διαμοιραστούν ως JSON αρχεία. Επίσης, μπορούν να φορτωθούν σε άλλες συσκευές τα δυαδικά εκτελέσιμα που δημιουργούνται λόγω μηδενικών εξαρτήσεων.
d. Υποστηρίζει συνδεσιμότητα σε IoT τεχνολογίες όπως MQTT, CoaP και REST.
e. Είναι βελτιστοποιημένη ώστε να λειτουργεί και σε μη αξιόπιστα περιβάλλοντα IoT.
f. Απευθύνεται τόσο σε developers και ειδικούς της διαδικασίας ολοκλήρωσης όσο και σε μη ειδικούς είτε γράφοντας πηγαίο κώδικα είτε αξιοποιώντας το Web UI που παρέχει για visual coding, testing και debugging.
g. Περιλαμβάνει καινοτόμα χαρακτηριστικά όπως έναν web debugger με δυνατότητα βημάτων προς τα πίσω για τη διαδικασία διαδραστικού σχεδιασμού και αποσφαλμάτωσης των διαδικασιών, για προσομοίωση γεγονότων από αισθητήρες και για την αλλαγή δεδομένων ή παραμετροποιήσεων, δίχως να χρειάζεται η επανεκκίνηση όλης της διαδικασίας.



Εικόνα 9: Το αποτύπωμα του Flogo και των edge εφαρμογών του συγκριτικά με αντίστοιχες πλατφόρμες

Όπως φαίνεται από την παραπάνω εικόνα, η οποία προέρχεται από την αρχική σελίδα του project Flogo, το αποτύπωμα του είναι 20-50 φορές μικρότερο συγκριτικά με το node.js και τα jvm. Αυτό είναι πολύ σημαντικό στο σημερινό κόσμο των μικρών IoT συσκευών και των μοντέλων πληρωμής με βάση τη χρήση που επικρατούν στον τομέα των cloud υποδομών. [26]



Εικόνα 10: Παράδειγμα ενός flow από το Web UI του Flogo για την ενεργοποίηση μιας συσκευής

Κλείνοντας την αναφορά μας στην πλατφόρμα του Flogo, θα αναφερθούμε πιο αναλυτικά σε μερικές από τις έννοιες που συνδέονται με αυτή [27]:

- **App:** Μια εφαρμογή στα πλαίσια του Flogo αποτελείται από ένα ή περισσότερα triggers και flows (actions). Μια εφαρμογή καθεαυτή είναι στην πραγματικότητα ένας μηχανισμός οργάνωσης που μπορεί να αξιοποιηθεί όταν αναπτύσσονται μικροϋπηρεσίες και λειτουργίες. Μια εφαρμογή περιέχει ένα σύνολο από παραμετροποιήσεις, όπως επίσης triggers και μια συλλογή από flows.
- **Trigger:** Το Flogo είναι μια πλατφόρμα που βασίζεται σε γεγονότα. Ένα trigger είναι το σημείο εισόδου για γεγονότα. Ένα trigger μπορεί να είναι μια εγγραφή σε ένα MQTT topic, ένα Kafka topic, μια διεπαφή HTTP REST ή ένας συγκεκριμένος αισθητήρας IoT. Ένα trigger είναι υπεύθυνο για να αποδέχεται το εισερχόμενο γεγονός και να ξεκινά μία ή περισσότερες καθορισμένες actions (flows). Τα triggers δεν είναι συζευγμένα με τα flows, πράγμα που σημαίνει ότι ένα flow μπορεί να υπάρξει και χωρίς ένα trigger.
- **Handler:** Ο χειριστής ενός trigger χρησιμοποιείται για να αντιστοιχίσει τα triggers σε actions (flows) για επεξεργασία. Ένα trigger μπορεί να έχει έναν ή περισσότερους handlers που να δρομολογούν γεγονότα σε διαφορετικά flows.
- **Action:** Τα εισερχόμενα γεγονότα μπορούν να αντιστοιχιθούν σε ένα action. Ένα action είναι μια γενική υλοποίηση για την επεξεργασία εισερχομένων γεγονότων. Μπορούν να υλοποιηθούν διαφορετικού τύπου actions, με συνέπεια να καθορίζονται και διαφορετικές μέθοδοι με τις οποίες ένα εισερχόμενο event

μπορεί να επεξεργαστεί. Αυτή τη στιγμή, μόνο ένας τύπος action έχει υλοποιηθεί, το flow.

- **Flow:** Ένα flow είναι η υλοποίηση ενός action και είναι το πρωταρχικό εργαλείο για την υλοποίηση επιχειρησιακής λογικής στο Flogo. Ένα flow μπορεί να αποτελείται από έναν αριθμό διαφορετικών δομών:
    - Ένα ή περισσότερα activities που υλοποιούν συγκεκριμένη ενέργεια (για παράδειγμα μια εγγραφή σε μία βάση, η κλήση μιας REST διεπαφής, κα)
    - Κάθε activity συνδέεται μέσω ενός link
    - Τα links μπορεί να εμπεριέχουν υποθετική λογική για να αλλάξουν το μονοπάτι ενός flow

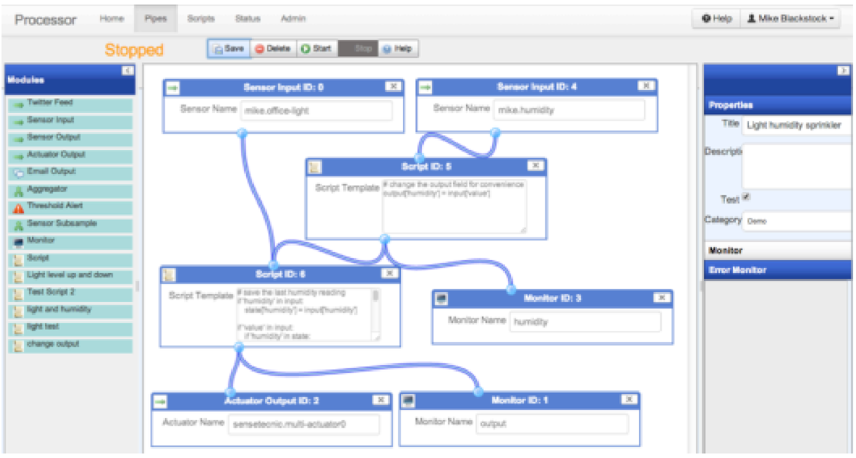
  Τα flows, όπως αναφέρθηκε παραπάνω στο κομμάτι των triggers, μπορεί να υπάρχουν χωρίς κάποιο trigger. Έτσι, τα flows λειτουργούν με τρόπο πολύ όμοιο με τις συναρτήσεις, δηλαδή ένα flow μπορεί να ορίζει τις δικές του παραμέτρους εισόδου και εξόδου, επιτρέποντας σε ένα flow να μπορεί να επαναχρησιμοποιηθεί ανεξάρτητα από το trigger εισόδου. Όλη η λογική σε ένα flow μπορεί να λειτουργήσει μόνο απέναντι στα ακόλουθα δεδομένα:
    - Παραμέτρους εισόδου του flow
    - Μεταβλητές περιβάλλοντος
    - Ιδιότητες εφαρμογής
    - Τα δεδομένα εξόδου από activities που αναφέρονται μέσα στο flow

  Ένα flow δεν μπορεί να έχει πρόσβαση απευθείας στα δεδομένα ενός trigger. Τα δεδομένα εισόδου και εξόδου ενός trigger πρέπει να αντιστοιχηθούν στις παραμέτρους είσοδου και εξόδου του flow.
- **Mapping:** Η φράση mapping εμφανίζεται αρκετά συχνά και αναφέρεται στην ιδέα του να παίρνεις ιδιότητες από ένα αντικείμενο και να τις συσχετίζεις με τις ιδιότητες ενός άλλου αντικειμένου. Για παράδειγμα, όταν το αντικείμενο A εκθέτει δύο ιδιότητες και το activity B δέχεται μόνο μία παράμετρο εισόδου, οι δύο ιδιότητες θα πρέπει να ενωθούν (για παράδειγμα) και να γίνουν “mapped” στην μοναδική είσοδο του activity B.
- **Activity:** Ένα activity είναι μια οντότητα εργασίας που μπορεί να αξιοποιηθεί μέσα σε ένα flow. Ένα activity μπορεί να είναι ένας οποιοσδήποτε αριθμός από πράγματα και μπορεί να συγκριθεί με μια απλή συνάρτηση στην γλώσσα Go ή σε οποιαδήποτε άλλη διαδικασιακή γλώσσα. Αυτό σημαίνει ότι ένα activity δέχεται παραμέτρους εισόδου και μπορεί να επιστρέψει ένα ή περισσότερα αντικείμενα. Τόσο οι παράμετροι εισόδου όσο και οι παράμετροι εξόδου καθορίζεται από τα μεταδεδομένα του activity.

2.3.4 WoTKit Processor

Το WoTKit Processor είναι μία υπηρεσία πολλαπλών χρηστών που αποτελεί μέρος της πλατφόρμας WoTKit και επιτρέπει στους χρήστες να επεξεργάζονται δεδομένα αισθητήρων και να αντιδρούν σε ενημερώσεις πραγματικού χρόνου από τους αισθητήρες και άλλα εξωτερικά συστήματα. Το WoTKit Processor είναι υλοποιημένο σε Java χρησιμοποιώντας το Spring Framework και αξιοποιεί το Java Concurrency Framework στην μηχανή εκτέλεσής του. Όπως και άλλα συστήματα ροής δεδομένων, η κύρια διεπαφή του είναι ένας βασισμένος σε browser οπτικός editor ροής δεδομένων. Οι χρήστες του Processor δημιουργούν προγράμματα ροής δεδομένων που ονομάζονται pipes, τα οποία αποτελούνται από modules που συνδέονται με τα λεγόμενα wires.

Μια σελίδα διαχείρισης των pipes απαριθμεί τα pipes που έχει δημιουργήσει ο χρήστης, υποδεικνύοντας την κατάσταση εκτέλεσής τους (λάθος, σε εκτέλεση, σε παύση). Οι χρήστες μπορούν να ξεκινήσουν, να σταματήσουν και να επεξεργαστούν τα δικά τους pipes από αυτή τη σελίδα. Δεν έχουν πρόσβαση στα pipes άλλων χρηστών που εκτελούνται στην πλατφόρμα. Για να αναπτύξουν ένα pipe οι χρήστες σέρνουν και αφήνουν modules στον βασικό καμβά του WoTKit Processor και τα συνδέουν μεταξύ τους με wires, όπως φαίνεται στην παρακάτω εικόνα.



Εικόνα 11: Ο βασισμένος σε browser editor ροής δεδομένων του WoTKit Processor

Αφού αποθηκεύσει ένα pipe, ο χρήστης μπορεί να ξεκινήσει την εκτέλεσή του στον editor ή στην σελίδα διαχείρισης. Στη συνέχεια το σύστημα ελέγχει το pipe για τυχόν λάθη και “μεταγλωττίζει” το pipe δημιουργώντας στιγμιότυπα των pipe modules στο server και προσθέτοντας wires σε έναν γενικό πίνακα “δρομολόγησης” που συνδέει τα άκρα των modules μεταξύ τους. Όταν όλα τα modules έχουν αποκτήσει στιγμιότυπο, το σύστημα εκκινεί το καθένα από αυτά. Στην εκκίνηση, τα modules εισόδου συνδέονται σε εξωτερικά συστήματα όπως το Twitter ή η πλατφόρμα συνάθροισης των δεδομένων αισθητήρων του WoTKit και εγγράφονται σε κατάλληλες ροές δεδομένων πραγματικού χρόνου.

Η πλατφόρμα αυτή χρησιμοποιεί έναν υπεύθυνο προγράμματος (scheduler) πολυνηματικής εκτέλεσης για να επεξεργάζεται τα δεδομένα όπως αυτά καταφθάνουν στο σύστημα από όλους τους χρήστες. Τα modules εισόδου που είναι συνδεδεμένα σε εξωτερικές υπηρεσίες προσθέτουν ενέργειες στην ουρά εκτέλεσης που αποτελούνται από ένα παραληφθέν μήνυμα και ένα στιγμιότυπο του module εισόδου. Ο scheduler περιμένει σε αυτή την ουρά και ανακτά την επόμενη ενέργεια, καλώντας την αντίστοιχη μέθοδο στο σχετικό module με τα νέα δεδομένα. Η υλοποίηση του module μπορεί να προσθέσει επιπρόσθετες ενέργειες (μηνύματα και σχετικά modules) στην ουρά εκτέλεσης ή να στείλει τα δεδομένα σε μια εξωτερική υπηρεσία μόνη της.

Για να σταματήσει την εκτέλεση ενός pipe, το σύστημα αναζητεί τα modules για το συγκεκριμένο pipe και καλεί την ρουτίνα stop() σε καθένα από αυτά. Αυτό επιτρέπει στα ακραία module να απεγγραφούν ή να αποσυνδεθούν από εξωτερικές υπηρεσίες. Στη συνέχεια αφαιρεί τα wires από τον πίνακα “δρομολόγησης” στο σύστημα, αφήνοντας να εκτελούνται τα άλλα modules και wires των υπόλοιπων pipes στο σύστημα. Αυτή η αρχιτεκτονική επιτρέπει να ελέγχονται και να διαχειρίζονται τα pipes ανεξάρτητα και αλλάζοντας την υλοποίηση της ουράς εκτέλεσης, να ελέγχεται η προτεραιότητα της εκτέλεσης των pipes μεταξύ των χρηστών.

Το WoTKit Processor περιλαμβάνει modules εισόδου και εξόδου για να στέλνει και να λαμβάνει δεδομένα από εξωτερικά συστήματα όπως αισθητήρες που φιλοξενούνται στην WoTKit Platform, ενημερώσεις του twitter και email. Ένα module παρακολούθησης χρησιμοποιείται για το τεστάρισμα και το debugging των pipes επιτρέποντας στους χρήστες να παρακολουθούν τα δεδομένα όπως ρέουν μέσα από το σύστημα. Υποστηρίζονται προειδοποιήσεις μέσω ενός email module για να στέλνονται emails όταν συμβαίνουν ενδιαφέρονται γεγονότα. Ένα module ενεργοποιητή επιτρέπει στους χρήστες να ελέγχουν τα πράγματα από ένα pipe. Περιλαμβάνονται ακόμα module συναρτήσεων για συνάθροιση δεδομένων και όρια κατωφλίου. Τέλος, το WoTKit Processor περιλαμβάνει τη δυνατότητα οι developers ή οι χρήστες του συστήματος να γράφουν τα δικά τους modules. Οι developers μπορούν να προσθέτουν νέα modules γραμμένα σε Java υλοποιώντας JavaBeans και καταχωρώντας τα στην βάση δεδομένων του Processor. Τελικοί χρήστες του συστήματος μπορούν να υλοποιούν τα δικά τους module στο UI χρησιμοποιώντας Python και αποθηκεύοντας αντίγραφα από αυτά τα scripts για να τα χρησιμοποιήσουν σε άλλα pipes.

Τα pipes του Processor περιγράφονται χρησιμοποιώντας έγγραφα JSON. O editor στον browser δημιουργεί αναπαραστάσεις σε JSON, οι οποίες αποστέλλονται στον server για αποθήκευση και εκτέλεση. [28]

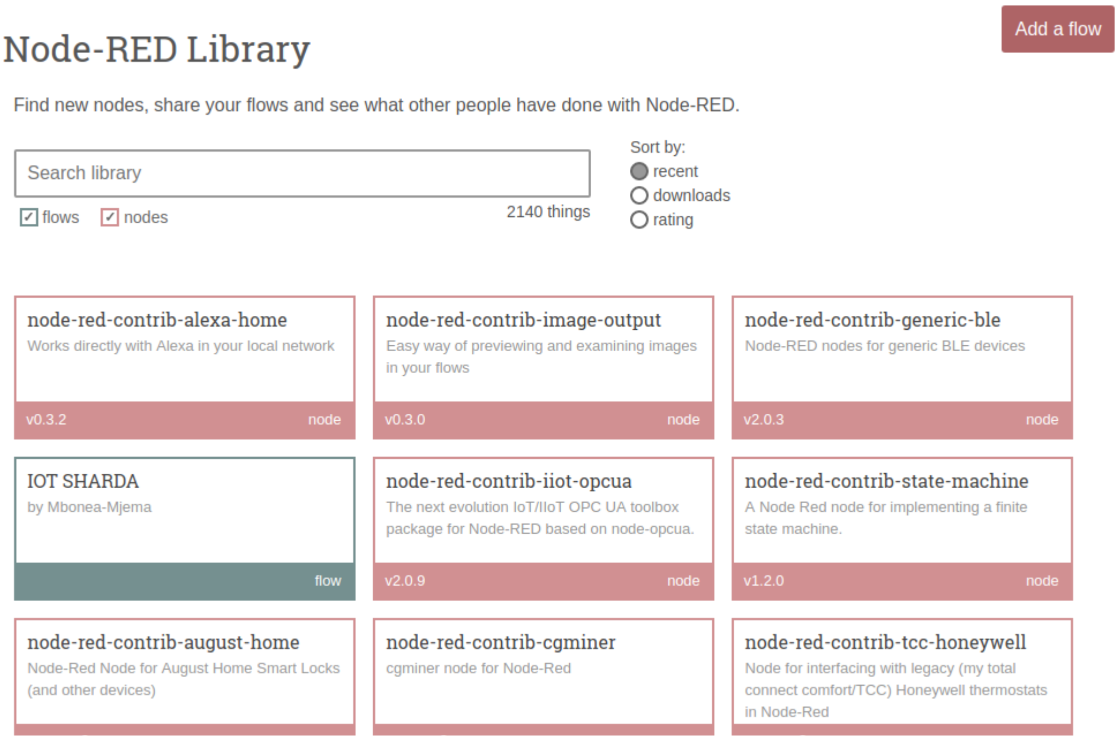
2.3.5 Node-RED

Το Node-RED είναι ένα ανοιχτού κώδικα web-based εργαλείο προγραμματισμού που επιτρέπει να συνδέονται μεταξύ τους συσκευές hardware, APIs και online υπηρεσίες. Παρέχει έναν browser-based editor ροών ή αλλιώς flows, όπως ονομάζονται στην ορολογία του Node-RED. Η πλατφόρμα αυτή είναι υλοποιημένη σε JavaScript χρησιμοποιώντας το framework Node.js, αξιοποιώντας το ενσωματωμένο μοντέλο γεγονότων του Node.js και την εγγενή του υποστήριξη για την Javascript τόσο στον client editor όσο και στον server.

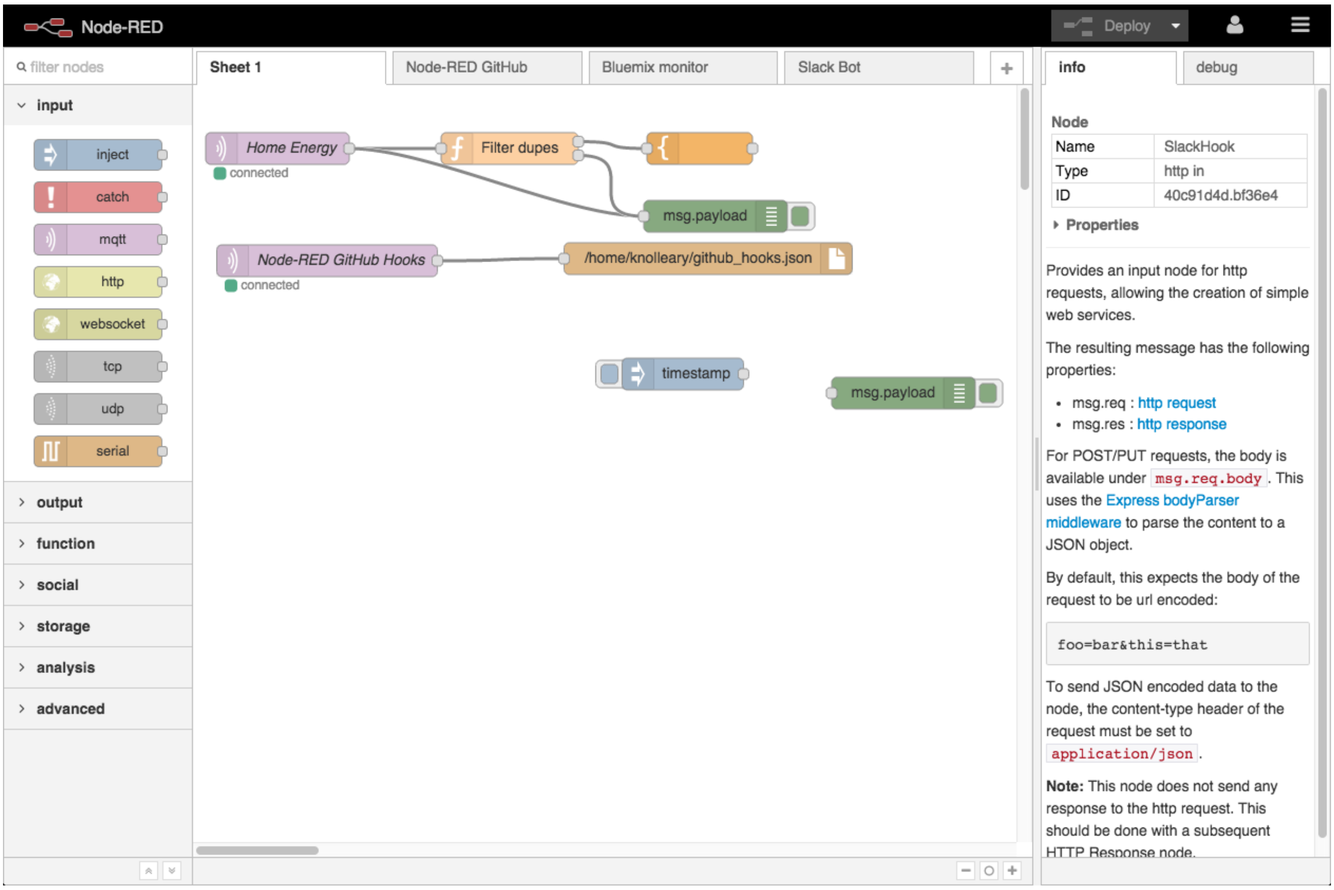
Όπως είπαμε, τα προγράμματα ροών δεδομένων στο Node-RED ονομάζονται flows και αποτελούνται από κόμβους συνδεδεμένους με καλώδια-wires. Το UI αποτελείται από τον editor των flows, όπου στα αριστερά υπάρχουν οι προκαθορισμένοι κόμβοι, τους οποίους ο χρήστης μπορεί με απλό drag&drop να τοποθετήσει στον καμβά δεξιά, όπου σχεδιάζει τα flows του. Σημαντικό να αναφέρουμε είναι ότι όλοι οι χρήστες που έχουν πρόσβαση στη διεύθυνση ενός συγκεκριμένου Node-RED editor διαχειρίζονται το ίδιο μοναδικό flow, το οποίο μπορεί να εμφανίζεται σε πολλές σελίδες ταυτόχρονα. Όταν ένα flow έχει δημιουργηθεί ή μετά από κάποια αλλαγή σε κάποιο flow που έχει ήδη φτιαχθεί, ο χρήστης τρέχει το flow, το οποίο αποθηκεύεται στον Node-RED server και ταυτόχρονα ξεκινά η εκτέλεση του σε αυτόν.

Οι κόμβοι σε ένα flow κληρονομούν από την βασική κλάση Node. Η κλάση Node είναι επίσης υποκλάση της κλάσης EventEmitter στο Node.js event API και υλοποιεί τη σχεδιαστική τεχνοτροπία observer για να διατηρήσει λίστες εγγεγραμμένων που καθορίζονται από wires και εκπέμπει γεγονότα σε χαμηλότερους κόμβους. Κατά τη διαδικασία δημιουργίας των στιγμιοτύπων, οι κόμβοι εισόδου μπορεί να εγγράφονται σε εξωτερικές υπηρεσίες να ξεκινούν να ακούνε για δεδομένα σε κάποια θύρα ή να ξεκινούν την επεξεργασία HTTP αιτημάτων. Μόλις ολοκληρωθεί η επεξεργασία των δεδομένων από έναν κόμβο, είτε από μια εξωτερική υπηρεσία είτε έχουν παραληφθεί από ένα ανώτερο κόμβο μέσω του χειριστή εισόδου του, ο κόμβος καλεί την μέθοδο send() της βασικής κλάσης Node με ένα Javascript αντικείμενο. Η μέθοδος send() αναθέτει στην μέθοδο EventEmitter.emit() να στείλει επώνυμα γεγονότα σε κατώτερα στιγμιότυπα Node τα οποία επεξεργάζονται τα δεδομένα και είτε δημιουργούν επιπλέον γεγονότα ή επικοινωνούν με εξωτερικές υπηρεσίες ή το λειτουργικό σύστημα.

Το Node-RED υποστηρίζεται από την IBM και από μια μεγάλη κοινότητα χρηστών οι οποίοι συνεισφέρουν με νέους κόμβους και flows. Νέοι κόμβοι μπορούν να υλοποιηθούν σε Javascript και να προστεθούν στο σύστημα προσθέτοντας ένα αρχείο HTML για την υλοποίηση του UI στον browser και ένα αρχείο Javascript για την επεξεργασία των δεδομένων ή την ενοποίηση στον server. Αναπαραστάσεις των flows σε μορφή κειμένου μπορούν να εισαχθούν και να εξαχθούν μεταξύ διαφορετικών στιγμιοτύπων της πλατφόρμας του Node-RED. Όταν ένας κόμβος που χρησιμοποιείται σε ένα στιγμιότυπο δεν είναι διαθέσιμος σε ένα άλλο, ένα σύμβολο υποκατάστασης εμφανίζεται στο UI για να υποδείξει στον χρήστη ότι πρέπει να εγκαταστήσει την υλοποίηση του κόμβου που λείπει για να μπορέσει να τρέξει το flow.



Εικόνα 12: Η online σελίδα flows.nodered.org όπου χρήστες μπορούν να αναζητήσουν έτοιμα flows ή κόμβους αλλά και να ανεβάσουν τα δικά τους



Εικόνα 13: Ο web-based flow editor του Node-RED.

Η μορφή κειμένου που χρησιμοποιείται για την αναπαράσταση των flows στο Node-RED είναι η JSON. Τα wires που ενώνουν τους κόμβους δεν είναι ξεχωριστά αντικείμενα, αλλά πίνακες που συσχετίζονται με κάθε κόμβο ενώνοντάς τον με κάποιον κατώτερο. Πληροφορίες παραμετροποίησης που μπορεί να διαμοιράζονται μεταξύ των κόμβων όπως το όνομα χρήστη ενός χρήστη του twitter ή ένα MQTT topic κρατιούνται σε έναν κόμβο παραμετροποίησης δίχως wires. Ακόμα, χρειάζεται προσοχή το γεγονός ότι flows σε διαφορετικά tabs του editor του Node-RED δεν διαχωρίζονται, αλλά θεωρούνται ως το ίδιο μοναδικό flow στο σύστημα. [28]

Το Node-RED αποτελεί μια αρκετά “ώριμη” πλατφόρμα, καθώς έχουν περάσει περίπου 4 χρόνια από την κυκλοφορία του και κανείς μπορεί να βρει στο Internet πολλά παραδείγματα και έγγραφα σχετικά με τη χρήση του. Υπάρχει επίσης και ειδική σελίδα με περισσότερα από 225.000 modules [29], όπου οι χρήστες μπορούν να βρουν κόμβους και flows που έχουν δημιουργήσει άλλοι χρήστες αλλά και να ανεβάσουν εκεί τα δικά τους, όπως φαίνεται στην παρακάτω εικόνα. Επίσης, το γεγονός ότι είναι αρκετά εύκολο στην εγκατάστασή του αλλά και στο να δημιουργήσει κανείς τα πρώτα flows του, το καθιστά κατάλληλο όχι μόνος για developers και integration specialists, αλλά και για κάποιον που δεν είναι αρκετά εξοικειωμένος με το γράψιμο κώδικα. [25]

Το περιβάλλον εκτέλεσης του Node-RED και η πλατφόρμα Node.js που το υποστηρίζει δεν είναι ιδιαίτερα απαιτητικά ως προς τις απαιτήσεις τους σε hardware και software και γι’ αυτό μπορεί να τρέξει τόσο σε cloud περιβάλλον και κανονικούς υπολογιστές (που τρέχουν linux, windows, κτλ.) όσο και συσκευές IoT με μικρότερες δυνατότητες, όπως το Raspberry Pi, το Beaglebone, το Arduino, κα. [29]

Τέλος, σε ό,τι αφορά τον editor του Node-RED είναι αρκετά εύκολος στη χρήση του και φιλικός προς τον χρήστη (παρέχει πχ αναλυτικές επεξηγήσεις για την λειτουργία των κόμβων), ενώ του παρέχει και πολλές έτοιμες δυνατότητες. Υπάρχουν αρκετοί διαφορετικοί κόμβοι εισόδου/εξόδου (HTTP, MQTT, websockets, TCP, UDP, serial ports, κα). Υπάρχουν πολλοί κόμβοι που εκτελούν συγκεκριμένες λειτουργίες, όπως ο κόμβος function που μπορεί ο χρήστης να γράψει δικές του συναρτήσεις σε Javascript, κόμβοι για http ή tcp requests, ο κόμβος switch που επιτρέπει τη διακλάδωση ενός flow ανάλογα με κάποιες προϋποθέσεις και κόμβοι για επεξεργασία μηνυμάτων και strings (split, csv, xml, json, κα). Επίσης, υπάρχουν κόμβοι που μπορούν να στείλουν και να δεχθούν email ή κόμβοι που μπορούν να αλληλεπιδράσουν με το Twitter. Τέλος, υπάρχουν κόμβοι για αποθήκευση σε αρχείο και σε διάφορους τύπους βάσεων δεδομένων, όπως mysql, mongodb, κα.

2.4 Επιλογή πλατφόρμας IoT

Σε ό,τι αφορά την επιλογή μιας πλατφόρμας IoT μπορούμε να αναφέρουμε μια σειρά από κριτήρια στα οποία μπορεί να ανατρέξει κανείς για να βοηθηθεί. Ας τα αναλύσουμε συνοπτικά παρακάτω [30]:

- **Κλιμακωσιμότητα:** Μια πλατφόρμα IoT θα πρέπει να μπορεί να υποστηρίξει την ανάπτυξη σε μέγεθος ενός project ή μιας επιχείρησης, αποτρέποντας τυχόν στιγμές απώλειας λειτουργίας λόγω έλλειψης πόρων. Θα πρέπει λοιπόν να παρέχει επιλογές για την κλιμάκωση του εαυτού της με τις ανάλογες μετρικές απόδοσης, μνήμης και διεκπεραιωτικότητας (throughput).
- **Αξιοπιστία:** Όταν ένα σύστημα χρησιμοποιείται στην παραγωγή, δεν υπάρχει η πολυτέλεια του καθορισμού των λόγων για τους οποίους χάνονται μηνύματα από συσκευές ή το σύνολο της πλατφόρμας δεν αποκρίνεται. Ένα σύστημα θα πρέπει να έχει δυνατότητες ανάνηψης από καταστροφές για να υποστηρίξει παραγωγικά επίπεδα αξιοπιστίας. Θα πρέπει να υλοποιεί ελέγχους υγείας, να παρακολουθεί τα συστατικά του μέρη και να παρέχει μηχανισμούς αυτόματης "ανάρρωσης" για τις περισσότερες περιπτώσεις.
- **Προσαρμοστικότητα:** Σημαντικό επίσης χαρακτηριστικό είναι ο βαθμός που μία πλατφόρμα επιτρέπει στο χρήστη να την προσαρμόζει στις απαιτήσεις του. Είναι σημαντικό να παρέχει εκτεταμένες βιβλιοθήκες, APIs και επιλογές ενοποίησης με άλλες πλατφόρμες ή υπηρεσίες, καθώς και μια δυνατότητα επέκτασης του βασικού πυρήνα της πλατφόρμας χρησιμοποιώντας προσαρμοσμένα plugins ή προσαρμοσμένα τμήματα κώδικα.
- **Διαχείριση λειτουργιών:** Θα πρέπει κανείς να υπολογίζει τι δυνατότητες παρέχει η πλατφόρμα για ενορχήστρωση και συντονισμό των δυνατοτήτων της, ειδικά όταν αυτή χρησιμοποιεί την αρχιτεκτονική των μικρο-υπηρεσιών. Ένα σημαντικό πλεονέκτημα είναι η ύπαρξη ενός ταμπλό (dashboard) που να εμφανίζει συνολικά στατιστικά για το σύστημα και πληροφορίες υγείας, την κατάσταση καθεμίας ξεχωριστής μονάδας ή υπηρεσίας, με δυνατότητα επιλογής ειδοποιήσεων σε περίπτωση ανάγκης.

- **Πρωτόκολλα:** Όσο περισσότερα πρωτόκολλα επικοινωνίας υποστηρίζει μια πλατφόρμα, τόσο το καλύτερο. Σε περίπτωση που κανείς δεν είναι σίγουρος αν θα χρησιμοποιεί για πάντα την ίδια πλατφόρμα, είναι σημαντικό αυτή να υποστηρίζει το πρωτόκολλο MQTT, καθώς είναι ένα από τα πιο παλιά και συχνά χρησιμοποιούμενα πρωτόκολλα IoT. Επιπλέον ευελιξία παρέχει επίσης η υποστήριξη API πάνω από Websockets, REST και CoAP.
- **Ανεξαρτησία από hardware:** Η βιομηχανία κατασκευής hardware εξελίσσεται πραγματικά γρήγορα και μπορεί κανείς να παρατηρήσει πόσο μικρές και ισχυρές μπορεί να είναι οι edge συσκευές. Ωστόσο με τις low-end συσκευές πρέπει κανείς να είναι προσεκτικός σε θέματα μνήμης και υπολογιστικής ισχύος και οι εφαρμογές θα πρέπει να είναι πολύ αποδοτικές στην κατανάλωση πόρων. Πριν διαλέξει κανείς μια πλατφόρμα IoT, καλό είναι να ελέγξει ποιες συσκευές υποστηρίζει, εάν διαθέτει εγγενείς υλοποιήσεις για low-end συσκευές ή αν υποστηρίζει ελαφριά πρωτόκολλα όπως το MQTT και το CoAP που μπορούν να τρέξουν πάνω από τέτοιες συσκευές.
- **Ανεξαρτησία από τεχνολογίες cloud:** Για να μην περιορίζεται η ευελιξία μιας πλατφόρμας, θα πρέπει να μπορεί να στηθεί για λειτουργία πάνω από οποιαδήποτε υποδομή είτε σε τοπικά μηχανήματα είτε στο cloud είτε σε υβριδικές περιπτώσεις. Αν μια πλατφόρμα εξαρτάται σε μεγάλο βαθμό από συγκεκριμένες υπηρεσίες cloud, τότε είναι πολύ πιθανό να λειτουργεί μόνο με το συγκεκριμένο cloud και αυτό δεν είναι καλό σε περιπτώσεις που μπορεί να χρειαστεί μελλοντικά κάποιο migration.
- **Υποστήριξη:** Είναι σημαντικό να ελέγξει κανείς πόσο εύκολο είναι να λάβει τεχνική υποστήριξη για μία πλατφόρμα, ακόμα κι αν αυτό δεν προβλέπεται να χρειαστεί σύντομα. Είναι σημαντικό να υπάρχει επαρκές documentation και άμεσα διαθέσιμο, μια ομάδα τεχνικής υποστήριξης ή ακόμα και μια ολόκληρη κοινότητα οργανωμένη σε mailing lists ή forums που να υποστηρίζουν το συγκεκριμένο προϊόν.
- **Αρχιτεκτονική και τεχνολογική δομή:** Η σημασία της αρχιτεκτονικής και των τεχνολογιών που έχουν χρησιμοποιηθεί σε μία πλατφόρμα παίζουν σημαντικό ρόλο, καθώς η μελλοντική συντήρηση της στηρίζεται κατά πολύ στον αρχικό σχεδιασμό. Η πλατφόρμα θα πρέπει να υποστηρίζει ευρέως διαδεδομένα frameworks, εργαλεία και γλώσσες. Η αρχιτεκτονική των μικρο-υπηρεσιών θα είναι καλό να ληφθεί υπόψη, μιας και παρότι παρουσιάζει κάποιες δυσκολίες, παρέχει την μεγαλύτερη ευελιξία στο να αντικατασταθεί/προσαρμοστεί κάποια υπηρεσία αφήνοντας το υπόλοιπο σύστημα άθικτο.
- **Ασφάλεια:** Η ασφάλεια στον τομέα του IoT είναι μία από τις σημαντικότερες παραμέτρους και θα πρέπει να είναι από τις πρώτες προτεραιότητες κατά την επιλογή πλατφόρμας IoT. Το λιγότερο θα πρέπει να παρέχει ένα ασφαλές επίπεδο μεταφοράς (TLS) για την επικοινωνία μεταξύ των συσκευών και των υπηρεσιών, καθώς και ταυτοποίηση για τις συσκευές και τους χρήστες. Είναι σημαντικό επίσης αν παρέχει δυνατότητα διατήρησης των δεδομένων σε κρυπτογράφηση.

Χαρακτηριστικό	Kura (+Camel)	Kaa	Flogo	WoTKit Processor	Node-RED
Αποτύπωμα εφαρμογής	>150MB	<50MB	<5MB	>150MB	<80MB
Web UI	Μόνο διαχειριστικό	Μόνο διαχειριστικό	Δημιουργία ροών και κώδικα	Δημιουργία ροών και κώδικα	Δημιουργία ροών και κώδικα
Βιβλιοθήκη έτοιμων κόμβων	Καθόλου	Καθόλου	Ναι, αλλά περιορισμένη ποικιλία	Ναι, αλλά περιορισμένη ποικιλία	Ναι, μεγάλη ποικιλία και διαρκώς επεκτάσιμη
Βιβλιοθήκη έτοιμων ροών	Καθόλου	Καθόλου	Ναι, αλλά περιορισμένη ποικιλία	Καθόλου	Ναι, μεγάλη ποικιλία
Γλώσσα Προγραμματισμού	Java	Java, C, C++	Go	Java	Javascript
Υποστηριζόμενα Λειτουργικά Συστήματα	Raspbian, Beaglebone Black, Fedora ARM	Linux, Windows, Raspbian, Beaglebone Black, Android, iOS and more	Works over Docker	Linux, Windows, Raspbian	Raspbian, Beaglebone Black, Linux, Windows, Android, Arduino, Docker
Πρωτόκολλα	MQTT	Υποστηρίζει όλα τα γνωστά πρωτόκολλα, όπως MQTT, REST, κτλ.	MQTT, REST, CoAP, κα	MQTT	MQTT, REST, Websockets, κα
Υποστήριξη	Εκτεταμένο documentation, mailing list, forum για συζητήσεις	Εκτεταμένο documentation, forum για συζητήσεις	Σύντομο documentation, forum για συζητήσεις	Σύντομο documentation	Εκτεταμένο documentation, mailing list
Ανεξαρτησία από Hardware	Ναι	Ναι	Ναι	Ναι	Ναι
Ανεξαρτησία από Cloud	Ναι	Ναι	Ναι	Ναι	Ναι

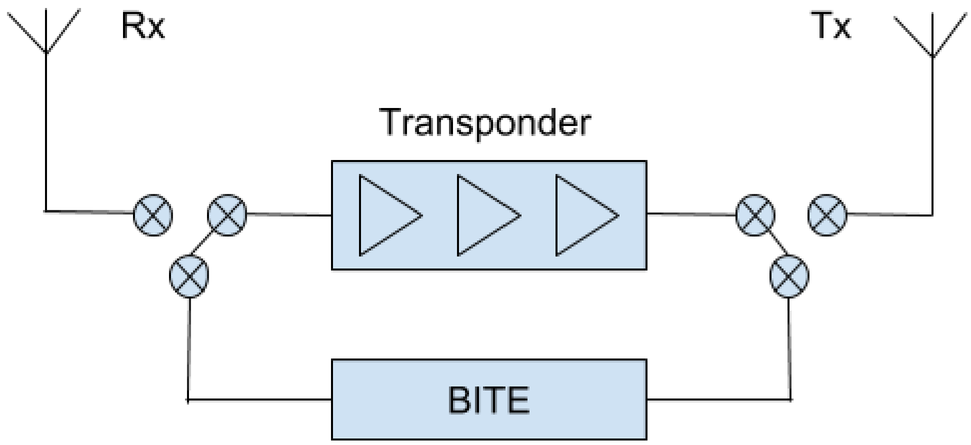
Πίνακας 4: Συγκριτικός πίνακας μεταξύ των πλατφορμών IoT που αναφέρθηκαν στην ενότητα 2.3

<u>Κεφάλαιο 3ο</u>
Περιγραφή του συστήματος της διπλωματικής

Στο κεφάλαιο αυτό αρχικά περιγράφουμε με συντομία τη λειτουργία που επιτελεί το σύστημα που υλοποιήθηκε στο πλαίσιο της παρούσας διπλωματικής εργασίας. Στη συνέχεια δίνουμε κάποιες βασικές πληροφορίες για τα διάφορα συστατικά μέρη του συστήματος αυτού.

3.1 Ανάλυση συστήματος

Στο πλαίσιο της παρούσας διπλωματικής εργασίας αναπτύχθηκε κώδικας ο οποίος αφορά τον προγραμματισμό του συστήματος που υλοποιεί τη βαθμονόμηση ενός αναμεταδότη σημάτων (transponder). Αυτός ο αναμεταδότης με τη σειρά του συμμετέχει στη βαθμονόμηση του δορυφόρου SENTINEL-3, για τον οποίο θα αναφέρουμε μερικές πληροφορίες στη συνέχεια. Ο αναμεταδότης είναι τοποθετημένος στον Ψηλορείτη τη Κρήτης μιας και τυχαίνει πολλές από τις διαφορετικές τροχιές του SENTINEL-3 να περνούν από εκείνο το σημείο και έτσι αυτό ενδείκνυται για τη βαθμονόμηση του δορυφόρου. Η βαθμονόμηση του δορυφόρου γίνεται με τη λήψη και την αναμετάδοση από τον transponder ενός σήματος που στέλνει ο δορυφόρος. Αντίστοιχα η βαθμονόμηση του transponder γίνεται με την προσομοίωση της παραπάνω διαδικασίας, απομονώνοντας με τη βοήθεια διακοπτών τις κεραίες λήψης και εκπομπής του transponder και στέλνοντας σε αυτόν μέσω ενός δεύτερου κυκλώματος μια σειρά από σήματα διαφόρων συχνοτήτων που παράγονται με τη βοήθεια ειδικού εξοπλισμού (built-in test equipment - BITE). Η βαθμονόμηση του transponder αφορά τη μέτρηση της καθυστέρησης ομάδας (group delay), έτσι ώστε να εξασφαλιστεί ότι αυτή βρίσκεται σε επιτρεπτά επίπεδα. Ο ειδικός εξοπλισμός περιλαμβάνει μια συσκευή synthesizer για την παραγωγή των συχνοτήτων η οποία επικοινωνεί με μια υπολογιστική πλακέτα Beaglebone Black για την οποία έχει επίσης σχεδιαστεί και τοποθετηθεί επάνω της ένα ειδικό Beaglebone Black Cape.



Εικόνα 14: Σχηματική αναπαράσταση του συστήματος το οποίο θα τοποθετηθεί στον Ψηλορείτη της Κρήτης

3.2 Δομικά στοιχεία υποσυστήματος

3.2.1 Ο δορυφόρος SENTINEL-3

SENTINEL-3 είναι το όνομα μιας ευρωπαϊκής δορυφορικής αποστολής παρατήρησης της Γης, η οποία αναπτύχθηκε για να υποστηρίξει εφαρμογές GMES (Global Monitoring for Environment & Security) σχετικά με τους ωκεανούς, την ξηρά, την ατμόσφαιρα, την κρυόσφαιρα (όλες οι περιοχές της Γης όπου το νερό είναι σε στερεή μορφή), επείγουσες καταστάσεις, κα. Πρόκειται για μια αποστολή που υποστηρίζεται από κοινού από την ESA και τον EUMETSAT για την παροχή λειτουργικών υπηρεσιών παρατήρησης για τους ωκεανούς και την ξηρά.

Η αποστολή καθορίζεται από την ανάγκη για συνέχιση στην παροχή δεδομένων βλάστησης από τις δορυφορικές αποστολές ERS, ENVISAT και SPOT με βελτιώσεις στην απόδοση και την κάλυψη των οργάνων. Ο κύριος στόχος της αποστολής SENTINEL-3 είναι η λήψη μετρήσεων με υψηλή ακρίβεια και αξιοπιστία σχετικά με την τοπογραφία της επιφάνειας της θάλασσας, την θερμοκρασία της επιφάνειας της θάλασσας και της ξηράς και το χρώμα της επιφάνειας των ωκεανών και της ξηράς για την υποστήριξη συστημάτων πρόβλεψης και παρακολούθησης για τους ωκεανούς, το περιβάλλον και το κλίμα. [31] Επιγραμματικά λοιπόν η αποστολή αυτή αφορά:

- Μέτρηση της τοπογραφίας της επιφάνειας της θάλασσας, του ύψους της επιφάνειας της θάλασσας και του ύψους σημαντικών κυμάτων
- Μέτρηση της θερμοκρασίας της επιφάνειας των ωκεανών και της ξηράς
- Μέτρηση του χρώματος της επιφάνειας των ωκεανών και της ξηράς
- Παρακολούθηση της τοπογραφίας του πάγου της θάλασσας και της ξηράς
- Παρακολούθηση της ποιότητας και της ρύπανσης του θαλασσινού νερού
- Παρακολούθηση του νερού της ξηράς συμπεριλαμβανομένων ποταμών και λιμνών
- Πρόγνωση του καιρού
- Παρακολούθηση και μοντελοποίηση του κλίματος
- Παρακολούθηση στην αλλαγή της χρήσης της ξηράς
- Χαρτογράφηση των περιοχών που καλύπτονται από δάση
- Ανίχνευση φωτιάς
- Μέτρηση της θερμικής ακτινοβολίας της Γης για ατμοσφαιρικές εφαρμογές

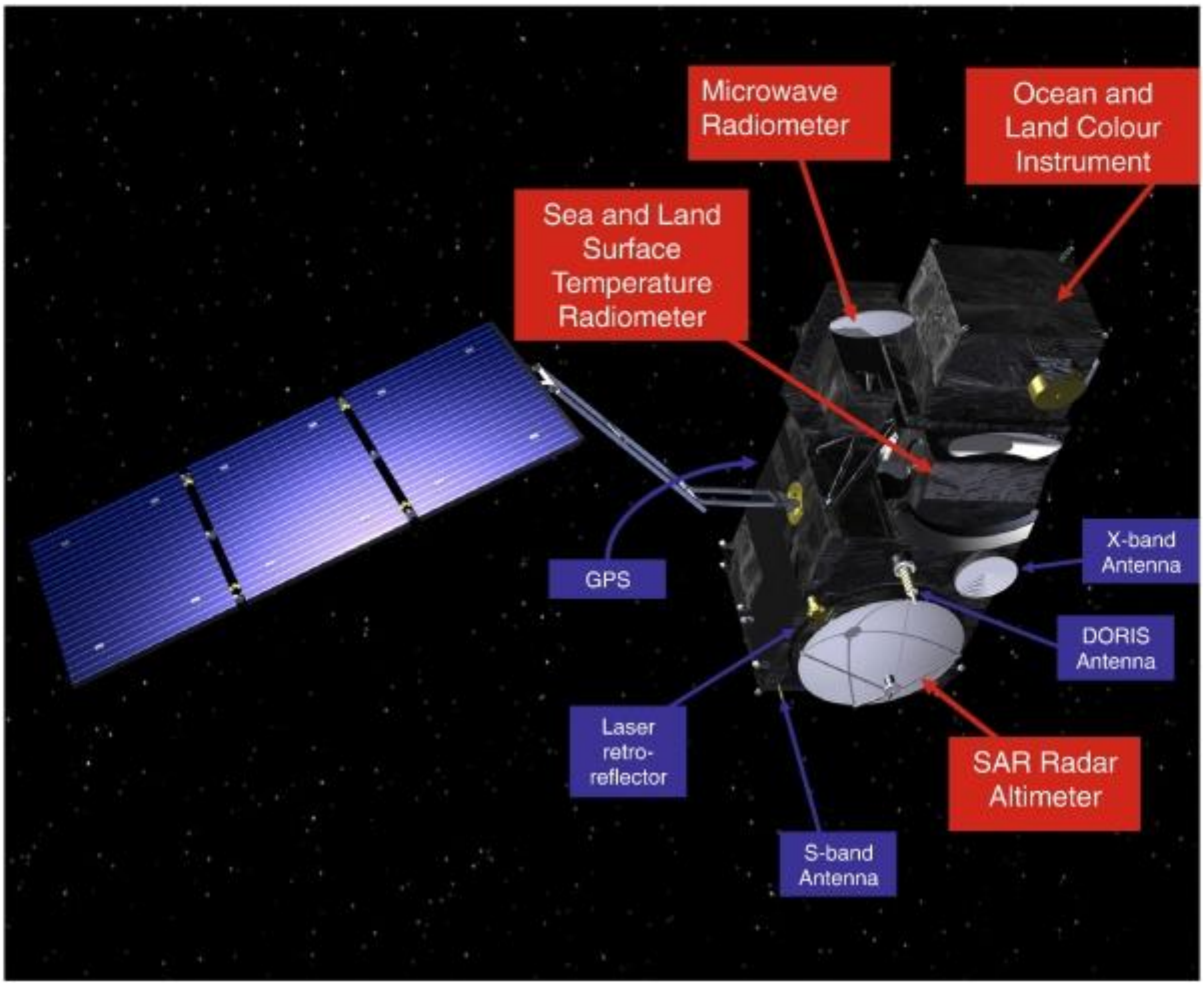
Ο SENTINEL-3 είναι ένας μεσαίου μεγέθους, χαμηλής τροχιάς γύρω από τη Γη δορυφόρος που είναι συμβατός με μικρούς εκτοξευτήρες συμπεριλαμβανομένων των VEGA και ROCKOT. Έχει σχεδιαστεί για λειτουργική περίοδο ζωής 7 χρόνων φέροντας 120 κιλά προωθητικού αερίου υδραζίνης, που επιτρέπει μέχρι και 12 χρόνια συνεχών επιχειρήσεων, συμπεριλαμβανομένης της απώλειας της τροχιάς στο τέλος της αποστολής. Η διάταξη του δορυφόρου καθορίζεται από την ανάγκη να παρέχει μια μεγάλη πρόσοψη που να "βλέπει" στο ψυχρό διάστημα για θερμικό έλεγχο, μια βαθμωτή σχεδίαση για τη στέγαση του φορτίου και απλοποιημένη διαχείριση όλων των ενσωματωμένων διεπαφών. Η ρύθμιση των μηχανικών παραμέτρων του δορυφόρου και η συμπεριφορά του εν πτήσει έχουν βελτιστοποιηθεί μέσω εντατικών μελετών ανάλυσης κατά τον καθορισμό της αποστολής, έχοντας ως αποτέλεσμα σημαντικές βελτιώσεις συγκριτικά με τον ENVISAT. [32]

Ο δορυφόρος μεταφέρει τέσσερα βασικά όργανα μετρήσεων:

- OLCI (Ocean and Land Colour Instrument)
- SLSTR (Sea and Land Surface Temperature Instrument)
- SRAL (SAR Radar Altimeter)
- MWR (Microwave Radiometer)

Αυτά συμπληρώνονται από τρία όργανα για τον ακριβή προσδιορισμό της τροχιάς (Precise Orbit Determination - POD):

- DORIS (Doppler Orbitography and Radiopositioning Integrated by Satellite)GNSS (Global Navigation Satellite System)
- LRR (Laser Retro-Reflector)



Εικόνα 15: Ο δορυφόρος Sentinel-3 και τα διάφορα όργανά του.

Θα αναφερθούμε αναλυτικότερα στο αλτίμετρο SRAL, καθώς αυτό σχετίζεται άμεσα με τη βαθμονόμηση του transponder, που όπως είπαμε αποτελεί το αντικείμενο αυτής της διπλωματικής εργασίας. Με τον όρο αλτιμετρία (altimetry) [33] περιγράφουμε μια τεχνική για την μέτρηση του ύψους. Η δορυφορική αλτιμετρία μετράει το χρόνο που χρειάζεται ο παλμός ενός ραντάρ για να ταξιδέψει από την κεραία του δορυφόρου στην επιφάνεια της γης και να επιστρέψει στον δορυφορικό δέκτη. Σε συνδυασμό με τα ακριβή δεδομένα της τοποθεσίας του δορυφόρου οι μετρήσεις αλτιμετρίας αποδίδουν το ύψος της επιφάνειας της θάλασσας.

Το SRAL είναι το βασικό τοπογραφικό όργανο που παρέχει ακριβείς τοπογραφικές μετρήσεις σχετικά με τους θαλάσσιους πάγους, τους παγετώνες, τα ποτάμια και τις λίμνες. Χρησιμοποιεί δύο ζώνες συχνοτήτων, την K_u και την C band και υποστηρίζεται από το μικροκυματικό ραδιόμετρο MWR για ατμοσφαιρική διόρθωση και τον δέκτη DORIS για προσδιορισμό της θέσης της τροχιάς. Αποτελείται από μια κεραία προσανατολισμένη προς ναδίρ, η οποία είναι τοποθετημένη στο +Zs πάνελ του δορυφόρου προς τη Γη και μια κεντρική ηλεκτρονική αλυσίδα που αποτελείται από μια Ψηφιακή Μονάδα Επεξεργασίας (DPU) και μια Μονάδα Ραδιοσυχνότητας (RFU). Η κεντρική ηλεκτρονική αλυσίδα είναι τοποθετημένη μέσα στο δορυφόρο πάνω στο -Ys πάνελ.

Η κύρια συχνότητα που χρησιμοποιείται για μετρήσεις εμβέλειας είναι η K_u-band (13,575 GHz και εύρος 350 MHz), ενώ η συχνότητα C-band (5,41 GHz και εύρος 320 MHz) χρησιμοποιείται για ιονοσφαιρική διόρθωση. Το όργανο SRAL περιλαμβάνει καταστάσεις μέτρησης, βαθμονόμησης και υποστήριξης. Οι καταστάσεις μέτρησης αποτελούνται από δύο καταστάσεις ραντάρ που συνδυάζονται με δύο καταστάσεις παρακολούθησης. Οι δύο καταστάσεις ραντάρ είναι:

- *Κατάσταση Low-Resolution (LRM)*, τυπική κατάσταση αλτιμέτρου περιορισμένου παλμού βασισμένη σε ένα μοτίβο παλμών 3 K_u / 1 C / 3 K_u.
- *Κατάσταση SAR*, υψηλής ανάλυσης κατάσταση που αποτελείται από 64 K_u-band παλμούς που περιβάλλονται από δύο C-band παλμούς.

Η αποστολή του SRAL λειτουργεί πάντα σε κατάσταση υψηλής ανάλυσης. Η κατάσταση Low-Resolution είναι εφεδρική. Οι δύο καταστάσεις παρακολούθησης που παρέχονται στον σχεδιασμό του SRAL είναι:

- *Κατάσταση closed-loop*, αυτόνομη τοποθέτηση του εύρους παραθύρου χρησιμοποιώντας τον αλγόριθμου διαμέσου.
- *Κατάσταση open-loop*, τοποθέτηση του εύρους παραθύρου με βάση *a-priori* γνώση του υψομέτρου του εδάφους που παράγεται από ένα Ψηφιακό Μοντέλο Υψομέτρου (Digital Elevation Model - DEM). [34]

Εικόνα 16: Το όργανο SRAL

Οι δύο καταστάσεις βαθμονόμησης (CAL1 και CAL2) επιτρέπουν την μέτρηση της εσωτερικής απόκρισης κρουστικού παλμού του οργάνου (ανακυκλώνοντας ένα μέρος του σήματος μετάδοσης) και τον καθορισμό της συνάρτησης μεταφοράς της αλυσίδας λήψης, που παράγεται από την απόκτηση και την εξαγωγή του μέσου όρου από χιλιάδες δείγματα θορύβου.

Υπάρχουν διαφορετικά επίπεδα καθυστέρησης των δεδομένων που σχετίζονται με τη διαθεσιμότητα των βοηθητικών δεδομένων:

- *Near Real-Time (NRT)*, που παραδίδονται σε λιγότερο από τρεις ώρες μετά την απόκτηση των δεδομένων.
- *Slow Time Critical (STC)*, που παραδίδονται μέσα σε 48 ώρες μετά την απόκτηση των δεδομένων.
- *Non-Time Critical (NTC)*, που παραδίδονται τυπικά εντός ενός μήνα μετά την απόκτηση των δεδομένων.

Υπάρχουν διαφορετικά προϊόντα δεδομένων που σχετίζονται με τα τρία επίπεδα επεξεργασίας των δεδομένων του αλτιμέτρου:

- *Level-0 (L0)* είναι τα ανεπεξέργαστα δεδομένα τηλεμετρίας.
- *Level-1 (L1)* είναι τα *Level-0* δεδομένα που έχουν διορθωθεί από επιδράσεις των οργάνων.
- *Level-2 (L2)* είναι τα *Level-1* δεδομένα που έχουν διορθωθεί από γεωφυσικές επιδράσεις.

Τα προϊόντα δεδομένων *Level-1* και *Level-2* είναι διαθέσιμα στους χρήστες. Τα προϊόντα δεδομένων *Level-0* δεν είναι διαθέσιμα στους χρήστες και θεωρούνται μόνο ως είσοδοι για την επεξεργασία *Level-1*.

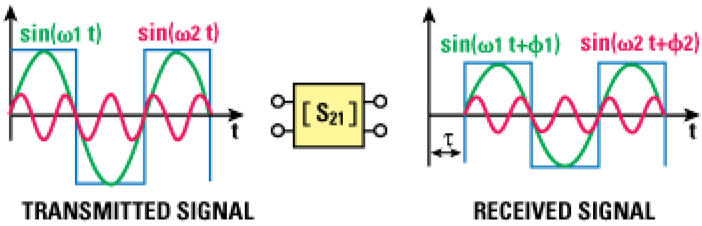
Ένα ολοκληρωμένο προϊόν *Level-2 SRAL/MWR* περιέχει τρία αρχεία δεδομένων:

- Ένα "ελαττωμένο" ("reduced" - Red) αρχείο δεδομένων που περιέχει ένα υποσύνολο από τις κύριες 1 Hz K_u band παραμέτρους.
- Ένα "στάνταρ" ("standard" - Std) αρχείο δεδομένων που περιέχει τις στάνταρ 1 Hz και 20 Hz K_u και C-band παραμέτρους.
- Ένα "ενισχυμένο" ("enhanced" - Enh) αρχείο δεδομένων που περιέχει τις στάνταρ 1 Hz και 20 Hz K_u και C-band παραμέτρους, τις κυματομορφές και τις συσχετισμένες παραμέτρους που είναι απαραίτητες για να επαναναλυθούν τα δεδομένα.

Τα προϊόντα *SRAL/MWR Level-2* παράγονται στο φορμάτ SAFE (Standard Archive Format for Europe). Όλες οι πληροφορίες σχετικά με το προϊόν συλλέγονται σε ένα μοναδικό πακέτο. Μέσα στο πακέτο τα συγκεκριμένα αντικείμενα που περιέχουν τα δεδομένα των μετρήσεων είναι κωδικοποιημένα στο φορμάτ netCDF.[35]

3.2.2 Υποσύστημα ελέγχου σημάτων

Για τη σωστή μετάδοση της πληροφορίας το σχήμα του σήματος στην έξοδο του συστήματος μετάδοσης θα πρέπει να είναι πανομοιότυπο με το σχήμα στην είσοδο του. Τα πλάτη των σημάτων εισόδου και εξόδου μπορεί να διαφέρουν, καθώς το σήμα μπορεί αν υφίσταται ενίσχυση ή εξασθένηση για να ταιριάζει το πλάτος του στις συνθήκες που επικρατούν στην πλευρά του δέκτη. Επίσης, δεν υπάρχει επίδραση στην ποιότητα του σήματος αν το σήμα φτάνει στο δέκτη με κάποια καθυστέρηση. Ωστόσο για τη σωστή μετάδοση όλες οι συνιστώσες συχνότητας μέσα σε ένα κανάλι μετάδοσης ή στο εύρος μετάδοσης πρέπει να υφίστανται πανομοιότυπη εξασθένιση ή ενίσχυση και να έχουν πανομοιότυπη καθυστέρηση σχετικά με το σήμα εισόδου. Στην εικόνα 17 μπορούμε να δούμε ένα παράδειγμα ενός μεταδιδόμενου και ενός λαμβανόμενου σήματος.



Εικόνα 17: Μεταδιδόμενο και λαμβανόμενο σήμα

Για να έχουν δύο σήματα με συχνότητες ω_1 και ω_2 την ίδια καθυστέρηση μέσα στο κανάλι μετάδοσης, θα πρέπει να ισχύουν τα ακόλουθα:

$$\varphi = -\omega\tau \ (1)$$

$$\varphi_1 = -\omega_1\tau$$

$$\varphi_2 = -\omega_2\tau$$

$$\varphi_2 - \varphi_1 = -\tau(\omega_2 - \omega_1)$$

$$-\frac{\Delta\varphi}{\Delta\omega} = \tau, \text{ όπου } \Delta\varphi = \varphi_2 - \varphi_1 \text{ και } \Delta\omega = \omega_2 - \omega_1$$

Για να υπάρχει σταθερή καθυστέρηση για όλες τις συχνότητες, η φάση του συντελεστή μετάδοσης S_{21} πρέπει να είναι μία γραμμική συνάρτηση της συχνότητας. Ξεκινώντας από την εξίσωση (1), αλλάζοντας τη μονάδα του Δφ από ακτίνια σε μοίρες, αντικαθιστώντας το Δω με το 2πΔf και μειώνοντας το μέγεθος του βήματος σε απειροελάχιστες τιμές, λαμβάνουμε την ακόλουθη σχέση:

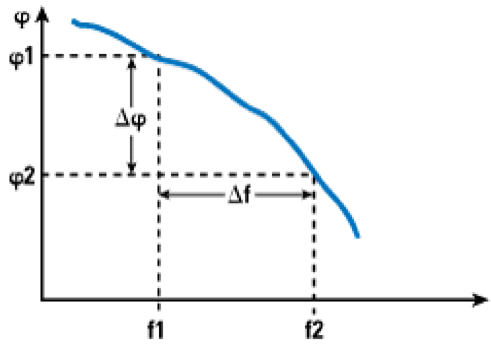
$$\tau = -\frac{1}{360^o} \bullet \frac{d\varphi}{df}$$

Η καθυστέρηση τ αναφέρεται ως καθυστέρηση ομάδας (group delay) και ορίζεται ως η αρνητική παράγωγος της φάσης ως προς τη συχνότητα. Για να μη δημιουργεί παραμόρφωση ένα κανάλι μετάδοσης, ο συντελεστής μετάδοσής του S_{21} πρέπει να έχει σταθερό μέγεθος και γραμμική φάση ως προς τη συχνότητα. Αντίθετα, οι απόλυτες τιμές της φάσης και της κλίσης της φάσης ως προς τη συχνότητα (δηλαδή της καθυστέρησης ομάδας) δεν επιδρούν στην ποιότητα της μετάδοσης.

Συχνά αντί του προσδιορισμού της παραγώγου της φάσης ως προς τη συχνότητα, η οποία στηρίζεται σε απειροελάχιστες τιμές, υπολογίζεται ο συντελεστής διαφοράς:

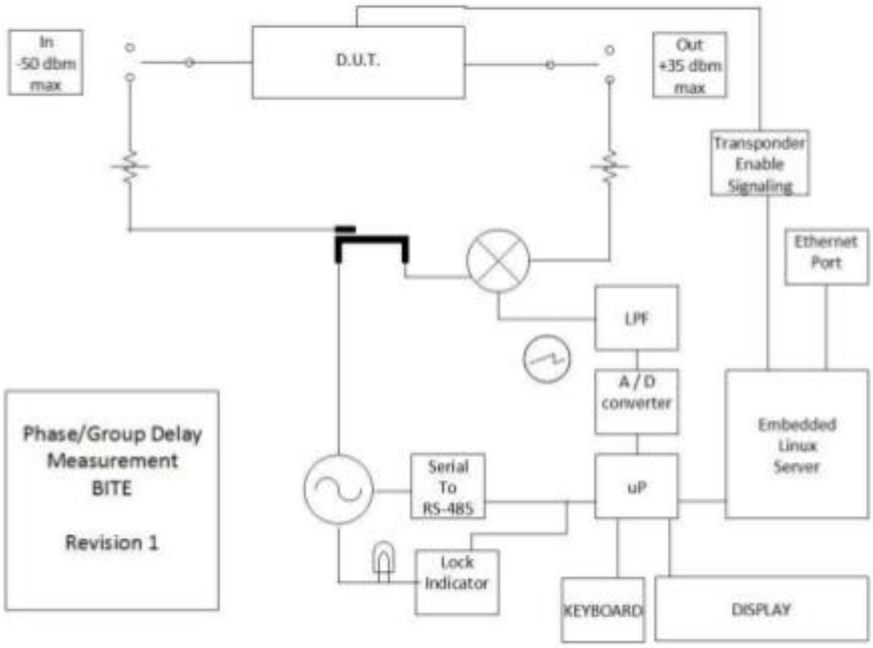
$$\tau = -\frac{1}{360^o} \bullet \frac{\Delta\varphi}{\Delta f} \ (2)$$

Ο παραπάνω τύπος δίνει μια καλή προσέγγιση της απαιτούμενης καθυστέρησης ομάδας. Στην εικόνα 18 φαίνεται πώς τα $\Delta\varphi = \varphi_2 - \varphi_1$ και $\Delta f = f_2 - f_1$ χρησιμοποιούνται για τον υπολογισμό της καθυστέρησης ομάδας. [36]



Εικόνα 18: Οι όροι Δφ και Δf για τον υπολογισμό της καθυστέρησης ομάδας.

Στο παρακάτω σχήμα φαίνεται η τοπολογία του ειδικού εξοπλισμού (BITE) που χρησιμοποιείται στο σύστημά μας για την εξαγωγή της καθυστέρησης ομάδας:



Εικόνα 19: Η τοπολογία του ειδικού εξοπλισμού (Built-In Test Equipment - BITE) του συστήματος

Ο σκοπός της παρακολούθησης της φάσης είναι για να εξαχθεί η εσωτερική καθυστέρηση. Πρόκειται λοιπόν να σαρώσουμε με το PLL (Phase-Locked-Loop) σύστημά μας ένα εύρος συχνοτήτων που καλύπτουν τη ζώνη του αλτιμέτρου (350 MHz γύρω από τα 13,775 GHz). Για κάθε συχνότητα θα πάρουμε τη διαφορά φάσης μεταξύ του PLL σήματος που δίνεται στην είσοδου του transponder και του σήματος στην έξοδο του transponder. Η διαφορά φάσης είναι ανάλογη με την εσωτερική καθυστέρηση. Στην είσοδο έχουμε $\sin(2\pi ft+\varphi_0)$ και στην έξοδο έχουμε $\sin(2\pi f(t+\Delta t)+\varphi_0)$, όπου f είναι η συχνότητα που παράγεται από το PLL και Δt είναι η καθυστέρηση. Έτσι, μετά τη σύγκριση της φάσης (η οποία εξάγεται πολλαπλασιάζοντας τα δύο ημιτονοειδή κύματα) παίρνουμε τη διαφορά της φάσης $\Delta\varphi = 2\pi f\Delta t$. Με βάση τη θεωρία μια μέτρηση με μόνο μία συχνότητα δεν είναι αρκετή για να επιλύσει την ασάφεια του 2π και να εξαχθεί μια τιμή για το Δt. Γι' αυτό το λόγο πραγματοποιούμε μια σειρά από μετρήσεις με μεταβαλλόμενο f με ένα σταθερό βήμα Δf. [37]

Όπως βλέπουμε στην εικόνα 19, στην τοπολογία του ειδικού εξοπλισμού που χρησιμοποιούμε γίνεται μίξη δύο σημάτων. Η έξοδος του μίκτη δίνει έναν καλό αριθμό από φασματικά προϊόντα. Αυτά στη συνέχεια φιλτράρονται από ένα βαθυπερατό φίλτρο έτσι ώστε να κρατήσουμε μόνο το DC σήμα. Έπειτα η DC αυτή τάση περνάει από έναν μετατροπέα A/D (Analog to Digital) των n bits και κβαντίζεται σε 2^n επίπεδα τάσης. Χρειάζεται λοιπόν να υπολογίσουμε τον ελάχιστο αριθμό από bits που χρειάζεται να έχει ο A/D μετατροπέας για να μπορούμε να μετρήσουμε την καθυστέρηση ομάδας ανάμεσα στα δύο αυτά πανομοιότυπα σήματα που μιξάρονται και φιλτράρονται. Θα αποδείξουμε ακόμα ότι η DC τάση που προκύπτει από αυτή τη μίξη είναι συνάρτηση της διαφοράς φάσης των δύο σημάτω και επιπρόσθετα ότι η διαφορά φάσης είναι γραμμική συνάρτηση της συχνότητας.

Η ισχύς του τοπικού ταλαντωτή είναι 10dBm (10mW) και το σήμα του οδηγείται σε ένα -3dB διαχωριστή (splitter) απ' όπου παίρνουμε δύο σήματα 7dBm και 5,9dBm αντίστοιχα (υπάρχει απώλεια 1,1dB λόγω της εξασθένισης του ενός σήματος που περνά από την DUT (Device Under Test), δηλαδή του transponder. Το εύρος ζώνης του τοπικού ταλαντωτή κυμαίνεται από 13 GHz έως 14 GHz. Έτσι, προκύπτει ότι V_1=0,5023 V_{RMS} και V_2=0,3027 V_{RMS}.

Όταν δύο σήματα της ίδιας συχνότητας μιξάρονται, το αποτέλεσμα θα πρέπει να είναι το ανάπτυγμα της σειράς Taylor των αρμονικών προϊόντων τους. Έστω, ότι τα δύο σήματα της ίδιας συχνότητας είναι της μορφής:

$$v_1(t) = V_1 \cdot \cos(\omega t)$$
$$v_2(t) = V_2 \cdot \cos(\omega t + \varphi)$$

Τότε, ο μίκτης θα δώσει ένα σήμα της μορφής:

$$v_{out}(t) = a_0 + a_1 \cdot \left[v_1(t) \cdot v_2(t)\right] + a_2 \cdot \left[v_1(t) \cdot v_2(t)\right]^2 + a_3 \cdot \left[v_1(t) \cdot v_2(t)\right]^3 + \ldots$$

Ακόμα, λαμβάνουμε υπόψη ότι οι συντελεστές α_2, α_3, …, α_n είναι αμελητέοι και κάνουμε χρήση της τριγωνομετρικής ταυτότητας:

$$\cos(a) \cdot \cos(b) = \frac{\cos(a+b) + \cos(a-b)}{2}$$

Έτσι, προκύπτει ότι η DC τάση εξόδου παίρνει την τιμή:

$$v_{out,DC} = a_0 + \frac{a_1 V_1 V_2}{2} \cos(\varphi)$$

Επιπλέον, ο μίκτης κατασκευάζεται με τέτοιο τρόπο ώστε ο συντελεστής α_0 να τείνει στο μηδέν και V_{p-p}(f) = 1V. Το βαθυπερατό φίλτρο θα αφήσει μόνο την DC συνιστώσα της τάσης να περάσει στον μετατροπέα A/D, έτσι το σήμα στην είσοδο του μετατροπέα γίνεται:

$$v_{in,A/D} = \frac{a_1 V_1 V_2}{2} \cos(\varphi)$$

Και επειδή είναι $α_1$=6,577, τελικά έχουμε:

$$v_{in,A/D} = 0.5\cos(\varphi)$$

$$v_{in,A/D} = 0.5\cos(k\Delta L)$$

$$v_{in,A/D} = 0.5\cos(\frac{2\pi f_{LO}}{c}\Delta L)$$

Πολλαπλοί παράγοντες όπως η διαφοροποίηση της θερμοκρασίας επιδρούν στο μήκος της γραμμής μετάδοσης, αφού ο χρόνος που χρειάζεται ένα κύμα για να διαπεράσει το εκάστοτε μέσο αλλάζει με τη θερμοκρασία. Έτσι, το μήκος της γραμμής θα είναι ουσιαστικά:

$$\Delta L_{dist} = \Delta L + x$$

όπου x είναι η παραμόρφωση του μήκους που μπορεί να πάρεις θετικές ή αρνητικές τιμές και ΔL_{dist} είναι το πραγματικό μήκος της γραμμής μετάδοσης. Μπορούμε ακόμα να αξιοποιήσουμε την εξίσωση της διάδοσης κύματος και να ξαναγράψουμε την παραπάνω εξίσωση ως:

$$\Delta L_{dist} = c \cdot (\Delta t + GroupDelay)$$

$$GroupDelay = \frac{x}{c}$$

Η ελάχιστη καθυστέρηση ομάδας που μπορούμε να επιτύχουμε για τη δική μας συσκευή είναι της τάξης των 20psec. Αυτό μας οδηγεί να υπολογίσουμε την ελάχιστη παραμόρφωση μήκους:

$$x_{\min} = c \cdot GroupDelay_{\min}$$

Έτσι, η διαφορά φάσης φ θα μεταβάλλεται με την παραμόρφωση μήκους και η τάση του σήματος στην είσοδο του μετατροπέα A/D θα δίνεται από την ακόλουθη εξίσωση:

$$v_{in,A/D} = 0.5\cos(\frac{2\pi f_{LO}}{c}\Delta L_{dist})$$

$$v_{in,A/D} = 0.5\cos\left[2\pi f_{LO}(\Delta t + GroupDelay)\right]$$

Το ελάχιστο επίπεδο τάσης που θα πρέπει να αναγνωρίζεται από τον μετατροπέα A/D πρέπει να είναι το ελάχιστο της απόλυτης διαφοράς ανάμεσα στην τάση εισόδου του A/D στους 25°C και της τάσης είδοσου του A/D σε κάποια άλλη θερμοκρασία πάνω από τους 10°C.

$$v_{in,A/D@25} = 0.5\cos(2\pi f_{LO}\Delta t)$$

$$v_{in,A/D@T} = 0.5\cos\left[2\pi f_{LO}(\Delta t + GroupDelay_{\min})\right]$$

$$V_{level} = \left|v_{in,A/D@25} - v_{in,A/D@T}\right|_{\min} \cdot 0.1$$

Λαμβάνοντας υπόψη ότι η τάση peak-to-peak είναι 1V μπορούμε να υπολογίσουμε τον αριθμό των bits για τον μετατροπέα A/D χρησιμοποιώντας τις παραπάνω σχέσεις. Η τάση peak-to-peak πρέπει να κβαντίζεται σε έναν αριθμό επιπέδων που δίνεται από τη σχέση:

$$NumberOfLevels = ceil\left(\frac{V_{p-p}}{V_{level}}\right)$$

$$NumberOfBits = ceil\left(\log_2\left(NumberOfLevels\right)\right)$$

Κάνοντας τους σχετικούς υπολογισμούς προκύπτει ότι ο μετατροπέας A/D χρειάζεται να έχει 16 bits. [38]

3.2.3 Synthesizer

Το synthesizer SLSM5 χρησιμοποιεί την πιο πρόσφατη τεχνολογία N τμημάτων δημιουργώντας ένα πολυχρηστικό synthesizer συχνοτήτων υψηλής απόδοσης. Η αρχιτεκτονική του SLSM5 επιτρέπει στα synthesizer να έχουν ένα πολύ μεγάλο εύρος ζώνης με βηματική ανάλυση 1KHz μέχρι και τα 32 GHz. Το SLSM5 είναι ιδανικό για πολλές εφαρμογές στους τομείς των επικοινωνιών και των ραντάρ και προσφέρει τέλεια απόδοση σε ό,τι αφορά θόρυβο στην φάση του σήματος.

Το synthesizer μπορεί να λειτουργήσει τόσο με εξωτερικό σήμα αναφοράς όσο και με εσωτερικό. Το εξωτερικό σήμα αναφοράς πρέπει να είναι της τάξης των 10 MHz και να είναι καλής ποιότητας σήμα στο επίπεδο των 0dBm±3dB. Στην παραμετροποίηση του συγκεκριμένου synthesizer χρησιμοποιείται μόνο εσωτερικό σήμα αναφοράς 10MHz από έναν ταλαντωτή TCXO (±0.5ppm -10°C to +70°C).

Η DC τροφοδοσία και ο έλεγχος της συχνότητας γίνεται μέσω του P1, ο οποίος είναι μια υποδοχή 9 αρσενικών pin D-SUB. Στον πίνακα της επόμενης σελίδας φαίνεται η λειτουργία του κάθε pin. Αν και το synthesizer διαθέτει έναν εσωτερικό LDR και φιλτράρισμα, η παρεχόμενη DC τάση θα πρέπει να προέρχεται από μία ποιοτική μονάδα τροφοδοσίας που να είναι ικανή να παρέχει ρεύμα 650 mA.

Η παρακολούθηση της κατάστασης του synthesizer για κλείδωμα της φάσης μπορεί να γίνει με τους δύο παρακάτω τρόπους:

- Μέσω του LD Pin 7 της υποδοχής P1. Αυτός ο ανιχνευτής κλειδώματος υποδεικνύει μια κατάσταση αποτυχίας και είναι χαμηλά όταν ανιχνεύεται κάποια αστοχία. Είναι παραμετροποιημένος ως ανοιχτός συλλέκτης και είναι έγκυρος για όλες τις καταστάσεις σφάλματος συμπεριλαμβανομένης και της απουσίας σήματος αναφοράς.
- Με polling μέσω της σειριακής διεπαφής. Ωστόσο αυτός ο τρόπος δεν θα δίνει απόκριση σε περίπτωση που αφαιρεθεί το σήμα αναφοράς.

Επίσης, οφείλουμε να σημειώσουμε ότι η παρακολούθηση του Pin 7 δεν προσθέτει θόρυβο στην έξοδο, ενώ το polling μέσω της σειριακής διεπαφής μπορεί να έχει ως αποτέλεσμα υποβάθμιση του φάσματος.

Αριθμός Pin	**Περιγραφή**
1	TX+ (out of synthesizer)
2	TX- (out of synthesizer)
3	RX- (into synthesizer)
4	RX+ (into synthesizer)
5	Επιλογή ταχύτητας (0=115.2K, 1=9600)
6	+5.5 Vdc ±0.5 V
7	(LD) Ανιχνευτής Κλειδώματος
8	MR (Master Reset) 0-3V (ενεργό χαμηλά)
9	GND

Πίνακας 5: Περιγραφή των λειτουργιών των 9 pin της υποδοχής P1

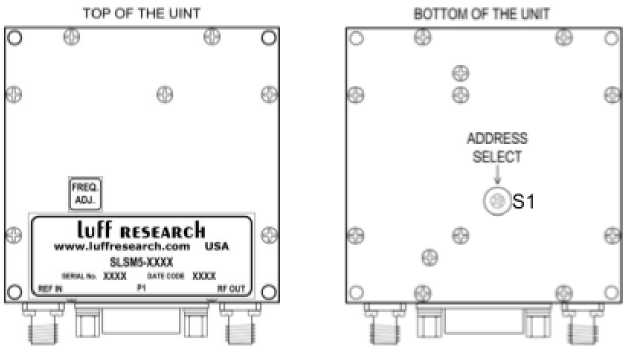
Το SLSM5 έχει εσωτερική μη πτητική μνήμη (EEPROM) η οποία διατηρεί τις ρυθμίσεις της τελευταίας εντολής κατάστασης. Η συχνότητα και η κατάσταση σιγής αποθηκεύονται σε αυτή τη μνήμη. Κατά την ενεργοποίησή του το synthesizer επαναφέρεται στην τελευταία αποθηκευμένη κατάσταση. Η μη πτητική μνήμη (EEPROM) περιορίζεται στους 1000000 κύκλους γραψίματος.

Για εφαρμογές που απαιτούν απεριόριστους κύκλους συντονισμού όπως η συνεχής σάρωση ζώνης ή η συνεχής τυχαία απαίτηση συχνότητας, παρέχεται η εντολή hop-frequency. Η εντολή αυτή δεν αποθηκεύει τη συχνότητα που ζητείται στη μη πτητική μνήμη (EEPROM).

Το synthesizer συχνοτήτων SLSM5 παρακολουθείται και ελέγχεται μέσω μιας ασύγχρονης σειριακής διεπαφής (UART). Η ηλεκτρική διεπαφή αντιλαμβάνεται το πρωτόκολλο RS485 και μπορεί να καλωδιωθεί είτε σε κατάσταση δύο είτε σε κατάσταση τεσσάρων καλωδίων. Η σειριακή επικοινωνία λειτουργεί είτε στα 9600 Baud είτε στα 115200 Baud, ρύθμιση που επιλέγεται όπως είδαμε παραπάνω μέσω της διεπαφής της υποδοχής. Επιπλέον ρυθμίσεις για τη σειριακή διεπαφή είναι τα 8 bits δεδομένων, 1 stop bit, απουσία parity και απουσία ελέγχου ροής. Το πρωτόκολλο είναι διευθυνσιοδοτήσιμο επιτρέποντας την επικοινωνία με πολλαπλές συσκευές. Η διεύθυνση επιλέγεται μέσω ενός περιστροφικού διακόπτη πάνω στη μονάδα.

Η σειριακή επικοινωνία λαμβάνει χώρα είτε μόνο με την εξωτερική αναφορά 10MHz είτε με τον εσωτερικό ταλαντωτή TCXO. Η σειριακή διεπαφή του synthesize λειτουργεί σε κατάσταση εντολής/απόκρισης. Ο απομακρυσμένος δέκτης παρέχει την επιθυμητή ρύθμιση συχνότητας σε ASCII τιμές και το synthesizer απαντά με μια εντολή απόκρισης αποδοχής ή απόρριψης. Ένας ASCII χαρακτήρας ‘A’ επιστρέφεται όταν η εντολή γίνεται αποδεκτή και ένας ASCII χαρακτήρας ‘R’ επιστρέφεται σε περίπτωση απόρριψης της εντολής. Η απόκριση ACK/NACK πρέπει να ακολουθείται από έναν χαρακτήρα αλλαγής γραμμής. Η έξοδος του synthesizer μπορεί να απενεργοποιηθεί (σίγαση) ή να ενεργοποιηθεί μέσω της σειριακής διεπαφής. Επίσης, μέσω της σειριακής διεπαφής μπορεί να ανακτηθεί η τρέχουσα κατάσταση του synthesizer.

Δύο χαρακτήρες ASCII που αναπαριστούν τη ρύθμιση της διεύθυνσης επιτρέπουν πολλαπλά synthesizer να είναι ενωμένα μαζί σε κατάσταση RS-422/RS-485. Το synthesizer θα πρέπει να δέχεται και να αποκρίνεται σε εντολές που περιέχουν τις τιμές διεύθυνσης που ταιριάζουν με την ρύθμιση του περιστροφικού διακόπτη S1, ο οποίος φαίνεται στην εικόνα 20 παρακάτω. Η διεύθυνση 0xFF χρησιμοποιείται ως γενική διεύθυνση και η μονάδα θα απαντά πάντα σε αυτή τη διεύθυνση. Η διεύθυνση στην απόκριση θα αντιστοιχεί στην πραγματική ρύθμιση. Αυτό το χαρακτηριστικό επιτρέπει στο χρήστη να καθορίσει τη ρύθμιση του περιστροφικού διακόπτη μέσω της σειριακής διεπαφής. Αυτό το χαρακτηριστικό δεν μπορεί να χρησιμοποιηθεί σε κατάσταση σύνδεσης πολλαπλών συσκευών. Η μονάδα εκ κατασκευής είναι ρυθμισμένη στη διεύθυνση 0x01. [39]



Εικόνα 20: Σχηματική αναπαράσταση του synthesizer SLSM5

Υπάρχουν τρεις εντολές διαθέσιμες για τον έλεγχο του synthesizer. Κατά την εκκίνηση το synthesizer επιστρέφει στην τελευταία κατάσταση που είναι αποθηκευμένη στην μη πτητική μνήμη του. Πριν παραθέσουμε τη σύνταξη των εντολών ελέγχου της συσκευής, προχωράμε σε μερικούς ορισμούς:

- Όλοι οι χαρακτήρες είναι ASCII χαρακτήρες.
- > δηλώνει την αρχή μηνύματος από τη συσκευή ελέγχου προς το synthesizer
- < δηλώνει την αρχή μηνύματος από το synthesizer προς τη συσκευή ελέγχου
- ? δηλώνει ερώτηση για την κατάσταση του synthesizer
- F δηλώνει την εντολή συχνότητας. Η τιμή της συχνότητας αποθηκεύεται στη μη πτητική μνήμη.
- H δηλώνει την εντολή hop-frequency. Η τιμή της συχνότητας δεν αποθηκεύεται στη μη πτητική μνήμη.
- n_1n_2 αναπαριστά τη διεύθυνση από το "00" έως το "0F" όπως τίθεται από το διακόπτη S1.
- fffffff αναπαριστά την τρέχουσα συχνότητα σε βήματα του 1 KHz.
- L είναι η κατάσταση κλειδώματος/μη κλειδώματος του synthesizer, "L"=Κατάσταση κλειδώματος, "U"=Κατάσταση μη κλειδώματος, "M"=Κατάσταση σίγασης
- ↵ είναι ο χαρακτήρας αλλαγής γραμμής

- M δηλώνει την εντολή σίγασης
- m παίρνει τιμή “0” ή “1”
- “0” δηλώνει ότι η έξοδος του synthesizer είναι σε κατάσταση OFF
- “1” δηλώνει ότι η έξοδος του synthesizer είναι σε κατάσταση ON
- A δηλώνει αποδοχή της εντολής
- R δηλώνει απόρριψη της εντολής

Παραθέτουμε τώρα τη σύνταξη των εντολών που δέχεται το synthesizer:

- *Εντολή συχνότητας*
  - Εντολή: >n_1n_2Fffffff↵
  - Απόκριση: <n_1n_2A
  - Παράδειγμα για ρύθμιση στα 3,3 GHz στη διεύθυνση 01:
    - Εντολή: >01F3300000↵
    - Απόκριση: <01A
  - Παράδειγμα με λάθος στη συχνότητα:
    - Εντολή: >01F3300↵
    - Απόκριση: <01R
- *Εντολή hop-frequency*
  - Εντολή: >n_1n_2Hfffffff↵
  - Απόκριση: <n_1n_2A
  - Παράδειγμα για ρύθμιση στα 3,3 GHz στη διεύθυνση 01:
    - Εντολή: >01H3300000↵
    - Απόκριση: <01A
  - Παράδειγμα με λάθος στη συχνότητα:
    - Εντολή: >01H3300↵
    - Απόκριση: <01R
- *Εντολή σίγασης*
  - Εντολή: >n_1n_2Mm↵
  - Απόκριση: <n_1n_2A
  - Παράδειγμα:
    - Εντολή:>01M1↵
    - Απόκριση: <01A
- *Εντολή αίτησης κατάστασης*
  - Εντολή: >n_1n_2? ↵
  - Απόκριση: <n_1n_2FfffffffL
  - Παράδειγμα:
    - Εντολή:>01? ↵
    - Απόκριση: <01F3300000L

3.2.4 Πλακέτα ελέγχου

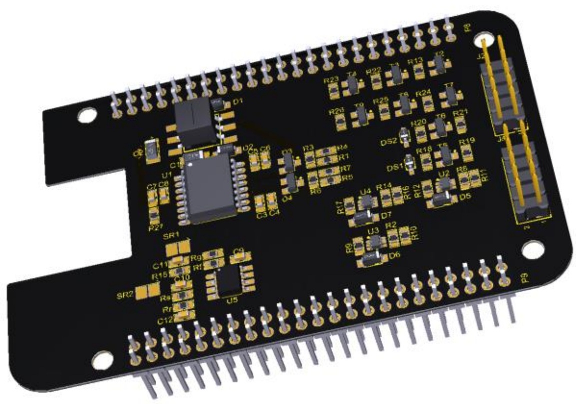
Για τον προγραμματισμό του συστήματος, την επικοινωνία με το synthesizer αλλά και τη λειτουργία του web server για την frond-end πλατφόρμα διαχείρισης και απεικόνισης των διαγραμμάτων των μετρήσεων επιλέχθηκε η υπολογιστική πλακέτα Beaglebone Black. Σε αυτή τρέχει ο κώδικας που λαμβάνει τις μετρήσεις από τον μετατροπέα A/D για τις διάφορες τιμές των συχνοτήτων, καθώς και ο κώδικας που στη συνέχεια εκτελεί όλους τους απαραίτητους υπολογισμούς.

Εικόνα 21: Η υπολογιστική πλακέτα Beaglebone Black

Ο λόγος που επιλέχθηκε η συγκεκριμένη υπολογιστική πλακέτα έναντι άλλων παρόμοιων, όπως το RaspberryPi, ήταν η υπεροχή της σε υπολογιστικούς πόρους την περίοδο σχεδιασμού του project σε συνδυασμό με το χαμηλό της κόστος. Ωστόσο υπήρχε το πρόβλημα ότι ο μετατροπέας A/D ήταν 12 bit και όχι 16 bit, όπως είδαμε παραπάνω ότι χρειάζεται το σύστημά μας. Το πρόβλημα αυτό αντιμετωπίστηκε κάνοντας ενίσχυση του σήματος επί 2 και επιβάλλοντας λήψη του μέσου όρου 10 τιμών στις μετρήσεις που λαμβάνουμε για κάθε συχνότητα.

Επιπλέον, για επέκταση των δυνατοτήτων του Beaglebone Black σχεδιάστηκε και τοποθετήθηκε στις ειδικές υποδοχές ειδικό Beaglebone Cape. Το Cape αυτό επιτελεί διάφορες λειτουργίες εκ των οποίων οι δύο σημαντικότερες είναι η εξής:

- Ενίσχυση του σήματος για να φτάσει στο εύρος 0-3 V
- Μετατροπή του σειριακού πρωτοκόλλου που χρησιμοποιεί το Beaglebone Black σε RS485 προκειμένου να είναι εφικτή η επικοινωνία με το synthesizer.



Εικόνα 22: Το ειδικό Beaglebone Cape που χρησιμοποιήθηκε

3.3 Πλατφόρμα πληροφοριακού συστήματος

Σε ό,τι αφορά τη δημιουργία της frond-end πλατφόρμας διαχείρισης του συστήματος και απεικόνισης των διαγραμμάτων χρησιμοποιήθηκαν οι γλώσσες HTML, Javascript και PHP, καθώς αποτελούν τρεις ευρέως διαδεδομένες γλώσσες στην κατασκευή ιστοσελίδων, οι οποίες είναι σχετικά απλές στη χρήση τους και κάλυπταν απόλυτα τις περισσότερες ανάγκες μας. Για τη βάση δεδομένων του συστήματός μας χρησιμοποιήθηκε η MySQL.

Σε ό,τι αφορά την ανάπτυξη του κώδικα για τη λήψη των μετρήσεων, την επικοινωνία με το synthesizer και την εκτέλεση των διαφόρων υπολογισμών επιλέχθηκε αυτός να μην γραφτεί στο μεγαλύτερο μέρος του σε κάποια native γλώσσα προγραμματισμού, όπως πχ η C, αλλά να ακολουθηθεί το μοντέλο του Dataflow Programmating. Στο κεφάλαιο 2 αναφέραμε μερικές πλατφόρμες IoT που υποστηρίζουν το συγκεκριμένο μοντέλο προγραμματισμού. Από αυτές επιλέχθηκε εν τέλει το **Node-RED**. Οι λόγοι για τους οποίους υπερίσχυσε έναντι των υπολοίπων πλατφορμών είναι:

- Αποτελεί μια πλατφόρμα "ώριμη", η οποία έχει χρησιμοποιηθεί σε πολλά IoT projects και υποστηρίζεται από μια μεγάλη κοινότητα, η οποία είναι αρκετά ενεργή.
- Από πλευράς απαιτήσεων σε hardware δεν είναι ιδιαίτερα απαιτητική και μπορεί να τρέξει σε συσκεύες edge, όπως το Beaglebone που εμείς χρησιμοποιήσαμε.
- Είναι μία πλατφόρμα εύκολη στην εγκατάστασή της και το περιβάλλον χρήσης της είναι εύκολο και φιλικό προς τον χρήστης, επιταχύνοντας τη διαδικασία ανάπτυξης λογισμικού και ολοκλήρωσης του project.
- Παρέχει πληθώρα έτοιμων κόμβων για πολλές διαφορετικές λειτουργίες από την επικοινωνία με σειριακή θύρα που εμείς χρειαζόμασταν μέχρι την αποθήκευση δεδομένων σε βάση. Έτσι, ο χρήστης διευκολύνεται και χρειάζεται να γράψει λιγότερο κώδικα.
- Είναι μια πλατφόρμα διαρκώς επεκτάσιμη από άποψη λειτουργιών και κόμβων, καθώς η κοινότητα που την χρησιμοποιεί συνεισφέρει με νέους κόμβους και flows.
- Στηρίζεται στη γλώσσα Javascript, η οποία είναι εύκολη στην εκμάθηση και χρησιμοποιείται ευρέως.
- Υπάρχει ειδικό πακέτο κόμβων που αφορούν τη λειτουργία της πλακέτας του Beaglebone που εμείς χρησιμοποιήσαμε.

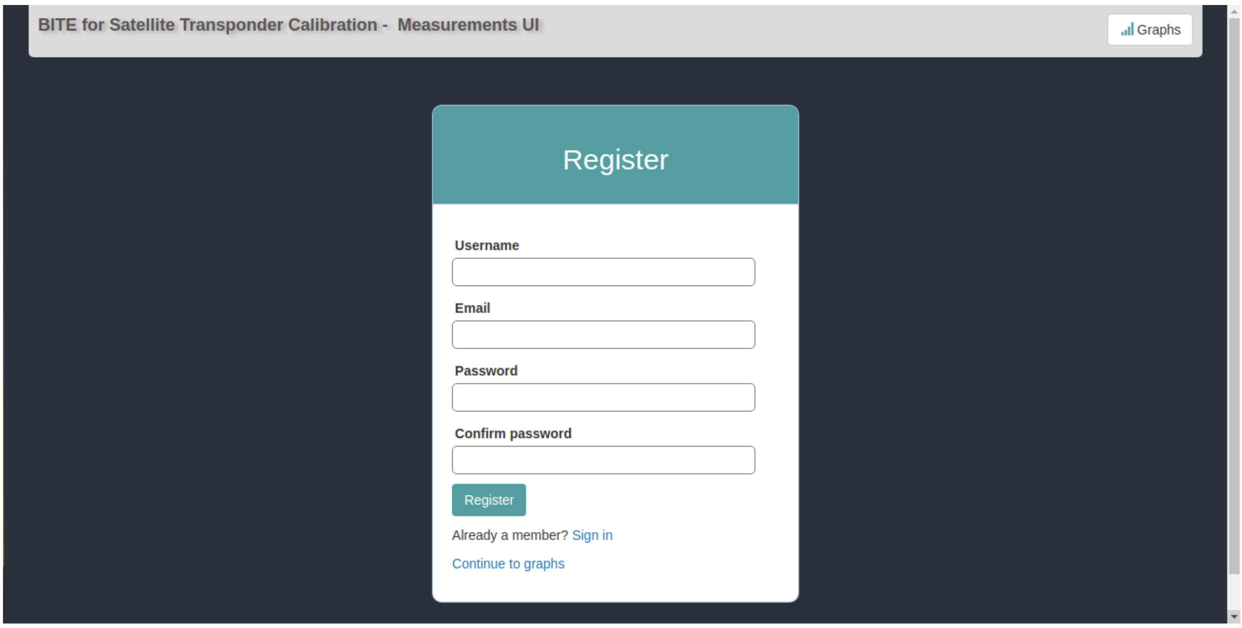
Κεφάλαιο 4ο
Περιγραφή της υλοποίησης

Στο κεφάλαιο αυτό θα περιγράψουμε την υλοποίηση του συστήματος για τη βαθμονόμηση του transponder και θα εξηγήσουμε τη λειτουργία του. Αυτό θα γίνει τόσο σε ό,τι αφορά το front-end μέρος του συστήματος, όσο και σε ό,τι αφορά το back-end.

4.1 Διαδικτυακή πλατφόρμα πληροφοριακού συστήματος

Το front-end μέρος του συστήματος πρόκειται ουσιαστικά για ένα σύνολο ιστοσελίδων μέσω των οποίων ο χειριστής του transponder μπορεί τόσο να εκκινήσει τη διαδικασία λήψης μετρήσεων για τη βαθμονόμηση του επιλέγοντας τις επιθυμητές παραμέτρους κάθε φορά, όσο και να προβάλει τα αποτελέσματα του τελευταίου πειράματος που έχει εκτελεστεί.

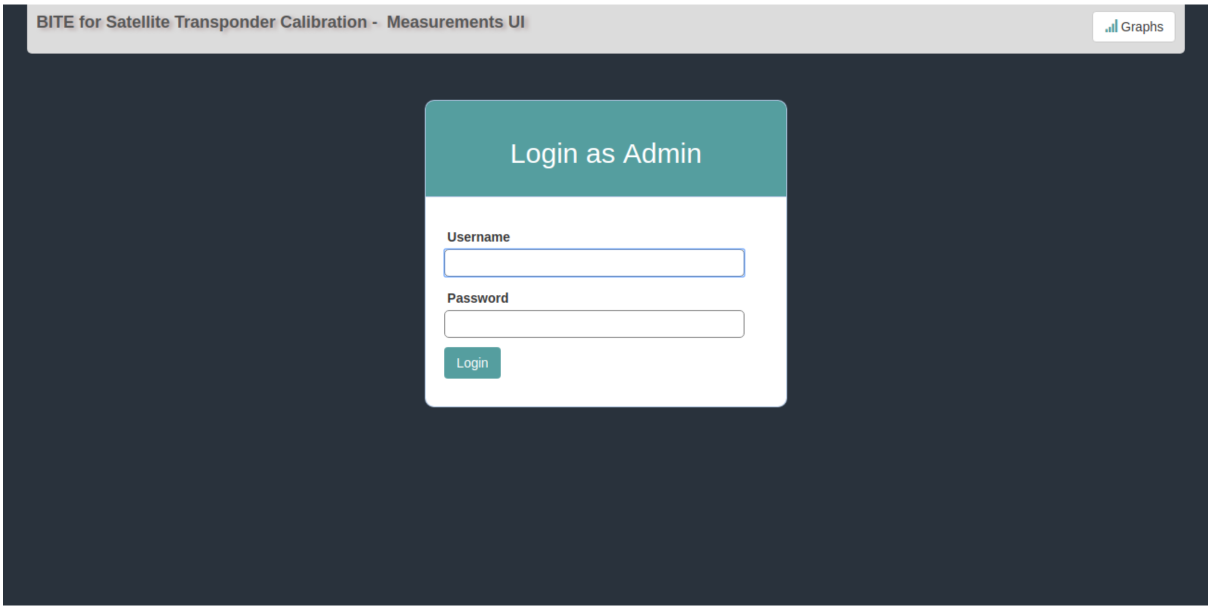
Αρχικά ένας νέος χρήστης εγγράφεται στη βάση χρηστών της πλατφόρμας μέσω της σελίδας register.php. Όπως φαίνεται στην παρακάτω εικόνα ο χρήστης επιλέγει το όνομα χρήστη που θέλει και τον κωδικό πρόσβασης. Το πεδίο email δίνει τη δυνατότητα σε μελλοντική επέκταση της πλατφόρμας να αποστέλλεται email επιβεβαίωσης στο χρήστη. Επιπλέον, υπάρχει η δυνατότητα αντί για την εγγραφή του χρήστη να γίνεται απευθείας προβολή των διαγραμμάτων του τελευταίου πειράματος είτε πατώντας το κουμπί “Graphs” είτε το σύνδεσμο “Continue to graphs”. Τέλος, σε περίπτωση που ο χρήστης είναι ήδη εγγεγραμμένος μπορεί απευθείας να μεταβεί στη σελίδα εισόδου μέσω του συνδέσμου “Sign in”.



Εικόνα 23: Η σελίδα εγγραφής νέου χρήστη register.php

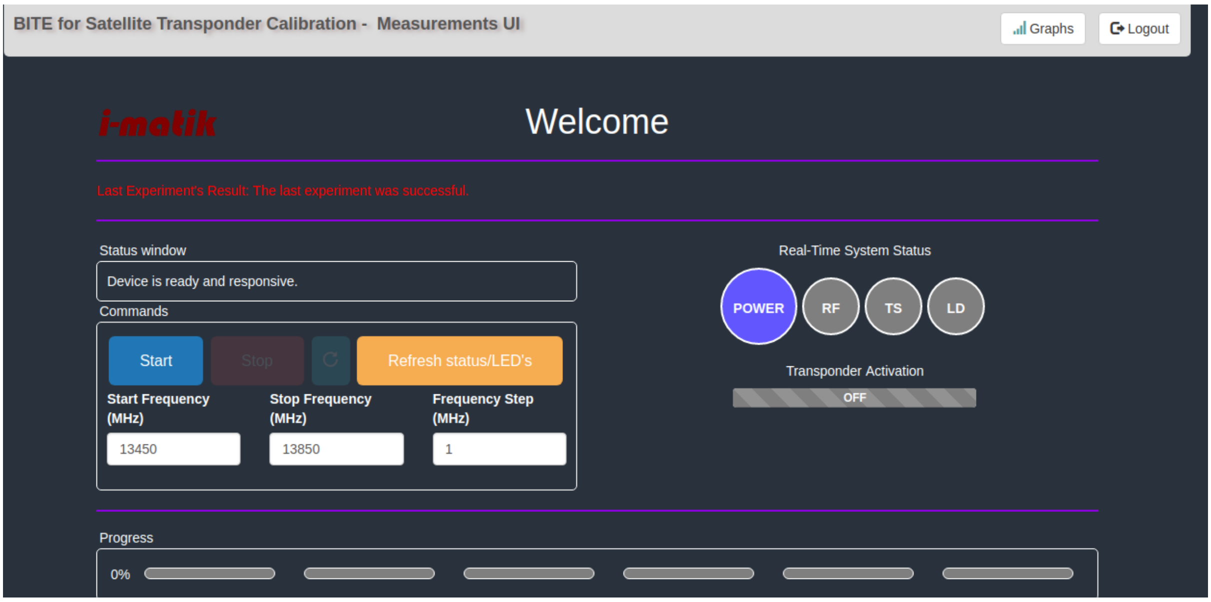
Αν κάποιος είναι ήδη εγγεγραμμένος χρήστης, μέσω της σελίδας login.php, η οποία φαίνεται παρακάτω, μπορεί να εισέλθει στην πλατφόρμα και να εκκινήσει ένα νέο πείραμα.

Επιπλέον, αντί της εισόδου μπορεί και πάλι μέσω του κουμπιού “Graphs” να προβάλει τα διαγράμματα του τελευταίου πειράματος.



Εικόνα 24: Η σελίδα εισόδου login.php

Με την επιτυχής ολοκλήρωση του login ο χρήστης μεταφέρεται στη σελίδα απ’ όπου μπορεί να εκκινήσει ένα νέο πείραμα. Μέρος της σελίδας αυτής φαίνεται στην εικόνα 25. Στη σελίδα αυτή εμφανίζονται ακόμα τα διαγράμματα της μέσης τιμής της τάσης εξόδου του transponder και της καθυστέρησης ομάδος για τις τάσεις αυτές στις αντίστοιχες συχνότητες. Μάλιστα τα διαγράμματα αυτά ανανεώνονται σε πραγματικό χρόνο την ώρα εκτέλεσης του πειράματος, όπως φαίνεται στις εικόνες που ακολουθούν στην επόμενη σελίδα.



Εικόνα 25: Η σελίδα εκκίνησης πειραμάτων και απεικόνισης της εξέλιξής τους

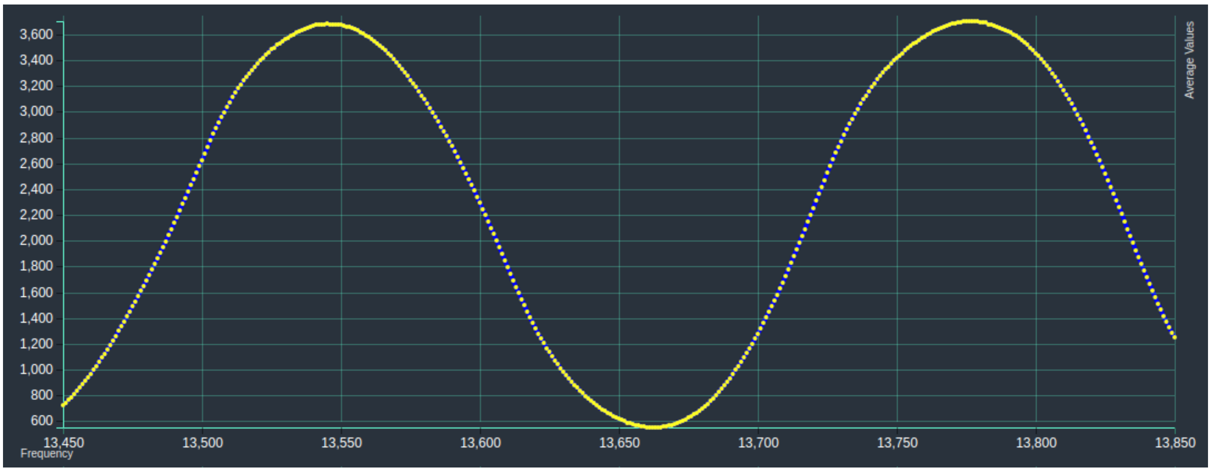
Εικόνα 26: Απεικόνιση της εξέλιξης της εκτέλεσης ενός πειράματος

Εικόνα 27: Ολοκλήρωση της εκτέλεσης του προηγούμενου πειράματος

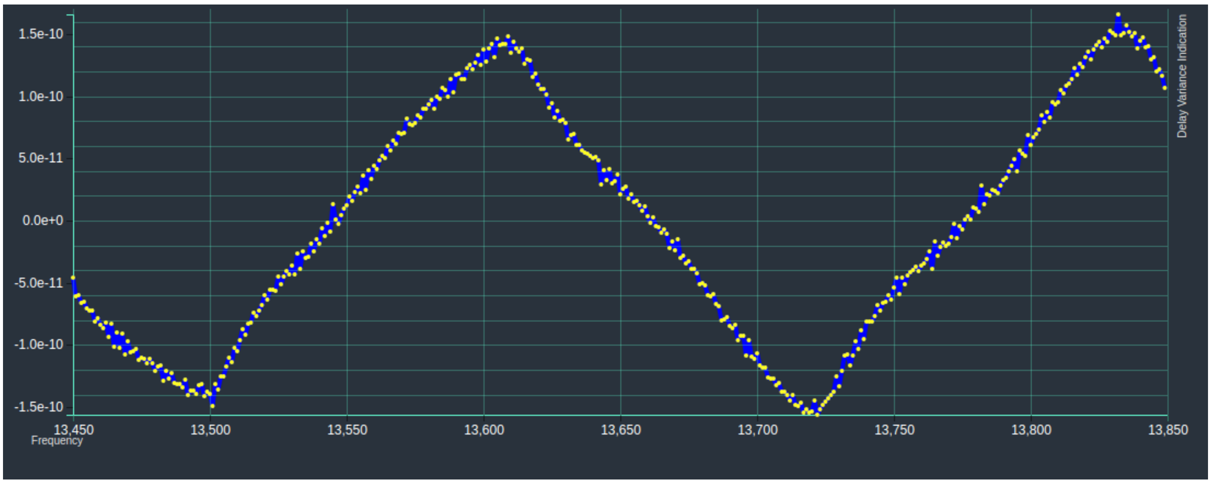
Στη σελίδα εκκίνησης πειράματος ο χρήστης μπορεί να ορίσει το εύρος των συχνοτήτων για τις οποίες θα εκτελεστεί το πείραμα, καθώς και το βήμα μεταξύ δύο διαδοχικών συχνοτήτων. Η ελάχιστη συχνότητα είναι τα 13000 MHz και η μέγιστη τα 14000 MHz, ενώ το βήμα μπορεί να πάρει και δεκαδικές τιμές έως τα 0,001 MHz. Μόλις ο χρήστης ορίσει τα παραπάνω μεγέθη, μπορεί να εκκινήσει το πείραμα πατώντας το κουμπί “Start”. Ο χρήστης μπορεί να διακόψει το πείραμα ανά πάσα στιγμή πιέζοντας το κουμπί “Stop”. Με την ολοκλήρωση του πειράματος ο χρήστης μπορεί να επαναλάβει τη διαδικασία πατώντας το κουμπί με το κυκλικό βέλος δεξιά του “Stop”. Η πρόοδος της εξέλιξης του πειράματος απεικονίζεται σε μια μπάρα προόδου μαζί με το ποσοστό των μετρήσεων που έχουν ληφθεί μέχρι στιγμής, ενώ μηνύματα για το στάδιο στο οποίο βρίσκεται το πείραμα εμφανίζονται στο πλαίσιο “Status window”. Το αποτέλεσμα του πειράματος και αν ήταν επιτυχές ή όχι απεικονίζεται με κόκκινα γράμματα στην κορυφή της σελίδας. Τέλος, στα δεξιά της σελίδας υπάρχουν ορισμένες ενδείξεις οι οποίες ενημερώνονται σε πραγματικό χρόνο και απεικονίζουν το αν ο transponder έχει ενεργοποιηθεί και την κατάσταση των leds που υπάρχουν στην πρόσοψη του εξοπλισμού. Όπως φαίνεται και από το υπόμνημα της επόμενης εικόνας, αυτά τα leds αντιστοιχούν κατά σειρά στην ύπαρξη τροφοδοσίας, στην ενεργοποίηση του synthesizer, στην ενεργοποίηση των διακοπτών για τις τροφοδοσίες 12V και 5V και στην ένδειξη Lock του synthesizer. Ο χρήστης μπορεί και χειροκίνητα να ζητήσει την κατάσταση των leds πατώντας το κουμπί "Refresh status/LED's".

Εικόνα 28: Απεικόνιση των leds στην πρόσοψη του εξοπλισμού

Στις επόμενες δύο εικόνες απεικονίζονται τα διαγράμματα κατά την ολοκλήρωση της εκτέλεσης του πειράματος, τα οποία δεν φαίνονταν ολόκληρα παραπάνω. Εδώ να σημειώσουμε ότι οι περισσότερες από τις δυναμικές λειτουργίες της σελίδας αυτής έχουν υλοποιηθεί με τη βοήθεια της Javascript, ενώ έχει χρησιμοποιηθεί και Ajax για την ανάκτηση της κατάστασης των leds και του εξοπλισμού. Για τα διαγράμματα χρησιμοποιήθηκε η βιβλιοθήκη d3 η οποία και πάλι στηρίζεται στη Javascript.

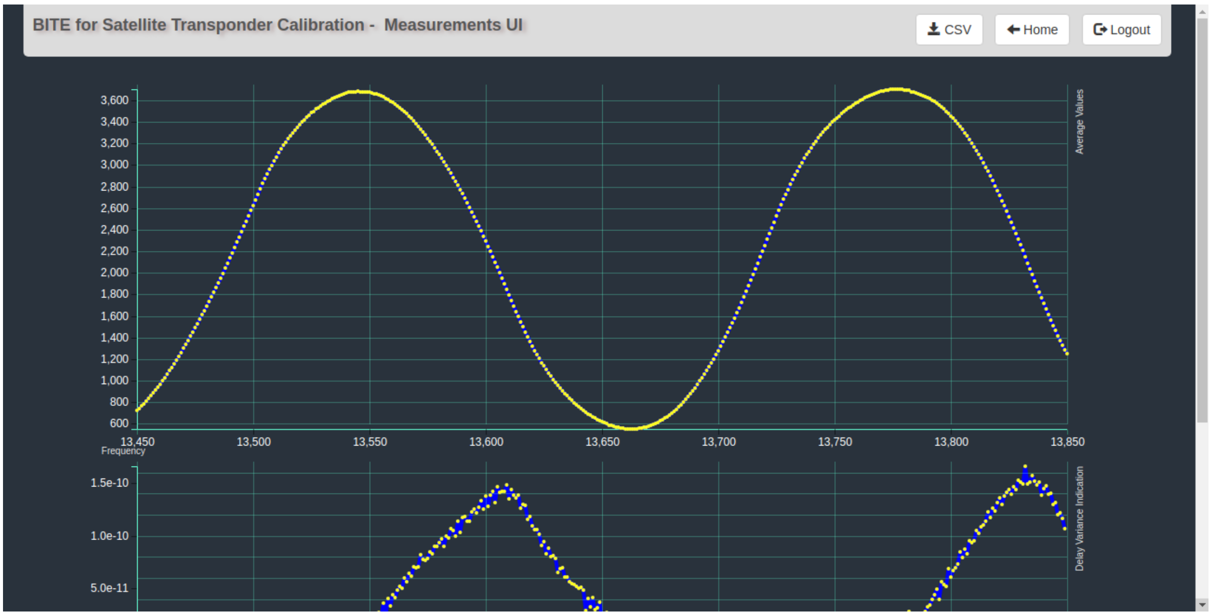


Εικόνα 29: Η γραφική παράσταση της μέσης τιμής των τάσεων εξόδου του transponder για το προηγούμενο πείραμα



Εικόνα 30: Η γραφική παράσταση της καθυστέρησης ομάδος στην έξοδο του transponder για το προηγούμενο πείραμα

Κλείνοντας αυτή την ενότητα να πούμε ότι ο χρήστης μπορεί αντί να εκτελέσει κάποιο πείραμα, να επιλέξει να προβάλει τα διαγράμματα από την εκτέλεση του πιο πρόσφατου πειράματος πατώντας το κουμπί “Graphs” δεξιά πάνω στη σελίδα. Έτσι, θα μεταβεί στη σελίδα η οποία φαίνεται στην επόμενη εικόνα και στην οποία απεικονίζονται τα δύο διαγράμματα για την μέση τιμή της τάσης και την καθυστέρηση ομάδας. Ακόμα, σε αυτή τη σελίδα δίνεται η δυνατότητα στο χρήστη να κατεβάσει σε αρχείο μορφής csv τα δεδομένα της εκτέλεσης του πειράματος για ενδεχόμενη περαιτέρω επεξεργασία.

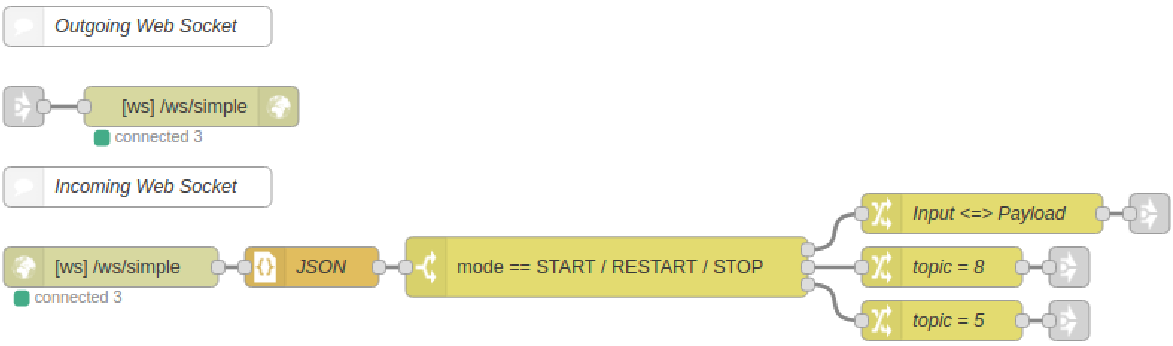


Εικόνα 31: Η σελίδα απεικόνισης των διαγραμμάτων του τελευταίου πειράματος που εκτελέστηκε

4.2 Λήψη μετρήσεων και εκτέλεση υπολογισμών στο Node-Red

Στην ενότητα αυτή θα περιγράψουμε το πώς λειτουργεί ο κώδικας στο Node-Red μέσω του οποίου γίνεται η λήψη των μετρήσεων από τον μετατροπέα ADC του Beaglebone και εν συνεχεία η εκτέλεση των σχετικών υπολογισμών, οι οποίοι αναπαρίστανται στο website που περιγράψαμε παραπάνω.

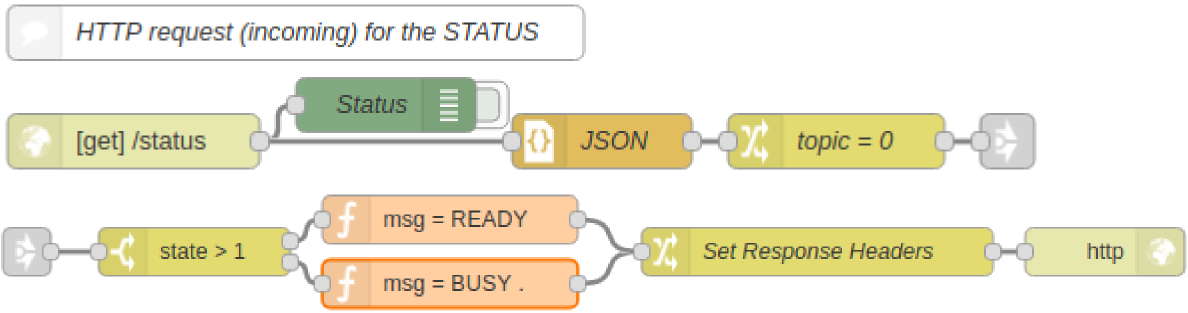
Το κυριότερο μέσο επικοινωνίας ανάμεσα στα flows του Node-Red και στο website είναι ένα websocket. Στην παρακάτω εικόνα φαίνονται τόσο το μέρος του websocket που αφορά την έξοδο δεδομένων από το Node-Red, όσο και το websocket που σχετίζεται με την αποστολή δεδομένων στο Node-Red από το website.



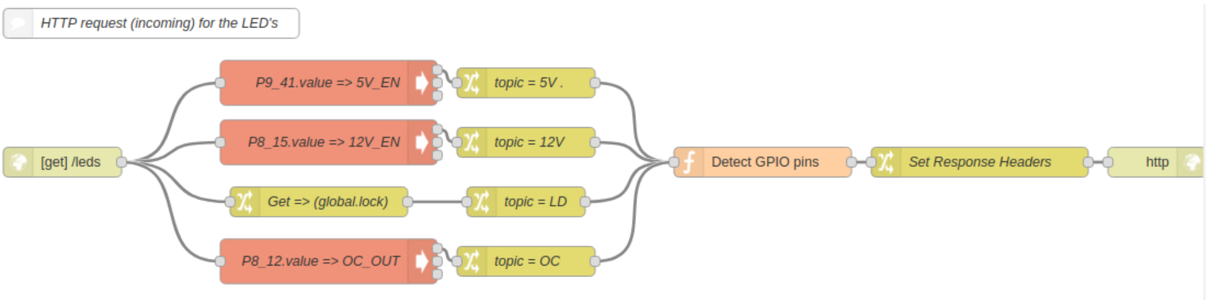
Εικόνα 32: Απεικόνιση του websocket εισόδου και εξόδου

Το websocket εξόδου χρησιμοποιείται για την αποστολή ενημερώσεων προς το website σχετικά με την κατάσταση στην οποία βρίσκεται η εκτέλεση του πειράματος, καθώς και το ποσοστό των μετρήσεων που έχουν εκτελεστεί. Επιπλέον, στέλνονται τα δεδομένα που αφορούν τη δημιουργία των διαγραμμάτων σε πραγματικό χρόνο. Το websocket εισόδου λαμβάνει από το website ένα μήνυμα σε μορφή Javascript object κάθε φορά που ο χρήστης πατάει το πλήκτρο “Start”, το πλήκτρο “Restart” ή το πλήκτρο “Stop”. Αφού το μήνυμα μετατραπεί σε JSON string, περνάει από έναν κόμβο Switch και ανάλογα με την τιμή που έχει το πεδίο msg.payload.mode εκτελούνται οι σχετικές ενέργειες (ενημέρωση του πεδίου msg.topic το οποίο θα πούμε στη συνέχεια που χρησιμοποιείται και ενημέρωση του msg.payload με τις τιμές που έχει δώσει ο χρήστης στο website σε περίπτωση που έχει πατηθεί το πλήκτρο “Start”). Ο κόμβος εισόδου/εξόδου με το βέλος ονομάζεται link και χρησιμεύει για τη σύνδεση ανάμεσα σε δύο ή περισσότερα μέρη του ίδιου ή διαφορετικών flows.

Πέραν του websocket επικοινωνία ανάμεσα στο website και το Node-Red γίνεται μέσω δύο κόμβων http request που αφορούν την κατάσταση στην οποία βρίσκεται το flow καθώς και την κατάσταση των leds. Οι κόμβοι αυτοί φαίνονται στις εικόνες της επόμενης σελίδας. Ο πρώτος λαμβάνει ένα HTTP request από το website, το Javascript object μετατρέπεται σε JSON string και ο κόμβος change θέτει το msg.topic στην κατάλληλη τιμή. Το δεύτερο μέρος του flow που αφορά το status λαμβάνει ένα μήνυμα και αν το πεδίο state έχει τιμή μεγαλύτερη του 1 στέλνεται στο website το μήνυμα READY, διαφορετικά το μήνυμα BUSY. Σε ό,τι αφορά την λήψη της κατάστασης των leds, όπως φαίνεται στην εικόνα 34, η τιμή τους λαμβάνεται με τη βοήθεια των κόμβων exec, οι οποίοι μας επιτρέπουν να εκτελέσουμε μια εντολή στο σύστημα σαν να είμαστε σε περιβάλλον terminal (εδώ την *cat /sys/class/gpio/gpio*/value*). Ο κόμβος function με όνομα "Detect GPIO pins" συγκεντρώνει τις τιμές για όλα τα leds και εκτελεί κώδικα ο οποίος τις επιστρέφει μόνο όταν έχουν έρθει όλες.

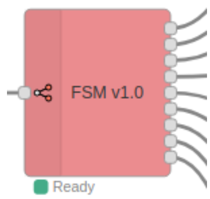


Εικόνα 33: Αίτηση και απόκριση για την κατάσταση του flow

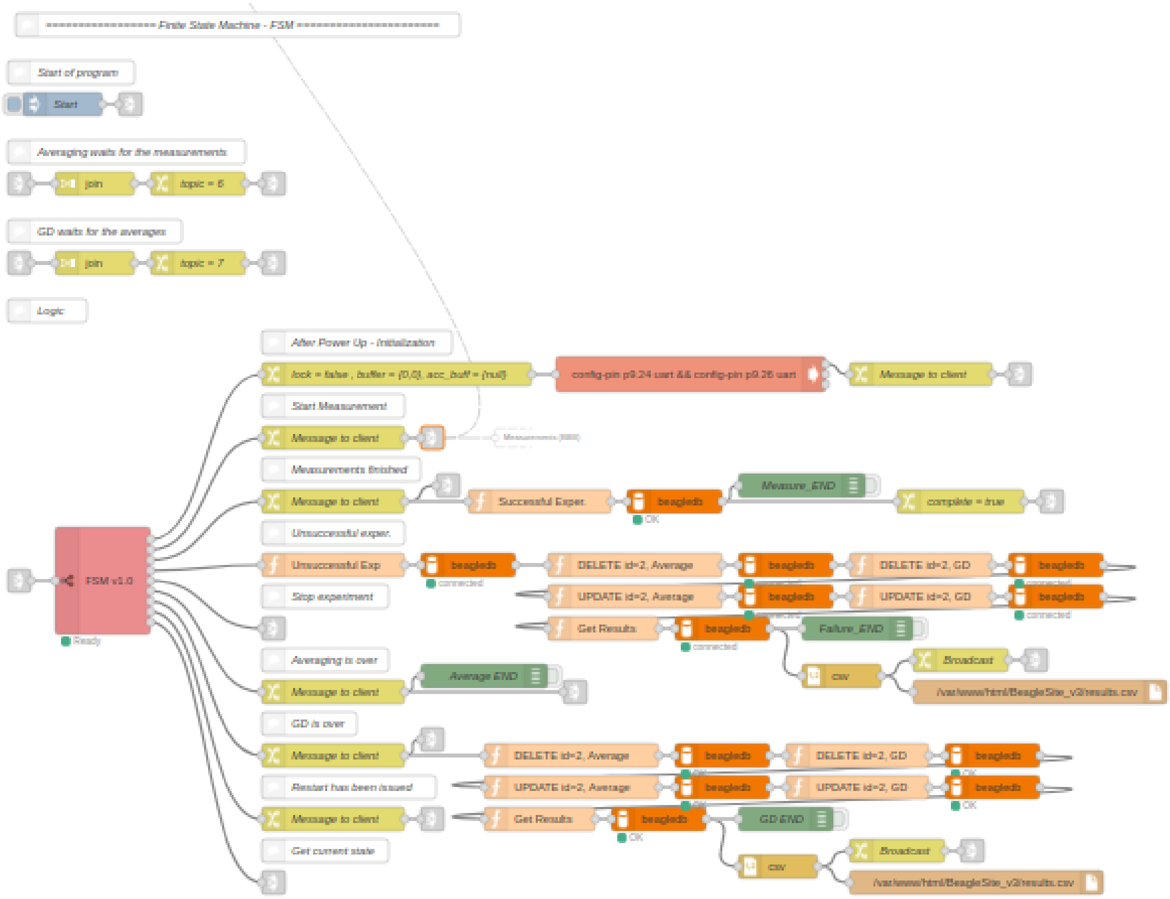


Εικόνα 34: Αίτηση και απόκριση για την κατάσταση των leds.

Αφού περιγράψαμε πώς γίνεται η επικοινωνία ανάμεσα στο Node-Red και το website, ας δούμε τώρα πώς λειτουργεί ουσιαστικά ο υπόλοιπος κώδικας των flows που έχουν δημιουργηθεί. Πρόκειται ουσιαστικά για μια μηχανή πεπερασμένων καταστάσεων (finite state machine ή FSM) η οποία καθορίζει σε ποια κατάσταση βρίσκεται κάθε στιγμή ο κώδικας σε ποια κατάσταση πρόκειται να μεταβεί. Για τον σκοπό αυτό υλοποιήθηκε σε javascript και html ένας κόμβος FSM, για να εξυπηρετήσει τις ανάγκες μας. Ο κόμβος αυτός λαμβάνει μηνύματα και με βάση έναν πίνακα μεταβάσεων και την τιμή του πεδίου msg.topic καθορίζει σε ποια κατάσταση πρόκειται να μεταβεί η εκτέλεση του πειράματος. Το σύνολο των κόμβων που απαρτίζουν το FSM αυτό φαίνεται στην εικόνα της επόμενης σελίδας. Θα αναλύσουμε το πώς λειτουργεί το κάθε τμήμα του ακολουθώντας την πορεία του κατά την εκτέλεση ενός πειράματος.



Εικόνα 35: Ο κόμβος FSM που δημιουργήθηκε

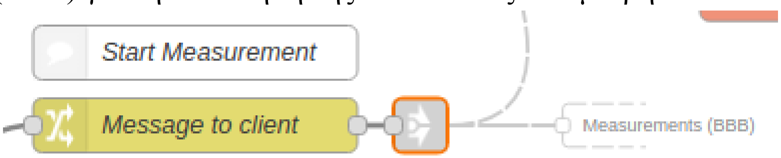


Εικόνα 36: Εποπτική απεικόνιση του συνόλου των κόμβων που αποτελούν το FSM

Η εκκίνηση του προγράμματος γίνεται αυτόματα από τον κόμβο inject με όνομα "Start", ο οποίος ταυτόχρονα θέτει το πεδίο msg.topic στην τιμή 1 και στέλνει το μήνυμα στον κόμβο FSM. Αυτός με τη σειρά του επιλέγει την πρώτη έξοδό του, η οποία αφορά αρχικοποιήσεις διαφόρων global μεταβλητών, την ενεργοποίηση της σειριακής θύρας με τις δύο εντολές config-pin και την αποστολή του μηνύματος στο website ότι πλέον το flow είναι έτοιμο να μεταβεί σε κατάσταση μετρήσεων. Το FSM πλέον έχει μεταβεί στο state 2. Οι κόμβοι αυτοί φαίνονται στην εικόνα 37. Η εκτέλεση του προγράμματος περνά από αυτούς μόνο μία φορά κατά το στάδιο της ενεργοποίησης του εξοπλισμού και δεν επιστρέφει ξανά σε αυτούς, εκτός αν απενεργοποιηθεί και ενεργοποιηθεί ξανά ο εξοπλισμός.

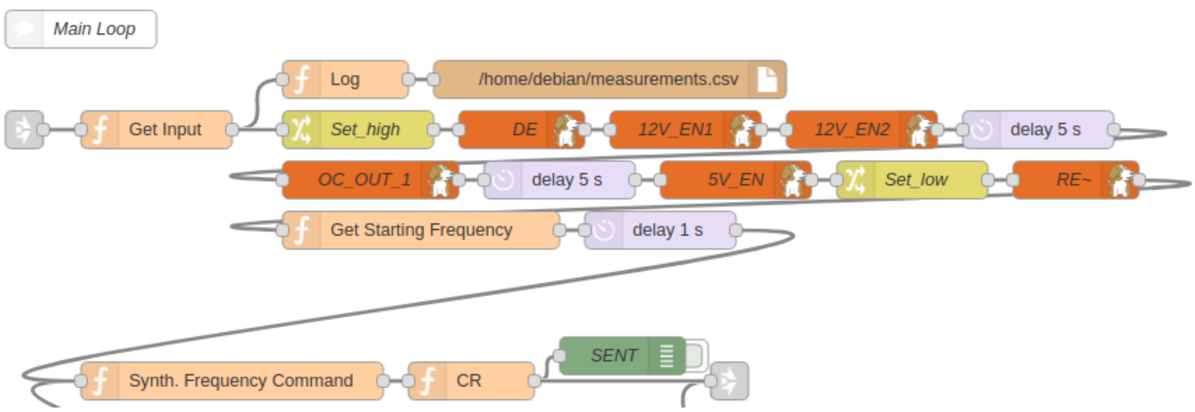
Εικόνα 37: State 1 - Αρχικοποίηση

Όταν ολοκληρωθούν οι αρχικοποιήσεις, ενημερώνεται το Status window στο website και ο χρήστης πλέον μπορεί να εκκινήσει κάποιο πείραμα αφού το κουμπί “Start” πλέον είναι ενεργοποιημένο. Όπως αναφέραμε παραπάνω, μετά το πάτημα του “Start” εκτελείται ο κώδικας του flow στο websocket εισόδου και στέλνεται στην είσοδο του FSM ένα μήνυμα με topic 2 και τα δεδομένα εισόδου που έδωσε ο χρήστης στο website για το πείραμα. Το FSM επιλέγει τη δεύτερη έξοδό του, η οποία οδηγεί στους κόμβους που φαίνονται παρακάτω. Ο κόμβος change θέτει το msg.payload στην JSON τιμή {"state":3} που υποδηλώνει ότι το το FSM μεταβαίνει πλέον στο state 3 και την global μεταβλητή synth_rejected στην τιμή 0, η οποία παίρνει την τιμή 1 μόνο αν το synthesizer γυρίσει Reject. Τέλος, στέλνονται μέσω του κόμβου link δύο μηνύματα, ένα προς το websocket εξόδου για την ενημέρωση του Status window στο website και ένα προς το flow Measurements (BBB) για την εκκίνηση της διαδικασίας των μετρήσεων.



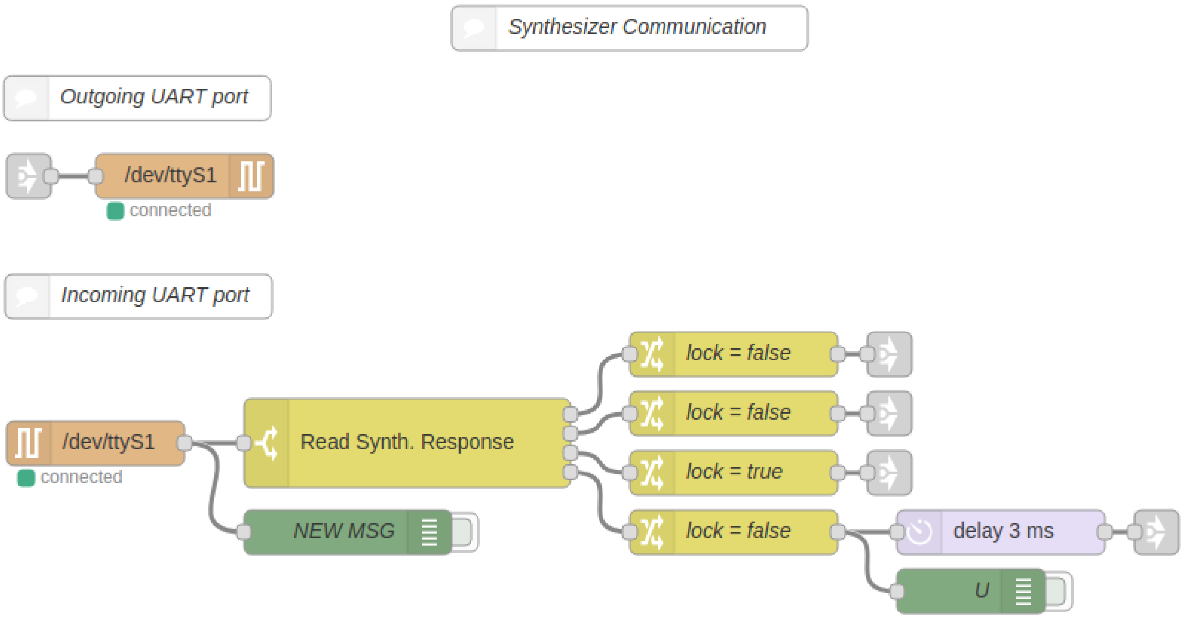
Εικόνα 38: Μετάβαση σε κατάσταση μετρήσεων

Το flow τώρα μεταβαίνει στο κομμάτι που αφορά την λήψη των μετρήσεων. Όπως φαίνεται στην εικόνα 39, ο κόμβος function “Get Input” λαμβάνει και θέτει στις τοπικές flow μεταβλητές fstart, fstop και fstep τις τιμές που έχω δώσει ο χρήστης από το website, αρχικοποιεί στις κατάλληλες τιμές μια μεταβλητή “φρουρό” για το αν ο χρήστης έχει πατήσει το “Stop” και τρεις μεταβλητές που αφορούν το ποσοστό των μετρήσεων που έχουν ληφθεί και λαμβάνει και θέτει στη flow μεταβλητή timestamp τη χρονική στιγμή που ξεκινούν οι μετρήσεις. Αυτό συμβαίνει γιατί ο ενισχυτής που χρησιμοποιείται δεν πρέπει να μένει αναμμένος περισσότερο από 5 λεπτά. Στη συνέχεια, δημιουργείται ένα αρχείο σε μορφή csv όπου αποθηκεύονται οι τιμές των μετρήσεων που λαμβάνονται σε περίπτωση που χρειαστεί να τις δει κάποιος (το αρχείο αυτό επανεγγράφεται στην αρχή κάθε πειράματος, δεν δημιουργείται νέο) και ενεργοποιούνται κάποιοι διακόπτες μέσω ειδικών κόμβων του Node-Red για τον έλεγχο των pins του Beaglebone.



Εικόνα 39: Διαδικασία ενεργοποίησης εξοπλισμού για λήψη μετρήσεων και αποστολή στο synthesizer της εντολής συχνότητας

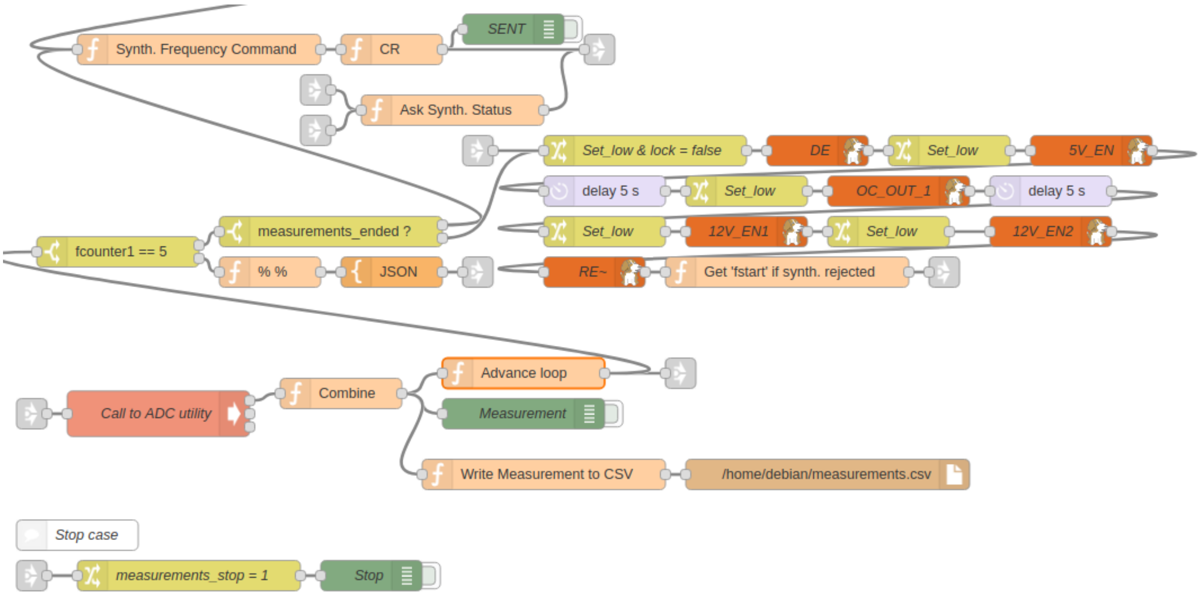
Ο function κόμβος Get Starting Frequency θέτει στο πεδίο msg.fstart την αρχική συχνότητα και μετά από 1 δευτερόλεπτο δημιουργείται στον κόμβο function Synth. Frequency Command η εντολή που περιμένει το synthesizer για να ρυθμιστεί στη συχνότητα αυτή. Ο επόμενος κόμβος προσθέτει απλά το carriage return στο τέλος της εντολής και στέλνει το μήνυμα προς το flow που αφορά την επικοινωνία με το synthesizer.



Εικόνα 40: Οι κόμβοι που αφορούν την επικοινωνία με το synthesizer

Στην παραπάνω εικόνα φαίνεται το flow που αφορά την επικοινωνία με το synthesizer. Σε ό,τι αφορά τα μηνύματα που στέλνονται προς το synthesizer, αυτά λαμβάνονται από έναν κόμβο link και προωθούνται απευθείας στη σειριακή θύρα S1 του Beaglebone. Αντίστοιχα οι απαντήσεις του synthesizer λαμβάνονται από τη σειριακή θύρα S1 και στη συνέχεια μέσω ενός κόμβου switch γίνεται έλεγχος για να καθοριστεί τι είδους απάντηση παραλήφθηκε. Συγκεκριμένα ελέγχεται αν η απάντηση του synthesizer περιέχει το γράμμα A, R, L ή U που αντιστοιχούν στις περιπτώσεις που το synthesizer αποδέχεται την εντολή που στάλθηκε, απορρίπτει την εντολή που στάλθηκε, απαντά ότι έχει κλειδώσει στην ζητούμενη συχνότητα ή ότι δεν έχει κλειδώσει. Στη συνέχεια τίθεται ανάλογα η global μεταβλητή lock ανάλογα με το αν το synthesizer έχει κλειδώσει ή όχι. Επιπλέον σε περίπτωση που το synthesizer απαντήσει με απόρριψη της εντολής, η global μεταβλητή synth_rejected παίρνει την τιμή 1. Τέλος, στην περίπτωση που το synthesizer απαντήσει ότι δεν έχει κλειδώσει υπάρχει και ένας κόμβος που επιβάλει καθυστέρηση 3ms.

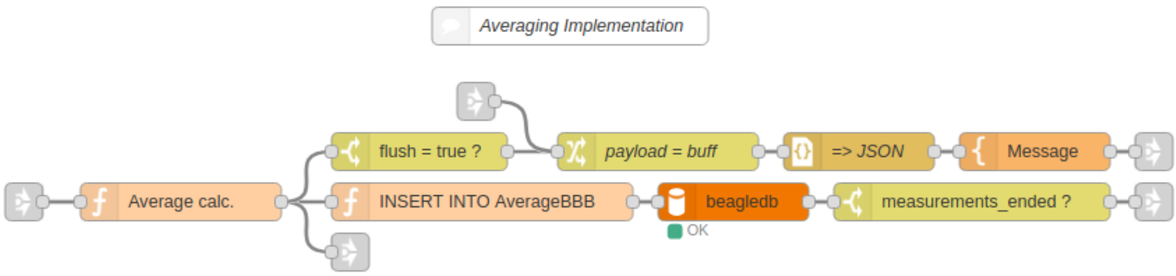
Αν το synthesizer επιστρέψει reject, τότε η εκτέλεση του πειράματος διακόπτεται και το flow συνεχίζεται με την απενεργοποίηση των διακοπτών που ενεργοποιήθηκαν κατά την έναρξη της διαδικασίας λήψης των μετρήσεων. Σε αυτή την περίπτωση πριν η εκτέλεση του flow μεταβεί στο κομμάτι που αφορά την ενημέρωση του FSM λαμβάνεται μέσω ενός κόμβου function η συχνότητα στην οποία το πείραμα διακόπηκε, όπως φαίνεται στην εικόνα 41 στην επόμενη σελίδα.



Εικόνα 41: Το σύνολο των κόμβων που αφορούν την λήψη των μετρήσεων, τη λήψη των απαντήσεων του synthesizer, την απενεργοποίηση των διακοπτών στο τέλος της διαδικασίας, κτλ.

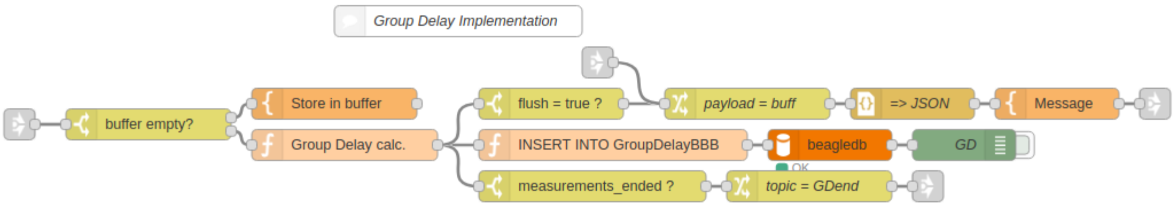
Αν το synthesizer επιστρέψει accept στην εντολή συχνότητας που του αποστείλαμε, τότε ζητάμε να μάθουμε το status του synthesizer. Το μήνυμα για αυτή την εντολή διαμορφώνεται κατάλληλα στον function κόμβο Ask Synth. Status. Αν το synthesizer επιστρέψει ότι δεν έχει κλειδώσει, επαναλαμβάνουμε την ερώτηση για το status. Αν επιστρέψει ότι έχει κλειδώσει στη ζητούμενη συχνότητα, τότε ξεκινά η λήψη των μετρήσεων. Η λήψη των μετρήσεων γίνεται εκτός Node-Red με την εκτέλεση ενός προγράμματος σε C που γράφτηκε για το σκοπό αυτό. Δεν χρησιμοποιήθηκε ο ειδικός κόμβος του Node-Red για την κλήση στον μετατροπέα ADC για λόγους ταχύτητας. Η κλήση στο πρόγραμμα που λαμβάνει τις μετρήσεις γίνεται με τον κόμβο exec Call to ADC utility. Ο κόμβος function Combine λαμβάνει την έξοδο του προγράμματος και ενημερώνει τα πεδία freq, volt και fstart με τη συχνότητα για την οποία λήφθηκαν οι μετρήσεις, με τον πίνακα των 30 μετρήσεων που λήφθηκαν και τη συχνότητα στην οποία βρίσκεται η εκτέλεση του πειράματος αυτή τη στιγμή αντίστοιχα. Στη συνέχεια από τη μία γράφονται στο log file οι μετρήσεις που λήφθηκαν στη δεδομένη συχνότητα, με τον κόμβο function Write Measurements to CSV να διαμορφώνει το μήνυμα που πρέπει να σταλεί για την εγγραφή των τιμών στο αρχείο. Από την άλλη, παράλληλα ο κόμβος function Advance loop εκτελεί τις διάφορες εντολές που χρειάζονται για να προχωρήσει η διαδικασία λήψης των μετρήσεων στην επόμενη συχνότητα. Πιο συγκεκριμένα, αυξάνει με βάση το step τη συχνότητα, ενημερώνει τους counter που χρησιμοποιούνται για τον υπολογισμό του ποσοστού των μετρήσεων που έχουν ληφθεί και ελέγχει τη συνθήκη τερματισμού του πειράματος, η οποία επαληθεύεται αν συμβαίνει ένα από τα τρία γεγονότα: έχει ξεπεραστεί η συχνότητα τερματισμού, έχει πατήσει “Stop” ο χρήστης ή έχουν παρέλθει τα 5 λεπτά από την έναρξη του πειράματος. Στην περίπτωση που συμβαίνει ένα από τα παραπάνω το πεδίο msg.measurements_ended παίρνει την τιμή 1.

Στο σημείο αυτό ξεκινάει η παράλληλη εκτέλεση του flow της λήψης των μετρήσεων για τις υπόλοιπες συχνότητες και των flows που αφορούν τον υπολογισμό του μέσου όρου της τάσης εξόδου και του group delay για κάθε συχνότητα. Πριν προχωρήσουμε στην περιγραφή αυτών των flows ας περιγράψουμε τη λειτουργία των υπόλοιπων κόμβων του flow λήψης των μετρήσεων. Ο κόμβος switch fcounter1 == 5 βγάζει πάντα ένα μήνυμα από την πρώτη του έξοδο προς τον κόμβο switch measurements ended ?. Στην περίπτωση που ο μετρητής fcounter1 είναι ίσος με 5 βγάζει και ένα μήνυμα από την δεύτερή του έξοδο το οποίο στέλνεται μέσω του websocket στο website για την ενημέρωση του ποσοστού των μετρήσεων που έχουν ληφθεί. Ο κόμβος switch measurements ended ελέγχει την τιμή του πεδίου msg.measurements_ended και όσο έχει την τιμή 0 συνεχίζει τη λήψη μετρήσεων για την επόμενη συχνότητα. Διαφορετικά η εκτέλεση του πειράματος ολοκληρώνεται με την απενεργοποίηση των διακοπτών και τη μετάβαση στο flow του FSM για την ενημέρωσή του. Τέλος, σε περίπτωση που ο χρήστης πατήσει το πλήκτρο “Stop” στο website κατά την εκτέλεση του πειράματος τότε λαμβάνεται το σχετικό μήνυμα μέσω του websocket και στο flow των μετρήσεων η flow μεταβλητή measurements_stop παίρνει την τιμή 1, κάτι που θα οδηγήσει στον τερματισμό του πειράματος.



Εικόνα 42: Οι κόμβοι που αφορούν τον υπολογισμό του μέσου όρου της τάσης

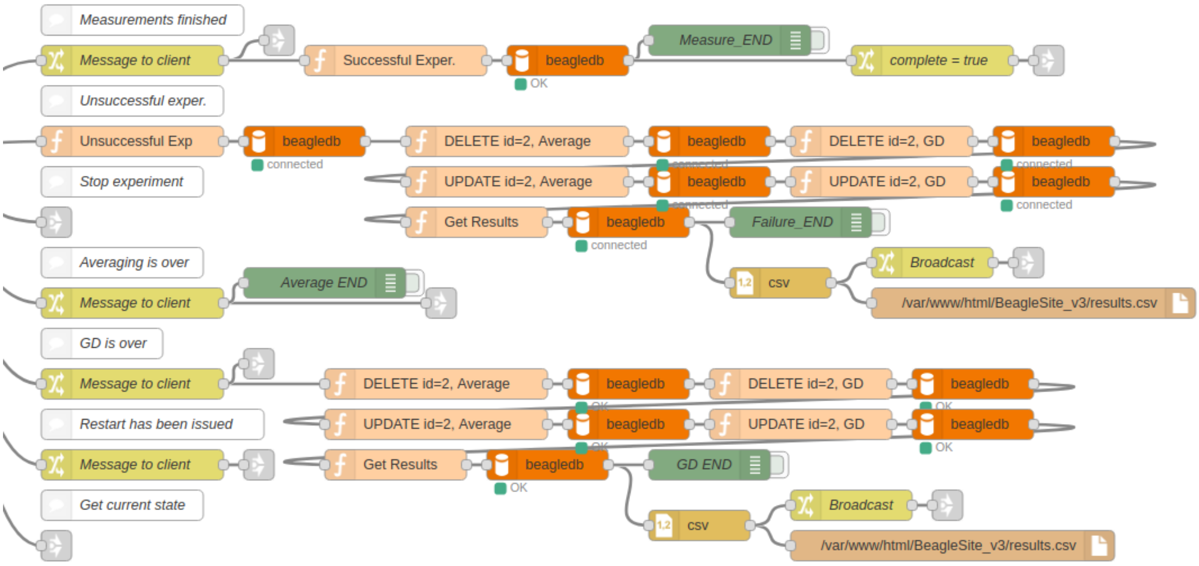
Ας περιγράψουμε τώρα τη λειτουργία του flow που αφορά τον υπολογισμό του μέσου όρου της τάσης σε μια συχνότητα, το οποίο αρχίζει να εκτελείται αμέσως μετά τη λήψη των μετρήσεων για την πρώτη συχνότητα του πειράματος. Στον function κομβο Average calc. γίνεται ο υπολογισμός της μέσης τιμής των 30 μετρήσεων, αφού αφαιρεθούν η μικρότερη και η μεγαλύτερη τιμή. Η τιμή αυτή και η συχνότητα στην οποία αντιστοιχεί αφενός τοποθετούνται σε έναν buffer που συλλέγει ανά 5 τα ζευγάρια τάσης-συχνότητας για την αποστολή τους στο website για τη δημιουργία των διαγραμμάτων σε πραγματικό χρόνο και αφετέρου τοποθετούνται στο payload ενός μηνύματος που στέλνεται για την εγγραφή τους στη βάση δεδομένων. Στην έξοδο του κόμβου function έχουμε μια τριπλή διακλάδωση. Η πρώτη διαδρομή ελέγχει εάν έχουν συμπληρωθεί 5 ζευγάρια τάσης-συχνότητας για την αποστολή τους στο website. Αν αυτό ισχύει, το msg.payload παίρνει την τιμή του buffer και αφού διαμορφωθεί κατάλληλα στέλνεται στο websocket και ο buffer αδειάζει για να λάβει τα επόμενα ζευγάρια. Η δεύτερη διαδρομή αφορά την αποθήκευση του κάθε ζεύγους στη βάση δεδομένων. Ο κόμβος switch measurements_ended ? ελέγχει εάν έχουν ολοκληρωθεί οι μετρήσεις και μόνο τότε το flow προχωράει στον κόμβο link για την ενημέρωση του FSM και την αποστολή των τελευταίων ζευγών στο website. Τέλος, η τρίτη διαδρομή αφορά τη συνέχιση της εκτέλεσης παράλληλα στο flow που αφορά τον υπολογισμό του group delay για κάθε συχνότητα.



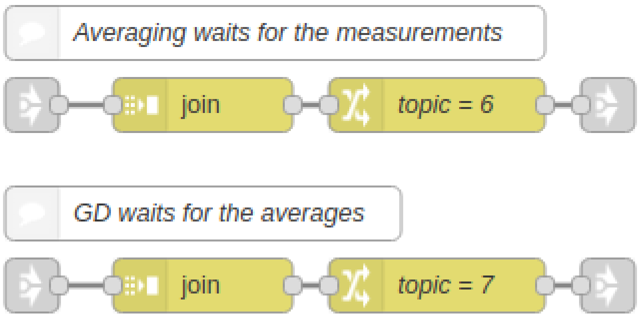
Εικόνα 43: Οι κόμβοι που αφορούν τον υπολογισμό του group delay

Σε ό,τι αφορά τον υπολογισμό του group delay για κάθε συχνότητα ενός πειράματος οι σχετικοί κόμβοι φαίνονται στην παραπάνω εικόνα. Επειδή για τον υπολογισμό του group delay χρειάζονται οι τιμές δύο διαδοχικών συχνοτήτων χρησιμοποιούμε έναν buffer όπου κρατιέται η μέση τιμή της τάσης και η αντίστοιχή της συχνότητα. Ο switch κόμβος buffer empty? ελέγχει αν ο buffer είναι άδειος, δηλαδή αν πρόκειται για την πρώτη συχνότητα του πειράματος. Στην περίπτωση αυτή απλά γίνεται αποθήκευση των τιμών στον buffer. Στον function κόμβο Group Delay calc. γίνεται αρχικά ο υπολογισμός του group delay. Επίσης, αποθηκεύονται στον buffer οι τιμές της συχνότητας και της μέσης τιμής της τάσης που μόλις ήρθαν στο flow. Η τιμή του group delay που υπολογίστηκε και η συχνότητα στην οποία αντιστοιχεί αφενός τοποθετούνται σε έναν δεύτερο buffer που συλλέγει ανά 5 τα ζευγάρια group delay-συχνότητας για την αποστολή τους στο website για τη δημιουργία των διαγραμμάτων σε πραγματικό χρόνο και αφετέρου τοποθετούνται στο payload ενός μηνύματος που στέλνεται για την εγγραφή τους στη βάση δεδομένων. Στην έξοδο του κόμβου function έχουμε μια τριπλή διακλάδωση. Η πρώτη διαδρομή ελέγχει εάν έχουν συμπληρωθεί 5 ζευγάρια group delay-συχνότητας για την αποστολή τους στο website. Αν αυτό ισχύει, το msg.payload παίρνει την τιμή του buffer και αφού διαμορφωθεί κατάλληλα στέλνεται στο websocket και ο buffer αδειάζει για να λάβει τα επόμενα ζευγάρια. Η δεύτερη διαδρομή αφορά την αποθήκευση του κάθε ζεύγους στη βάση δεδομένων. Τέλος, στην τρίτη διαδρομή ο κόμβος switch measurements_ended ? ελέγχει εάν έχουν ολοκληρωθεί οι μετρήσεις και μόνο τότε το flow προχωράει αρχικά στον κόμβο change για το άδειασμα του buffer για το επόμενο πείραμα και στη συνέχεια στον κόμβο link για την ενημέρωση του FSM και την αποστολή των τελευταίων ζευγών στο website.

Κλείνοντας την ενότητα αυτή θα περιγράψουμε τις υπόλοιπες καταστάσεις από τις οποίες περνάει το FSM, όπως αυτές φαίνονται στις εικόνες 44 και 45 της επόμενης σελίδας. Μόλις ολοκληρωθεί η διαδικασία των μετρήσεων και απενεργοποιηθούν οι διάφοροι διακόπτες, στέλνεται μήνυμα στον κόμβο του FSM. Στη συνέχεια ακολουθεί ενημέρωση του website για το τέλος της διαδικασίας των μετρήσεων, ενημέρωση της βάσης δεδομένων για την επιτυχή εκτέλεση του πειράματος και τέλος μέσω ενός κόμβου change το πεδίο msg.complete παίρνει την τιμή true. Το μήνυμα αυτό μέσω του κόμβου link στέλνεται στην πρώτη από τις δύο σειρές κόμβων που φαίνονται στην εικόνα 45. Στον κόμβο join καταλήγουν μηνύματα από δύο σημεία. Από αυτό που μόλις αναφέραμε παραπάνω και ένα μήνυμα μόλις ολοκληρωθεί ο υπολογισμός των μέσων τιμών τάσης για όλες τις συχνότητες. Επειδή δεν θέλουμε να ενημερωθεί το FSM για το τέλος των υπολογισμών των μέσων τιμών πριν να έχει ενημερωθεί για το τέλος των μετρήσεων, ο κόμβος join δεν επιτρέπει στο flow να συνεχίσει παρά μόνο όταν φτάσει σε αυτόν μήνυμα με το πεδίο msg.complete να έχει την τιμή true. Αυτό όπως είδαμε συμβαίνει μόνο αφού ενημερωθεί το FSM για το τέλος της διαδικασίας των μετρήσεων. Στη συνέχεια ενημερώνεται το FSM για την ολοκλήρωση του υπολογισμού των μέσων τιμών τάσης και στέλνονται πάλι τα αντίστοιχα μηνύματα ενημέρωσης στο website αλλά και με το πεδίο msg.complete με τιμή true στη δεύτερη σειρά κόμβων της εικόνας 45. Η λειτουργία αυτού του κόμβου join αφορά ό,τι ακριβώς και του προηγούμενου αλλά για τη σωστή σειρά ολοκλήρωσης των flow της μέσης τιμής και του group delay. Μόλις λοιπόν ενημερωθεί το FSM και για την ολοκλήρωση του υπολογισμού των group delays, στέλνεται στο website το αντίστοιχο μήνυμα και γίνονται κάποιες απαραίτητες ενημερώσεις στη βάση δεδομένων. Επιπλέον λαμβάνονται και αποθηκεύονται στο αρχείο results.csv τα αποτελέσματα τόσο για τις μέσες τιμές τάσης όσο και για τα group delays για να είναι διαθέσιμα για κατέβασμα μέσω του website.



Εικόνα 44: Οι κόμβοι που αφορούν τις υπόλοιπες καταστάσεις του FSM



Εικόνα 45: Οι κόμβοι που αφορούν το σημείο αναμονής των flows μέχρι να ολοκληρωθεί η διαδικασία των μετρήσεων

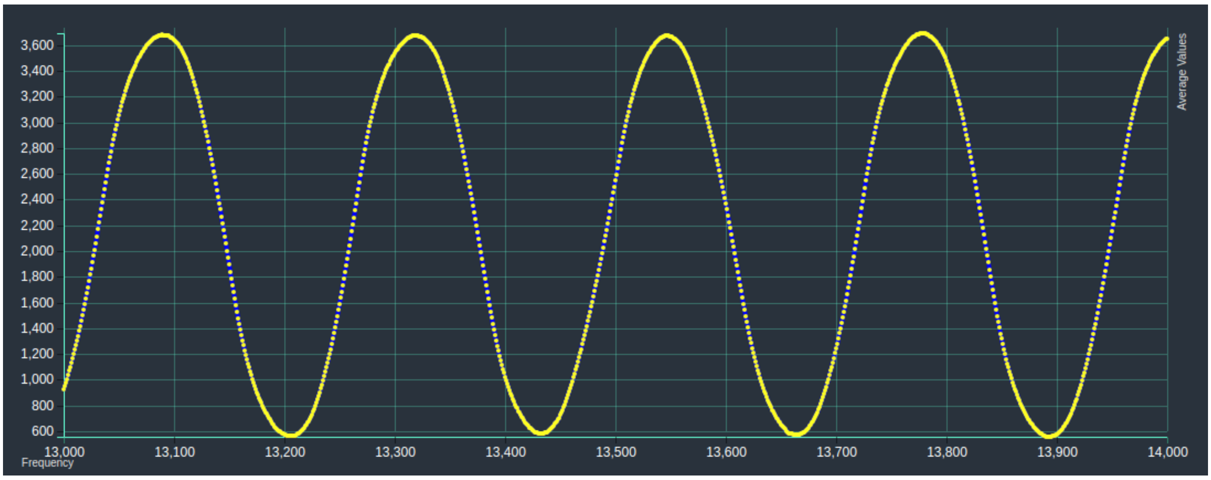
Τέλος, να πούμε ότι το FSM διαθέτει τέσσερις ακόμα καταστάσεις που αφορούν την ανεπιτυχή ολοκλήρωση του πειράματος σε περίπτωση που το synthesizer επιστρέψει reject σε κάποια συχνότητα, το ενδεχόμενο να πατήσει ο χρήστης “Stop” στο website, το ενδεχόμενο να πατήσει ο χρήστης “Restart” στο website για επανέναρξη ενός πειράματος και το ενδεχόμενο να ζητηθεί η τρέχουσα κατάσταση του FSM πατώντας το κουμπί Refresh status/LEDS.

Κεφάλαιο 5ο
Τελικά αποτελέσματα & μελλοντικές επεκτάσεις

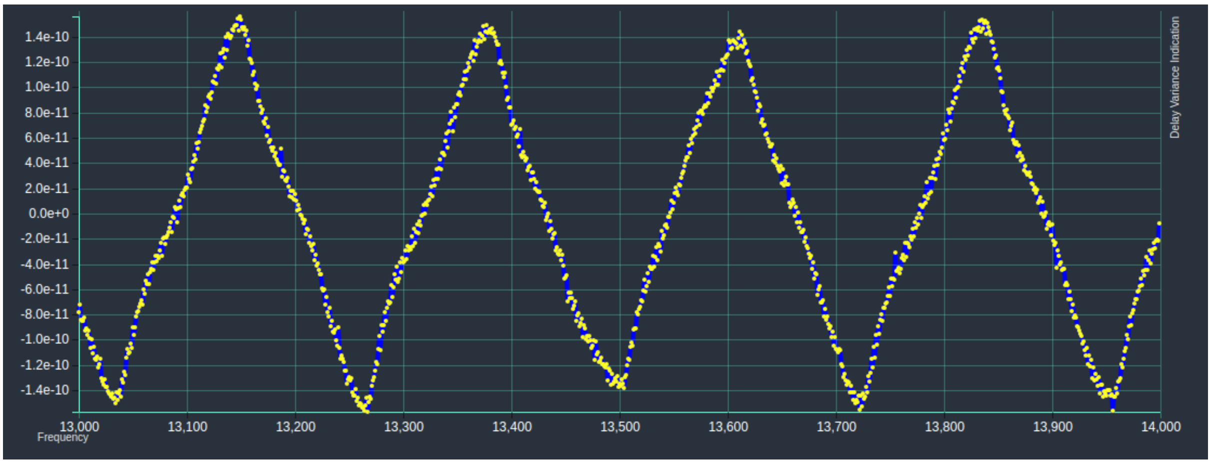
Στο κεφάλαιο αυτό αρχικά θα παρουσιάσουμε τα αποτελέσματα των μετρήσεων και των υπολογισμών για όλο το εύρος συχνοτήτων. Στη δεύτερη ενότητα του κεφαλαίου θα αναφέρουμε μερικές από τις ενδεχόμενες μελλοντικές επεκτάσεις του συστήματος.

5.1 Τελικά αποτελέσματα

Παρακάτω βλέπουμε γραφήματα για την μέση τιμής της τάσης εξόδου του μετατροπέα ADC και την καθυστέρηση ομάδος (group delay) για τις τάσεις αυτές στις αντίστοιχες συχνότητες για όλο το εύρος συχνοτήτων των πειραμάτων. Όπως βλέπουμε οι μέγιστες τιμές για την τάση είναι όπως αναμενόταν λίγο πάνω από τα 3.6V και οι ελάχιστες λίγο κάτω από τα 0.6V, ενώ η τάση ακολουθεί ημιτονοειδή μορφή. Η γραφική παράσταση του group delay έχει τριγωνική περιοδική μορφή και είναι σχετικά συμμετρική γύρω από το 0.



Εικόνα 46: Απεικόνιση της μέσης τιμής της τάσης της εξόδου του μετατροπέα ADC σε όλο το πιθανό εύρος συχνοτήτων του πειράματος.



Εικόνα 47: Απεικόνιση του group delay της τάσης της εξόδου του μετατροπέα ADC σε όλο το πιθανό εύρος συχνοτήτων του πειράματος

5.2 Μελλοντικές επεκτάσεις

Παρακάτω αναφέρουμε τρεις πιθανές μελλοντικές επεκτάσεις του συστήματος που υλοποιήθηκε στα πλαίσια αυτής της διπλωματικής εργασίας.

Στην παρούσα φάση της υλοποίησης είναι διαθέσιμα για απεικόνιση και κατέβασμα μόνο τα δεδομένα του πιο πρόσφατα εκτελεσμένου πειράματος. Δεν γίνεται κάποια αποθήκευση των δεδομένων ανά πείραμα έτσι ώστε αυτά να είναι διαθέσιμα και μελλοντικά για αναφορά στους χρήστες του συστήματος. Θα μπορούσε λοιπόν να υλοποιηθεί μηχανισμός που να αποθηκεύει τα δεδομένα κάθε πειράματος σε μια βάση δεδομένων έτσι ώστε αυτά να είναι διαθέσιμα είτε για απεικόνιση σε γραφήματα στο website είτε για κατέβασμα σε μορφή csv και περαιτέρω επεξεργασία.

Σε περίπτωση που αποφασιστεί ότι η βαθμονόμηση του δορυφόρου Sentinel-3 χρειάζεται να γίνεται σε περισσότερα από 1 σημεία ανά την υφήλιο, θα ήταν πιο αποδοτικό η επεξεργασία των δεδομένων που λαμβάνονται στα διάφορα σημεία να γίνεται κεντρικοποιημένα και όχι σε κάθε επιμέρους συσκευή. Έτσι, θα μπορούσε να αναπτυχθεί μια κεντρική πλατφόρμα cloud η οποία να λαμβάνει τις μετρήσεις από κάθε επιμέρους συσκευή βαθμονόμησης και να εκτελούνται εκεί οι διάφοροι μετέπειτα υπολογισμοί αλλά και η απεικόνιση των αποτελεσμάτων. Για λόγους οικονομίας ενδεχομένως να χρειάζεται κάποια συμπίεση των δεδομένων πριν την αποστολή τους στην κεντρική υποδομή επεξεργασίας, κάτι το οποίο είναι εφικτό στο Node-RED. Η συμπίεση και η αποστολή των μετρήσεων σε μια κεντρική πλατφόρμα cloud με μεγαλύτερες υπολογιστικές δυνατότητες απ' ότι για παράδειγμα το Beaglebone μπορεί να χρειαστεί και αν μελλοντικά αποφασιστεί να γίνονται υπολογισμοί περισσότερο απαιτητικοί σε υπολογιστικούς πόρους τους οποίους δεν θα μπορεί να παρέχει το Beaglebone.

Όπως είδαμε στην ενότητα 5.1, η γραφική παράσταση group delay είναι τριγωνικής μορφής και αυτό οφείλεται στο γεγονός ότι για τον υπολογισμό του δεν λαμβάνεται η φάση της τάσης εξόδου αλλά το μέτρο της τιμής της, καθώς αποδείχθηκε πειρατικά ότι αυτό δεν επηρεάζει σημαντικά την ακρίβεια των υπολογισμών. Ωστόσο σε περίπτωση που θα θέλαμε μεγαλύτερη ακρίβεια θα μπορούσαμε να εκτελέσουμε κάποιους περαιτέρω υπολογισμούς στο Node-RED πριν τον υπολογισμό του group delay, έτσι ώστε το ημίτονο που λαμβάνουμε στην γραφική παράσταση της τάσης εξόδου του μετατροπέα ADC να βρίσκεται στο εύρος [-1,1]. Έτσι, στη συνέχεια εφαρμόζοντας τη συνάρτηση arcsin στις τιμές που θα έχουμε θα μπορούμε να πάρουμε τη φάση της τάσης και να χρησιμοποιήσουμε αυτή στους υπολογισμούς για το group delay.

ΚΕΦΑΛΑΙΟ 6º
Βιβλιογραφία

[1] “The NIST Definition of Cloud Computing“
https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-145.pdf
[2] “Cloud Computing Bible”, Barrie Sosinsky
https://arpitapatel.files.wordpress.com/2014/10/cloud-computing-bible1.pdf
[3] https://www.salesforce.com/uk/blog/2015/11/why-move-to-the-cloud-10-benefits-of-cloud-computing.html
[4] http://internetofthingsagenda.techtarget.com/definition/fog-computing-fogging
[5] “The NIST Definition of Fog Computing”
https://csrc.nist.gov/CSRC/media//Publications/sp/800-191/draft/documents/sp800-191-draft.pdf
[6] “Fog Computing: Will it be the Future of Cloud Computing?”
https://www.researchgate.net/publication/266477246_Fog_Computing_Will_it_be_the_Future_of_Cloud_Computing
[7] http://searchdatacenter.techtarget.com/definition/edge-computing
[8] https://www.networkworld.com/article/3224893/internet-of-things/what-is-edge-computing-and-how-it-s-changing-the-network.html
[9] https://www.networkworld.com/article/3234708/internet-of-things/why-edge-computing-is-critical-for-the-iot.html
[10] “Edge of Things: The Big Picture on the Integration of Edge, IoT and the Cloud in a Distributed Computing Environment“,
http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8166730
[11] http://www.synergic.gr/web/synergic-software/-/internet-of-things-iot-
[12] https://internetofthingswiki.com/iot-applications-examples/541/#Smart-Home
[13] https://www.echelon.com/assets/bltd6560c36a2c5f39f/Lighting-City-of-Oslo-Street-lighting-case-study.pdf
[14] https://www.patecco.com/blog/what-are-the-benefits-of-integration-are-between-cloud-computing-and-internet-of-things
[15] http://www.eweek.com/big-data-and-analytics/10-benefits-of-analyzing-data-at-the-edge-in-an-iot-environment
[16] https://en.wikipedia.org/wiki/Dataflow_programming
[17] https://www.kaaproject.org/what-is-iot/
[18] https://www.slideshare.net/KaiWaehner/iot-open-source-integration-comparison-kura-nodered-flogo-apache-nifi-streamsets/7
[19] http://eclipse.github.io/kura/intro/intro.html
[20] https://github.com/apache/camel/blob/master/README.md
[21] http://eclipse.github.io/kura/camel/kura-camel.html
[22] https://www.kaaproject.org/overview/
[23] https://kaaproject.github.io/kaa/docs/v0.10.0/Programming-guide/Key-platform-features/

[24] https://www.kaaproject.org/platform/
[25] https://www.tibco.com/blog/2016/12/09/open-source-project-flogo-overview-architecture-and-live-demo/
[26] http://www.flogo.io/
[27] https://tibcosoftware.github.io/flogo/glossary/
[28] “Toward a Distributed Data Flow Platform for the Web of Things”, paper των Michael Blackstock & Rodger Lea
[29] https://nodered.org/
[30] https://medium.com/iotforall/how-to-choose-your-iot-platform-should-you-go-open-source-23148a0809f3
[31] https://sentinel.esa.int/web/sentinel/missions/sentinel-3/overview
[32] https://en.wikipedia.org/wiki/Sentinel-3
[33] http://www.altimetry.info/glossary_/altimetry/
[34] https://sentinel.esa.int/web/sentinel/technical-guides/sentinel-3-altimetry/instrument/sral
[35] https://sentinel.esa.int/web/sentinel/user-guides/sentinel-3-altimetry/overview
[36] “New Group Delay and Phase Measurement Method for Long Distance Transmission”, Thilo Bednorz and Jochen Wolle, Rohde & Schwarz
[37] BITE Review PDF
[38] BITE Calibration PDF
[39] SLSM5 Interface Definition

Παράρτημα

Κώδικας flows του Node-RED

Network & FSM

[{"id":"13b95697.ee62a1","type":"tab","label":"Network &
FSM"},{"id":"df615ba6.96c8c","type":"websocket
out","z":"13b95697.ee62a1","name":"","server":"26087f6b.2a439","client":""
,"x":180,"y":160,"wires":[]},{"id":"8094e49f.d67768","type":"link
in","z":"13b95697.ee62a1","name":"SocketOut","links":["a78b9b8f.743b6","f8
5c049b.560698","5303b012.8a99","f0d21b07.6ade68","e5d62256.ad4bf","566b1f3
4.5e6bd","cc5c2c5a.84fe8","36eeabb8.c35884","c5fe731.c7e209","1cfd650a.77c
48b","8d04e3fe.c6b01","6cec57aa.c0c478","e4f09a24.47f2e8","c393277e.4f3e68
","4611453a.353b8c","8364aa01.a78028","2fb384d3.e03574","3fcbbafa.3f5186",
"e99972c2.78307","9c00a931.6107b","ae476b5c.420738","4a45ebbb.166824","edd
44d27.840f98","c1a37ee.ae9718","5fac3116.b3eff8","bb295946.dbfc48","5d3667
f8.2fb1f8","10628a30.9260de","fa2021f.ff77f6","2b50c61c.21892a","7340c414.
691f44","eb827457.ce0d3","cc047ea.42a8b8","f7fcf1bd.3bc268","14982d04.6fcb
83","a529c568.931928","7abfcfb7.06b8a8","8a972673.fdfe78","60292eac.b0893"
,"303d1197.35194e","8310a1c8.e1d01","f92fd03f.da141","a4cd34b5.781d1","783
f92aa.17b0dc","c07bc97a.4bec9","31d09885.1c63a8","3029692d.830e06","3c00c0
c1.1c221","42f55fbd.b2cd4","9c7eecc9.c31698","a4966b8f.b183"],"x":55,"y":1
60,"wires":[["df615ba6.96c8c"]]},{"id":"cc47275a.b08548","type":"websocket
in","z":"13b95697.ee62a1","name":"","server":"26087f6b.2a439","client":"",
"x":120,"y":280,"wires":[["7cd44d61.f84bec"]]},{"id":"7cd44d61.f84bec","ty
pe":"json","z":"13b95697.ee62a1","name":"JSON","x":270,"y":280,"wires":[["
a9ec38f3.6f3f18"]]},{"id":"e246078f.1690a","type":"link
out","z":"13b95697.ee62a1","name":"SocketIn_Start","links":["4ca04501.5460
9c","f1563bd6.975f78","b9663a56.0543a"],"x":895,"y":240,"wires":[]},{"id":
"a9ec38f3.6f3f18","type":"switch","z":"13b95697.ee62a1","name":"mode ==
START / RESTART /
STOP","property":"payload.mode","propertyType":"msg","rules":[{"t":"eq","v
":"START","vt":"str"},{"t":"eq","v":"RESTART","vt":"str"},{"t":"else"}],"c
heckall":"true","outputs":3,"x":490,"y":280,"wires":[["1b20d688.c33389"],[
"d205cdec.8f204"],["4f4a8712.c18108"]]},{"id":"36e42c89.fd5a6c","type":"li
nk
out","z":"13b95697.ee62a1","name":"SocketIn_Stop","links":["3ca99234.dce32
e"],"x":835,"y":320,"wires":[]},{"id":"f5c6113c.40c77","type":"comment","z
":"13b95697.ee62a1","name":"Outgoing Web
Socket","info":"","x":140,"y":100,"wires":[]},{"id":"ac3c3aed.f04218","typ
e":"comment","z":"13b95697.ee62a1","name":"Incoming Web
Socket","info":"","x":140,"y":220,"wires":[]},{"id":"fce49863.8dca98","typ
e":"http
in","z":"13b95697.ee62a1","name":"","url":"/status","method":"get","swagge
rDoc":"","x":110,"y":440,"wires":[["57965d3d.84182c","87d99253.cfd"]]},{"i
d":"57965d3d.84182c","type":"debug","z":"13b95697.ee62a1","name":"Status",
"active":false,"console":"false","complete":"payload","x":250,"y":420,"wir
es":[]},{"id":"87d99253.cfd","type":"json","z":"13b95697.ee62a1","name":"J
SON","x":370,"y":440,"wires":[["f4f21f55.144188"]]},{"id":"926e6d29.52f6f"
,"type":"comment","z":"13b95697.ee62a1","name":"HTTP request (incoming)
for the
STATUS","info":"","x":200,"y":380,"wires":[]},{"id":"c252e895.d51c88","typ
e":"switch","z":"13b95697.ee62a1","name":"state >
1","property":"payload","propertyType":"msg","rules":[{"t":"gt","v":"1","v
t":"str"},{"t":"else"}],"checkall":"true","outputs":2,"x":740,"y":440,"wir
es":[["44e38949.46a4e"],["83ab1a9d.56237"]]},{"id":"a33b55ab.6a9ac8","type
":"comment","z":"13b95697.ee62a1","name":"HTTP request (incoming) for the
LED's","info":"","x":190,"y":520,"wires":[]},{"id":"1c17b066.c30f78","type
":"http

in","z":"13b95697.ee62a1","name":"","url":"/leds","method":"get","swaggerD
oc":"","x":100,"y":660,"wires":[["120b266c.121e2a","70ab35c3.210a44","8130
4713.768fa","3c1eff2a.b851d8"]]},{"id":"ff250e28.1337d","type":"function",
"z":"13b95697.ee62a1","name":"Detect GPIO pins","func":"context.data =
context.data || {};\n\nswitch (msg.topic) {\n   case \"5V\":\n
context.data.temp1 = msg.payload;\n   context.data.original = msg;\n
msg = null;\n   break;\n   \n   case \"12V\":\n
context.data.temp2 = msg.payload;\n   msg = null;\n   break;\n
\n   case \"LD\":\n   context.data.temp3 = msg.payload ? \"1\" :
\"0\";\n   msg = null;\n   break;\n   \n   case \"OC\":\n
context.data.temp4 = msg.payload;\n   msg = null;\n   break;\n
\n   default:\n   msg = null;\n
\tbreak;\n\n}\n\n\nif(context.data.temp1 != null && context.data.temp2 !=
null && context.data.temp3 != null && context.data.temp4 != null)
{\n\tmsg2 = context.data.original;\n   msg2.payload =
'{\"led1\":1,\"led2\":' + context.data.temp1 + ',\"led3\":' +
context.data.temp2 + ',\"led4\":' + context.data.temp3 + ',\"led5\":' +
context.data.temp4 + '}';\n   context.data=null;\n\treturn msg2;\n}
","outputs":1,"noerr":0,"x":810,"y":660,"wires":[["afd6c58.ec99938"]]},{"i
d":"f6482e6b.eac288","type":"http
response","z":"13b95697.ee62a1","name":"","x":1210,"y":660,"wires":[]},{"i
d":"afd6c58.ec99938","type":"change","z":"13b95697.ee62a1","name":"Set
Response
Headers","rules":[{"t":"set","p":"headers","pt":"msg","to":"{}","tot":"jso
n"},{"t":"set","p":"headers.content-
type","pt":"msg","to":"application/json","tot":"str"},{"t":"set","p":"head
ers.Access-Control-Allow-
Origin","pt":"msg","to":"*","tot":"str"}],"action":"","property":"","from"
:"","to":"","reg":false,"x":1030,"y":660,"wires":[["f6482e6b.eac288"]]},{"
id":"83ab1a9d.56237","type":"function","z":"13b95697.ee62a1","name":"msg =
BUSY   .","func":"msg.payload = '{\"msg\":\"BUSY\",\"state\":' +
msg.payload + '}';\nreturn
msg;","outputs":1,"noerr":0,"x":900,"y":460,"wires":[["efa0a3fc.794d3"]]},
{"id":"44e38949.46a4e","type":"function","z":"13b95697.ee62a1","name":"msg
= READY","func":"msg.payload = '{\"msg\":\"READY\",\"state\":' +
msg.payload + '}';\nreturn
msg;","outputs":1,"noerr":0,"x":900,"y":420,"wires":[["efa0a3fc.794d3"]]},
{"id":"efa0a3fc.794d3","type":"change","z":"13b95697.ee62a1","name":"Set
Response
Headers","rules":[{"t":"set","p":"headers","pt":"msg","to":"{}","tot":"jso
n"},{"t":"set","p":"headers.content-
type","pt":"msg","to":"application/json","tot":"str"},{"t":"set","p":"head
ers.Access-Control-Allow-
Origin","pt":"msg","to":"*","tot":"str"}],"action":"","property":"","from"
:"","to":"","reg":false,"x":1130,"y":440,"wires":[["81020b9.827fd78"]]},{"
id":"81020b9.827fd78","type":"http
response","z":"13b95697.ee62a1","name":"","x":1310,"y":440,"wires":[]},{"i
d":"2574ca4.c4a3736","type":"comment","z":"13b95697.ee62a1","name":"Networ
k Operations - FSM
Implementation","info":"","x":480,"y":40,"wires":[]},{"id":"8343244b.d49d7
8","type":"comment","z":"13b95697.ee62a1","name":"=================
Finite State Machine - FSM
=====================","info":"","x":330,"y":820,"wires":[]},{"id":"1b20d
688.c33389","type":"change","z":"13b95697.ee62a1","name":"Input <=>
Payload","rules":[{"t":"set","p":"inputs","pt":"msg","to":"payload","tot":
"msg"},{"t":"set","p":"topic","pt":"msg","to":"2","tot":"str"}],"action":"
","property":"","from":"","to":"","reg":false,"x":770,"y":240,"wires":[["e
246078f.1690a"]]},{"id":"313ee27e.991876","type":"link
out","z":"13b95697.ee62a1","name":"","links":["1c658584.477f1a","4ca04501.
54609c"],"x":835,"y":280,"wires":[]},{"id":"70ab35c3.210a44","type":"exec"
,"z":"13b95697.ee62a1","command":"cat

/sys/class/gpio/gpio47/value","addpay":true,"append":"","useSpawn":"","timer":"","name":"P8_15.value => 12V_EN","x":370,"y":640,"wires":[["3f519855.64ce2"],[],[]]},{"id":"120b266c.121e2a","type":"exec","z":"13b95697.ee62a1","command":"cat /sys/class/gpio/gpio20/value","addpay":true,"append":"","useSpawn":"","timer":"","name":"P9_41.value => 5V_EN","x":370,"y":580,"wires":[["6fa50015.ed50d"],[],[]]},{"id":"81304713.768fa","type":"change","z":"13b95697.ee62a1","name":"Get => (global.lock)","rules":[{"t":"set","p":"payload","pt":"msg","to":"lock","tot":"global"}],"action":"","property":"","from":"","to":"","reg":false,"x":360,"y":700,"wires":[["43ebfc1e.f3a664"]]},{"id":"6fa50015.ed50d","type":"change","z":"13b95697.ee62a1","name":"topic = 5V .","rules":[{"t":"set","p":"topic","pt":"msg","to":"5V","tot":"str"}],"action":"","property":"","from":"","to":"","reg":false,"x":570,"y":580,"wires":[["ff250e28.1337d"]]},{"id":"3f519855.64ce2","type":"change","z":"13b95697.ee62a1","name":"topic = 12V","rules":[{"t":"set","p":"topic","pt":"msg","to":"12V","tot":"str"}],"action":"","property":"","from":"","to":"","reg":false,"x":570,"y":640,"wires":[["ff250e28.1337d"]]},{"id":"43ebfc1e.f3a664","type":"change","z":"13b95697.ee62a1","name":"topic = LD","rules":[{"t":"set","p":"topic","pt":"msg","to":"LD","tot":"str"}],"action":"","property":"","from":"","to":"","reg":false,"x":570,"y":700,"wires":[["ff250e28.1337d"]]},{"id":"c0de000b.ff07d8","type":"FSM v1.0","z":"13b95697.ee62a1","name":"","x":160,"y":1520,"wires":[["3990eda.2026312"],["fbf45333.277f8"],["b8947828.80ff4"],["35e19479.8cf0e4"],["d380d1bb.a3bf68"],["e8039a84.acba7"],["b183ae72.edf348"],["3797599f.10bb06"],["70625208.dae7ac"]]},{"id":"53e25a6c.37837c","type":"comment","z":"13b95697.ee62a1","name":"After Power Up - Initialization","info":"","x":480,"y":1220,"wires":[]},{"id":"98a1aced.15211","type":"comment","z":"13b95697.ee62a1","name":"Start Measurement","info":"","x":450,"y":1300,"wires":[]},{"id":"5bbf7910.8e5ef8","type":"comment","z":"13b95697.ee62a1","name":"Measurements finished","info":"","x":460,"y":1380,"wires":[]},{"id":"388b7df4.4a20c2","type":"comment","z":"13b95697.ee62a1","name":"Unsuccessful exper.","info":"","x":450,"y":1460,"wires":[]},{"id":"43bec6b6.cbbee","type":"comment","z":"13b95697.ee62a1","name":"Stop experiment","info":"","x":440,"y":1540,"wires":[]},{"id":"2a28312.27177ce","type":"comment","z":"13b95697.ee62a1","name":"Averaging is over","info":"","x":440,"y":1620,"wires":[]},{"id":"5b805b5e.0b2424","type":"comment","z":"13b95697.ee62a1","name":"GD is over","info":"","x":420,"y":1700,"wires":[]},{"id":"cf705890.7da2a8","type":"comment","z":"13b95697.ee62a1","name":"Restart has been issued","info":"","x":470,"y":1780,"wires":[]},{"id":"7e1ab5b.095e04c","type":"comment","z":"13b95697.ee62a1","name":"Get current state","info":"","x":440,"y":1860,"wires":[]},{"id":"3990eda.2026312","type":"change","z":"13b95697.ee62a1","name":"lock = false , buffer = {0,0}, acc_buff = {null}","rules":[{"t":"set","p":"lock","pt":"global","to":"false","tot":"bool"},{"t":"set","p":"buffer","pt":"global","to":"{\"freq\":0,\"value\":0}","tot":"json"},{"t":"set","p":"acc_buff","pt":"global","to":"{\"array\":[]}","tot":"json"},{"t":"set","p":"acc_buff_2","pt":"global","to":"{\"array\":[]}","tot":"json"}],"action":"","property":"","from":"","to":"","reg":false,"x":530,"y":1260,"wires":[["d8cc2e6f.e54be"]]},{"id":"4ca04501.54609c","type":"link in","z":"13b95697.ee62a1","name":"MASTER_FSM_INPUT","links":["af1cf5d3.d8f258","313ee27e.991876","e246078f.1690a","4197f962.eaeef8","7bafe213.6e6f5c","c7dec9a1.0279f","85960953.73879"],"x":55,"y":1520,"wires":[["c0de000b.ff07d8"]]},{"id":"d8cc2e6f.e54be","type":"exec","z":"13b95697.ee62a1","command":"config-pin p9.24 uart && config-pin p9.26 uart","addpay":false,"append":"","useSpawn":"","timer":"","name":"","x":900,"y":1260,"wires":[["8df8f649.ea615"],[],[]]},{"id":"cc047ea.42a8b8","typ

e":"link
out","z":"13b95697.ee62a1","name":"End_state2","links":["8094e49f.d67768"]
,"x":1315,"y":1260,"wires":[]},{"id":"fbf45333.277f8","type":"change","z":
"13b95697.ee62a1","name":"Message to
client","rules":[{"t":"set","p":"payload","pt":"msg","to":"{\"state\":3}",
"tot":"json"},{"t":"set","p":"synth_rejected","pt":"global","to":"0","tot"
:"num"}],"action":"","property":"","from":"","to":"","reg":false,"x":450,"
y":1340,"wires":[["f7fcf1bd.3bc268"]]},{"id":"f7fcf1bd.3bc268","type":"lin
k
out","z":"13b95697.ee62a1","name":"End_state3","links":["42d78505.06fccc",
"8094e49f.d67768"],"x":575,"y":1340,"wires":[]},{"id":"8df8f649.ea615","ty
pe":"change","z":"13b95697.ee62a1","name":"Message to
client","rules":[{"t":"set","p":"payload","pt":"msg","to":"{\"state\":2}",
"tot":"json"}],"action":"","property":"","from":"","to":"","reg":false,"x"
:1190,"y":1260,"wires":[["cc047ea.42a8b8"]]},{"id":"b8947828.80ff4","type"
:"change","z":"13b95697.ee62a1","name":"Message to
client","rules":[{"t":"set","p":"payload","pt":"msg","to":"{\"state\":4}",
"tot":"str"}],"action":"","property":"","from":"","to":"","reg":false,"x":
450,"y":1420,"wires":[["14982d04.6fcb83","5d8dd4a6.b649fc"]]},{"id":"14982
d04.6fcb83","type":"link
out","z":"13b95697.ee62a1","name":"End_state4","links":["8094e49f.d67768"]
,"x":595,"y":1400,"wires":[]},{"id":"5d8dd4a6.b649fc","type":"function","z
":"13b95697.ee62a1","name":"Successful Exper.","func":"msg.topic=\"UPDATE
ExperimentResult SET id=1, message='The last experiment was successful.'
WHERE id=1\";\n\n\nreturn
msg;","outputs":1,"noerr":0,"x":710,"y":1420,"wires":[["c52d0a18.aa2d9"]]}
,{"id":"c52d0a18.aa2d9","type":"mysql","z":"13b95697.ee62a1","mydb":"d5193
bf.585bac8","name":"","x":880,"y":1420,"wires":[["a805fe24.75e2b8","bcccad
1b.6d7938"]]},{"id":"80f57be4.87928","type":"link
out","z":"13b95697.ee62a1","name":"End_state4","links":["f8308a02.b8389","
fab854d4.15ba18"],"x":1355,"y":1420,"wires":[]},{"id":"a805fe24.75e2b8","t
ype":"debug","z":"13b95697.ee62a1","name":"Measure_END","active":false,"co
nsole":"false","complete":"true","x":1040,"y":1400,"wires":[]},{"id":"35e1
9479.8cf0e4","type":"function","z":"13b95697.ee62a1","name":"Unsuccessful
Exp","func":"var errfreq = msg.fstart;\nmsg.topic=\"UPDATE
ExperimentResult SET id=1, message='The last experiment was unsuccessful.
The synth rejected at \"+errfreq+\" MHz frequency' WHERE id=1\";\n\nreturn
msg;","outputs":1,"noerr":0,"x":450,"y":1500,"wires":[["d73ff741.e3648"]]}
,{"id":"d73ff741.e3648","type":"mysql","z":"13b95697.ee62a1","mydb":"d5193
bf.585bac8","name":"","x":620,"y":1500,"wires":[["ef7a28e2.09f3f"]]},{"id"
:"c7ffa23b.0c1378","type":"debug","z":"13b95697.ee62a1","name":"Failure_EN
D","active":false,"console":"false","complete":"true","x":1110,"y":1580,"w
ires":[]},{"id":"d380d1bb.a3bf68","type":"link
out","z":"13b95697.ee62a1","name":"","links":["3ca99234.dce32e"],"x":375,"
y":1580,"wires":[]},{"id":"e8039a84.acba7","type":"change","z":"13b95697.e
e62a1","name":"Message to
client","rules":[{"t":"set","p":"payload","pt":"msg","to":"{\"state\":5}",
"tot":"str"},{"t":"set","p":"topic","pt":"msg","to":"5end","tot":"str"},{"
t":"set","p":"complete","pt":"msg","to":"true","tot":"bool"}],"action":"",
"property":"","from":"","to":"","reg":false,"x":450,"y":1660,"wires":[["7a
bfcfb7.06b8a8","997a8b5a.9e7fe8"]]},{"id":"7abfcfb7.06b8a8","type":"link
out","z":"13b95697.ee62a1","name":"End_state5","links":["8094e49f.d67768",
"b5c86e36.1c3a5"],"x":755,"y":1660,"wires":[]},{"id":"997a8b5a.9e7fe8","ty
pe":"debug","z":"13b95697.ee62a1","name":"Average
END","active":false,"console":"false","complete":"payload","x":640,"y":164
0,"wires":[]},{"id":"b183ae72.edf348","type":"change","z":"13b95697.ee62a1
","name":"Message to
client","rules":[{"t":"set","p":"payload","pt":"msg","to":"{\"state\":7}",
"tot":"str"}],"action":"","property":"","from":"","to":"","reg":false,"x":
450,"y":1740,"wires":[["2bb7ddcb.2719ba","60292eac.b0893"]]},{"id":"e83865
2c.650f5","type":"debug","z":"13b95697.ee62a1","name":"GD

END","active":false,"console":"false","complete":"payload","x":1020,"y":18
20,"wires":[]},{"id":"8a972673.fdfe78","type":"link
out","z":"13b95697.ee62a1","name":"End_state7","links":["8094e49f.d67768"]
,"x":1255,"y":1860,"wires":[]},{"id":"2bb7ddcb.2719ba","type":"function","
z":"13b95697.ee62a1","name":"DELETE id=2,
Average","func":"\nmsg.topic=\"DELETE FROM AverageBBB WHERE
id=2\";\n\nreturn
msg;","outputs":1,"noerr":0,"x":750,"y":1740,"wires":[["b00ecef1.6ce038"]]
},{"id":"c9df3254.3d5528","type":"function","z":"13b95697.ee62a1","name":"
UPDATE id=2, Average","func":"\nmsg.topic=\"UPDATE AverageBBB SET id=2
WHERE id=1\";\n\n\nreturn
msg;","outputs":1,"noerr":0,"x":750,"y":1780,"wires":[["ad4a0480.5a0bf"]]}
,{"id":"7327e9e2.4942f8","type":"csv","z":"13b95697.ee62a1","name":"","sep
":",","hdrin":"","hdrout":true,"multi":"one","ret":"\\n","temp":"frequency
New, average,
GroupDelay","x":1010,"y":1880,"wires":[["7370381b.eb2178","c3ba047c.3663e"
]]},{"id":"7370381b.eb2178","type":"file","z":"13b95697.ee62a1","name":"",
"filename":"/var/www/html/BeagleSite_v3/results.csv","appendNewline":true,
"createDir":false,"overwriteFile":"true","x":1260,"y":1900,"wires":[]},{"i
d":"f3026450.daf2a","type":"function","z":"13b95697.ee62a1","name":"Get
Results","func":"msg.topic = \"SELECT frequencyNew, average, GroupDelay
FROM AverageBBB LEFT JOIN GroupDelayBBB ON AverageBBB.frequencyNew =
GroupDelayBBB.frequencyGD\";\nreturn
msg;","outputs":1,"noerr":0,"x":710,"y":1820,"wires":[["6ff2e404.4a8be4"]]
},{"id":"6ff2e404.4a8be4","type":"mysql","z":"13b95697.ee62a1","mydb":"786
fa99.5e8f158","name":"","x":860,"y":1820,"wires":[["7327e9e2.4942f8","e838
652c.650f5"]]},{"id":"9402b9dd.a9e21","type":"function","z":"13b95697.ee62
a1","name":"DELETE id=2, GD","func":"\nmsg.topic=\"DELETE FROM
GroupDelayBBB WHERE id=2\";\n\nreturn
msg;","outputs":1,"noerr":0,"x":1110,"y":1740,"wires":[["cff3ac06.fc8c88"]
]},{"id":"24cb3018.cf7ad8","type":"function","z":"13b95697.ee62a1","name":
"UPDATE id=2, GD","func":"\nmsg.topic=\"UPDATE GroupDelayBBB SET id=2
WHERE id=1\";\n\n\nreturn
msg;","outputs":1,"noerr":0,"x":1110,"y":1780,"wires":[["4d6247c8.9e57e8"]
]},{"id":"ad4a0480.5a0bf","type":"mysql","z":"13b95697.ee62a1","mydb":"786
fa99.5e8f158","name":"","x":940,"y":1780,"wires":[["24cb3018.cf7ad8"]]},{"
id":"b00ecef1.6ce038","type":"mysql","z":"13b95697.ee62a1","mydb":"786fa99
.5e8f158","name":"","x":940,"y":1740,"wires":[["9402b9dd.a9e21"]]},{"id":"
c3ba047c.3663e","type":"change","z":"13b95697.ee62a1","name":"Broadcast","
rules":[{"t":"set","p":"payload","pt":"msg","to":"{\"state\":6}","tot":"js
on"}],"action":"","property":"","from":"","to":"","reg":false,"x":1160,"y"
:1860,"wires":[["8a972673.fdfe78"]]},{"id":"cff3ac06.fc8c88","type":"mysql
","z":"13b95697.ee62a1","mydb":"786fa99.5e8f158","name":"","x":1280,"y":17
40,"wires":[["c9df3254.3d5528"]]},{"id":"4d6247c8.9e57e8","type":"mysql","
z":"13b95697.ee62a1","mydb":"786fa99.5e8f158","name":"","x":1280,"y":1780,
"wires":[["f3026450.daf2a"]]},{"id":"60292eac.b0893","type":"link
out","z":"13b95697.ee62a1","name":"","links":["8094e49f.d67768"],"x":575,"
y":1720,"wires":[]},{"id":"303d1197.35194e","type":"link
out","z":"13b95697.ee62a1","name":"End_state2","links":["8094e49f.d67768"]
,"x":575,"y":1820,"wires":[]},{"id":"3797599f.10bb06","type":"change","z":
"13b95697.ee62a1","name":"Message to
client","rules":[{"t":"set","p":"payload","pt":"msg","to":"{\"state\":2}",
"tot":"json"}],"action":"","property":"","from":"","to":"","reg":false,"x"
:450,"y":1820,"wires":[["303d1197.35194e"]]},{"id":"8999b0aa.a6489","type"
:"inject","z":"13b95697.ee62a1","name":"Start","topic":"1","payload":"","p
ayloadType":"date","repeat":"","crontab":"","once":true,"x":110,"y":920,"w
ires":[["af1cf5d3.d8f258"]]},{"id":"af1cf5d3.d8f258","type":"link
out","z":"13b95697.ee62a1","name":"","links":["4ca04501.54609c"],"x":195,"
y":920,"wires":[]},{"id":"d205cdec.8f204","type":"change","z":"13b95697.ee
62a1","name":"topic =
8","rules":[{"t":"set","p":"topic","pt":"msg","to":"8","tot":"str"}],"acti

on":"","property":"","from":"","to":"","reg":false,"x":740,"y":280,"wires":[["313ee27e.991876"]]},{"id":"ef7a28e2.09f3f","type":"function","z":"13b95697.ee62a1","name":"DELETE id=2, Average","func":"\nmsg.topic=\"DELETE FROM AverageBBB WHERE id=2\";\n\nreturn msg;","outputs":1,"noerr":0,"x":830,"y":1500,"wires":[["f33c9ec2.cb8be"]]},{"id":"e4231a8c.fd317","type":"function","z":"13b95697.ee62a1","name":"UPDATE id=2, Average","func":"\nmsg.topic=\"UPDATE AverageBBB SET id=2 WHERE id=1\";\n\n\nreturn msg;","outputs":1,"noerr":0,"x":830,"y":1540,"wires":[["cd719095.6ad4a8"]]},{"id":"427d12bd.33fabc","type":"function","z":"13b95697.ee62a1","name":"Get Results","func":"msg.topic = \"SELECT frequencyNew, average, GroupDelay FROM AverageBBB LEFT JOIN GroupDelayBBB ON AverageBBB.frequencyNew = GroupDelayBBB.frequencyGD\";\nreturn msg;","outputs":1,"noerr":0,"x":790,"y":1580,"wires":[["a458425f.147378"]]},{"id":"a458425f.147378","type":"mysql","z":"13b95697.ee62a1","mydb":"786fa99.5e8f158","name":"","x":940,"y":1580,"wires":[["6b4437df.ed02","c7ffa23b.0c1378"]]},{"id":"cd719095.6ad4a8","type":"mysql","z":"13b95697.ee62a1","mydb":"786fa99.5e8f158","name":"","x":1020,"y":1540,"wires":[["ad473d50.1f25"]]},{"id":"f33c9ec2.cb8be","type":"mysql","z":"13b95697.ee62a1","mydb":"786fa99.5e8f158","name":"","x":1020,"y":1500,"wires":[["ff2fa084.bee1e"]]},{"id":"ff2fa084.bee1e","type":"function","z":"13b95697.ee62a1","name":"DELETE id=2, GD","func":"\nmsg.topic=\"DELETE FROM GroupDelayBBB WHERE id=2\";\n\nreturn msg;","outputs":1,"noerr":0,"x":1190,"y":1500,"wires":[["7bbe9069.a862e"]]},{"id":"ad473d50.1f25","type":"function","z":"13b95697.ee62a1","name":"UPDATE id=2, GD","func":"\nmsg.topic=\"UPDATE GroupDelayBBB SET id=2 WHERE id=1\";\n\n\nreturn msg;","outputs":1,"noerr":0,"x":1190,"y":1540,"wires":[["1b7d7334.41e98d"]]},{"id":"6b4437df.ed02","type":"csv","z":"13b95697.ee62a1","name":"","sep":",","hdrin":"","hdrout":true,"multi":"one","ret":"\\n","temp":"frequencyNew, average, GroupDelay","x":1090,"y":1640,"wires":[["76d24604.60b01","fd6352e6.a46ba"]]},{"id":"76d24604.60b01","type":"file","z":"13b95697.ee62a1","name":"","filename":"/var/www/html/BeagleSite_v3/results.csv","appendNewline":true,"createDir":false,"overwriteFile":"true","x":1340,"y":1660,"wires":[]},{"id":"fd6352e6.a46ba","type":"change","z":"13b95697.ee62a1","name":"Broadcast","rules":[{"t":"set","p":"payload","pt":"msg","to":"{\"state\":8}","tot":"json"},{"t":"delete","p":"topic","pt":"msg"}],"action":"","property":"","from":"","to":"","reg":false,"x":1240,"y":1620,"wires":[["8310a1c8.e1d01"]]},{"id":"8310a1c8.e1d01","type":"link out","z":"13b95697.ee62a1","name":"End_state7","links":["8094e49f.d67768"],"x":1335,"y":1620,"wires":[]},{"id":"1b7d7334.41e98d","type":"mysql","z":"13b95697.ee62a1","mydb":"786fa99.5e8f158","name":"","x":1360,"y":1540,"wires":[["427d12bd.33fabc"]]},{"id":"7bbe9069.a862e","type":"mysql","z":"13b95697.ee62a1","mydb":"786fa99.5e8f158","name":"","x":1360,"y":1500,"wires":[["e4231a8c.fd317"]]},{"id":"6fdb88e4.f2433","type":"join","z":"13b95697.ee62a1","name":"","mode":"custom","build":"string","property":"topic","propertyType":"msg","key":"topic","joiner":"\\n","timeout":"","count":"","x":150,"y":1020,"wires":[["255d75f0.729cda"]]},{"id":"fab854d4.15ba18","type":"link in","z":"13b95697.ee62a1","name":"Avg_wait_point","links":["80f57be4.87928","c59d66f.22fd998"],"x":55,"y":1020,"wires":[["6fdb88e4.f2433"]]},{"id":"1e29f56a.22f0c3","type":"join","z":"13b95697.ee62a1","name":"","mode":"custom","build":"string","property":"topic","propertyType":"msg","key":"topic","joiner":"\\n","timeout":"","count":"","x":150,"y":1120,"wires":[["30d61aac.ef23be"]]},{"id":"b5c86e36.1c3a5","type":"link in","z":"13b95697.ee62a1","name":"GD_wait_point","links":["61a7f26.d6a780c","7abfcfb7.06b8a8"],"x":55,"y":1120,"wires":[["1e29f56a.22f0c3"]]},{"id":"f4f21f55.144188","type":"change","z":"13b95697.ee62a1","name":"topic = 0","rules":[{"t":"set","p":"topic","pt":"msg","to":"0","tot":"str"}],"action":"","property":"","from":"","to":"","reg":false,"x":500,"y":440,"wires"

:[["7bafe213.6e6f5c"]]},{"id":"7bafe213.6e6f5c","type":"link out","z":"13b95697.ee62a1","name":"","links":["4ca04501.54609c"],"x":595,"y":440,"wires":[]},{"id":"f1aedc63.45b46","type":"link in","z":"13b95697.ee62a1","name":"http_report","links":["70625208.dae7ac"],"x":635,"y":440,"wires":[["c252e895.d51c88"]]},{"id":"70625208.dae7ac","type":"link out","z":"13b95697.ee62a1","name":"","links":["f1aedc63.45b46"],"x":375,"y":1900,"wires":[]},{"id":"255d75f0.729cda","type":"change","z":"13b95697.ee62a1","name":"topic = 6","rules":[{"t":"set","p":"topic","pt":"msg","to":"6","tot":"str"}],"action":"","property":"","from":"","to":"","reg":false,"x":280,"y":1020,"wires":[["c7dec9a1.0279f"]]},{"id":"30d61aac.ef23be","type":"change","z":"13b95697.ee62a1","name":"topic = 7","rules":[{"t":"set","p":"topic","pt":"msg","to":"7","tot":"str"}],"action":"","property":"","from":"","to":"","reg":false,"x":280,"y":1120,"wires":[["85960953.73879"]]},{"id":"c7dec9a1.0279f","type":"link out","z":"13b95697.ee62a1","name":"","links":["4ca04501.54609c"],"x":375,"y":1020,"wires":[]},{"id":"bcccad1b.6d7938","type":"change","z":"13b95697.ee62a1","name":"complete = true","rules":[{"t":"set","p":"complete","pt":"msg","to":"true","tot":"bool"}],"action":"","property":"","from":"","to":"","reg":false,"x":1240,"y":1420,"wires":[["80f57be4.87928"]]},{"id":"7df8979c.f9eee8","type":"comment","z":"13b95697.ee62a1","name":"Start of program","info":"","x":120,"y":880,"wires":[]},{"id":"1c4a2846.9e0df","type":"comment","z":"13b95697.ee62a1","name":"Averaging waits for the measurements","info":"","x":190,"y":980,"wires":[]},{"id":"b168957e.2ab96","type":"comment","z":"13b95697.ee62a1","name":"GD waits for the averages","info":"","x":150,"y":1080,"wires":[]},{"id":"4f4a8712.c18108","type":"change","z":"13b95697.ee62a1","name":"topic = 5","rules":[{"t":"set","p":"topic","pt":"msg","to":"5","tot":"str"}],"action":"","property":"","from":"","to":"","reg":false,"x":740,"y":320,"wires":[["36e42c89.fd5a6c"]]},{"id":"85960953.73879","type":"link out","z":"13b95697.ee62a1","name":"","links":["4ca04501.54609c"],"x":375,"y":1120,"wires":[]},{"id":"665ee13e.fce1c8","type":"comment","z":"13b95697.ee62a1","name":"Logic","info":"","x":90,"y":1180,"wires":[]},{"id":"3c1eff2a.b851d8","type":"exec","z":"13b95697.ee62a1","command":"cat /sys/class/gpio/gpio44/value","addpay":true,"append":"","useSpawn":"","timer":"","name":"P8_12.value => OC_OUT","x":370,"y":760,"wires":[["208a4f12.11f05"],[],[]]},{"id":"208a4f12.11f05","type":"change","z":"13b95697.ee62a1","name":"topic = OC","rules":[{"t":"set","p":"topic","pt":"msg","to":"OC","tot":"str"}],"action":"","property":"","from":"","to":"","reg":false,"x":570,"y":760,"wires":[["ff250e28.1337d"]]},{"id":"26087f6b.2a439","type":"websocket-listener","z":"","path":"/ws/simple","wholemsg":"false"},{"id":"d5193bf.585bac8","type":"MySQLdatabase","z":"","host":"localhost","port":"3306","db":"beagledb","tz":""},{"id":"786fa99.5e8f158","type":"MySQLdatabase","z":"","host":"localhost","port":"3306","db":"beagledb","tz":""}]

Synthesizer

[{"id":"d60c9168.4e3cd","type":"tab","label":"Synthesizer"},{"id":"9aaa556a.12c57","type":"serial out","z":"d60c9168.4e3cd","name":"","serial":"1e98266.1e9d45a","x":160,"y":140,"wires":[]},{"id":"6398030.804147c","type":"link in","z":"d60c9168.4e3cd","name":"SYNTH_out","links":["b5ed395c.545588"],"x":55,"y":140,"wires":[["9aaa556a.12c57"]]},{"id":"e39f62f1.79262","type":"debug","z":"d60c9168.4e3cd","name":"NEW MSG","active":false,"console":"false","complete":"payload","x":270,"y":380,"wires":[]},{"id":"d352d73d.4b1818","type":"serial in","z":"d60c9168.4e3cd","name":"","serial":"1e98266.1e9d45a","x":100,"y":320,"wires":[["7613b1a6.921f4","e39f62f1.79262"]]},{"id":"7613b1a6.921f4",

"type":"switch","z":"d60c9168.4e3cd","name":"Read Synth.
Response","property":"payload","propertyType":"msg","rules":[{"t":"cont","v":"A","vt":"str"},{"t":"cont","v":"R","vt":"str"},{"t":"cont","v":"L","vt":"str"},{"t":"cont","v":"U","vt":"str"}],"checkall":"true","outputs":4,"x":310,"y":320,"wires":[["346691e3.a43cde"],["4a58e014.fb15a"],["9c34bb57.c67b5"],["958bc0f.080674"]]},{"id":"4730007c.cce7c8","type":"debug","z":"d60c9168.4e3cd","name":"U","active":false,"console":"false","complete":"payload","x":690,"y":420,"wires":[]},{"id":"ff57ee5c.ad5d08","type":"delay","z":"d60c9168.4e3cd","name":"","pauseType":"delay","timeout":"3","timeoutUnits":"milliseconds","rate":"1","nbRateUnits":"1","rateUnits":"second","randomFirst":"1","randomLast":"5","randomUnits":"seconds","drop":false,"x":710,"y":380,"wires":[["be7aa3c0.3dc618"]]},{"id":"15aacf99.42068","type":"link
out","z":"d60c9168.4e3cd","name":"","links":["e13593ca.ed6ea8"],"x":635,"y":260,"wires":[]},{"id":"e6bdff10.314178","type":"link
out","z":"d60c9168.4e3cd","name":"","links":["a9783ed6.22283"],"x":635,"y":340,"wires":[]},{"id":"be7aa3c0.3dc618","type":"link
out","z":"d60c9168.4e3cd","name":"","links":["8f08f152.148398"],"x":815,"y":380,"wires":[]},{"id":"9c34bb57.c67b5","type":"change","z":"d60c9168.4e3cd","name":"lock =
true","rules":[{"t":"set","p":"lock","pt":"global","to":"true","tot":"bool"}],"action":"","property":"","from":"","to":"","reg":false,"x":530,"y":340,"wires":[["e6bdff10.314178"]]},{"id":"346691e3.a43cde","type":"change","z":"d60c9168.4e3cd","name":"lock =
false","rules":[{"t":"set","p":"lock","pt":"global","to":"false","tot":"bool"}],"action":"","property":"","from":"","to":"","reg":false,"x":530,"y":260,"wires":[["15aacf99.42068"]]},{"id":"4a58e014.fb15a","type":"change","z":"d60c9168.4e3cd","name":"lock =
false","rules":[{"t":"set","p":"lock","pt":"global","to":"false","tot":"bool"},{"t":"set","p":"synth_rejected","pt":"global","to":"1","tot":"num"}],"action":"","property":"","from":"","to":"","reg":false,"x":530,"y":300,"wires":[["955e5cd1.f37eb"]]},{"id":"958bc0f.080674","type":"change","z":"d60c9168.4e3cd","name":"lock =
false","rules":[{"t":"set","p":"lock","pt":"global","to":"false","tot":"bool"}],"action":"","property":"","from":"","to":"","reg":false,"x":530,"y":380,"wires":[["ff57ee5c.ad5d08","4730007c.cce7c8"]]},{"id":"276b56d1.8f35d2","type":"comment","z":"d60c9168.4e3cd","name":"Outgoing UART
port","info":"","x":128.21665954589844,"y":87.3499984741211,"wires":[]},{"id":"a953d7f2.fcf3a","type":"comment","z":"d60c9168.4e3cd","name":"Synthesizer
Communication","info":"","x":460,"y":40,"wires":[]},{"id":"74f2c078.78f198","type":"comment","z":"d60c9168.4e3cd","name":"Incoming UART
port","info":"","x":128.99998474121094,"y":221,"wires":[]},{"id":"955e5cd1.f37eb","type":"link
out","z":"d60c9168.4e3cd","name":"","links":["234c14c6.58bf2c","a0d1eeff.fa0a4","d0d0630.99a56a"],"x":635,"y":300,"wires":[]},{"id":"1e98266.1e9d45a","type":"serial-
port","z":"","serialport":"/dev/ttyS1","serialbaud":"115200","databits":"8","parity":"none","stopbits":"1","newline":"0x0D","bin":"false","out":"char","addchar":false}]

Measurements (BBB)

[{"id":"49c5eba8.eb31a4","type":"tab","label":"Measurements
(BBB)"},{"id":"debf51a3.0289b","type":"debug","z":"49c5eba8.eb31a4","name":"Measurement","active":false,"console":"false","complete":"true","x":520,"y":740,"wires":[]},{"id":"f4163840.8e9c98","type":"switch","z":"49c5eba8.eb31a4","name":"measurements_ended
?","property":"measurements_ended","propertyType":"msg","rules":[{"t":"eq","v":"0","vt":"num"},{"t":"else"}],"checkall":"true","outputs":2,"x":330,"y":560,"wires":[["1dd08ce.3a9c073"],["2761e0e8.a02a"]]},{"id":"22641429.6c

c2dc","type":"function","z":"49c5eba8.eb31a4","name":"Advance
loop","func":"msg.fstart+=flow.get(\"fstep\");\nflow.set(\"fstart\",msg.fs
tart);\n\nvar endTime = new Date();\nvar seconds = Math.round((endTime -
flow.get(\"timestamp\"))/1000);\n// 300 sec => 5 min timeout\nif
((msg.fstart > flow.get(\"fstop\")) || (flow.get(\"measurements_stop\") ==
1) || (seconds > 299)){\n   msg.measurements_ended = 1;\n}\nelse{\n
msg.measurements_ended = 0;\n}\nvar counter1 = flow.get(\"fcounter1\") +
1;\nif (counter1 == 5){\n   msg.flush =
true;\n}\nflow.set(\"fcounter1\",counter1);\nflow.set(\"fcounter2\",flow.g
et(\"fcounter2\")+1);\nreturn
msg;","outputs":1,"noerr":0,"x":520,"y":700,"wires":[["b20ea6f5.a7fbc8","6
babbfc6.cc92a8"]]},{"id":"e40e6716.2112f8","type":"function","z":"49c5eba8
.eb31a4","name":"Combine","func":"var input = msg.payload;\nvar
freq=flow.get(\"fstart\");\nvar volt=input.split(\",\");\nmsg.frequency =
freq.toFixed(3);\nmsg.volt = volt;\nmsg.fstart = freq;\nreturn
msg;","outputs":1,"noerr":0,"x":340,"y":720,"wires":[["debf51a3.0289b","22
641429.6cc2dc","fe374e3.26bdeb"]]},{"id":"8ca65845.ea9138","type":"functio
n","z":"49c5eba8.eb31a4","name":"Get Input","func":"var
startf=Number(parseFloat(msg.inputs.fstart).toFixed(3));\nvar
stopf=Number(parseFloat(msg.inputs.fstop).toFixed(3));\nvar
stepf=Number(parseFloat(msg.inputs.fstep).toFixed(3));\n\nflow.set(\"fstar
t\",startf);\nflow.set(\"fstop\",stopf);\nflow.set(\"fstep\",stepf);\nflow
.set(\"measurements_stop\",0);\nflow.set(\"fcounter1\",0);\nflow.set(\"fco
unter2\",0);\nflow.set(\"famount\",(stopf-startf)/stepf +
1);\nflow.set(\"timestamp\",new Date());\n\nreturn
msg;","outputs":1,"noerr":0,"x":140,"y":180,"wires":[["dbf2f2cc.27a63","f6
b111a6.ebaf4"]]},{"id":"1dd08ce.3a9c073","type":"function","z":"49c5eba8.e
b31a4","name":"Synth. Frequency Command","func":"var input =
Number(msg.fstart.toFixed(3))*1000;\nvar command =
\">00F\"+input.toString();\nmsg.payload = command;\nreturn
msg;","outputs":1,"noerr":0,"x":200,"y":380,"wires":[["4ab60021.154b7"]]},
{"id":"225017e3.13b998","type":"function","z":"49c5eba8.eb31a4","name":"As
k Synth. Status","func":"msg.payload = \">00?\";\nmsg.payload =
msg.payload+String.fromCharCode(13); \nreturn
msg;","outputs":1,"noerr":0,"x":450,"y":440,"wires":[["b5ed395c.545588"]]}
,{"id":"21b3eb35.a4b1b4","type":"exec","z":"49c5eba8.eb31a4","command":"/h
ome/debian/temp.out","addpay":false,"append":"","useSpawn":"","timer":"","
name":"Call to ADC
utility","x":160,"y":740,"wires":[["e40e6716.2112f8"],[],[]]},{"id":"c28c9
e26.eb9f2","type":"bbb-discrete-
out","z":"49c5eba8.eb31a4","pin":"P9_12","inverting":false,"toggle":false,
"defaultState":"0","name":"DE","x":430,"y":180,"wires":[["c0acd619.6bde88"
]]},{"id":"c0acd619.6bde88","type":"bbb-discrete-
out","z":"49c5eba8.eb31a4","pin":"P8_15","inverting":false,"toggle":false,
"defaultState":"0","name":"12V_EN1","x":560,"y":180,"wires":[["c2eaf1da.40
e9f"]]},{"id":"c2eaf1da.40e9f","type":"bbb-discrete-
out","z":"49c5eba8.eb31a4","pin":"P8_16","inverting":false,"toggle":false,
"defaultState":"0","name":"12V_EN2","x":700,"y":180,"wires":[["7b97c68c.38
b42"]]},{"id":"ce21b0f2.5b3f7","type":"bbb-discrete-
out","z":"49c5eba8.eb31a4","pin":"P9_41","inverting":false,"toggle":false,
"defaultState":"0","name":"5V_EN","x":600,"y":220,"wires":[["6ace6978.78b2
a8"]]},{"id":"322fa07b.2de08","type":"bbb-discrete-
out","z":"49c5eba8.eb31a4","pin":"P9_15","inverting":false,"toggle":false,
"defaultState":"0","name":"RE~","x":870,"y":220,"wires":[["2422920a.83a78e
"]]},{"id":"c27b9759.d5f258","type":"bbb-discrete-
out","z":"49c5eba8.eb31a4","pin":"P9_12","inverting":false,"toggle":false,
"defaultState":"0","name":"DE","x":810,"y":480,"wires":[["83d95c06.aac22"]
]},{"id":"4e640e25.860ce","type":"bbb-discrete-
out","z":"49c5eba8.eb31a4","pin":"P8_16","inverting":false,"toggle":false,
"defaultState":"0","name":"12V_EN2","x":1040,"y":560,"wires":[["82f54a2f.1
58d48"]]},{"id":"82f54a2f.158d48","type":"bbb-discrete-

out","z":"49c5eba8.eb31a4","pin":"P9_15","inverting":false,"toggle":false,
"defaultState":"0","name":"RE~","x":590,"y":600,"wires":[["4df870c0.3e21e"
]]},{"id":"e4831713.cf28e","type":"debug","z":"49c5eba8.eb31a4","name":"SE
NT","active":false,"console":"false","complete":"true","x":510,"y":360,"wi
res":[]},{"id":"4ab60021.154b7","type":"function","z":"49c5eba8.eb31a4","n
ame":"CR","func":"msg.payload = msg.payload+String.fromCharCode(13);
\nreturn
msg;","outputs":1,"noerr":0,"x":390,"y":380,"wires":[["e4831713.cf28e","b5
ed395c.545588"]]},{"id":"321e356e.b1cf6a","type":"delay","z":"49c5eba8.eb3
1a4","name":"","pauseType":"delay","timeout":"1","timeoutUnits":"seconds",
"rate":"1","nbRateUnits":"1","rateUnits":"second","randomFirst":"1","rando
mLast":"5","randomUnits":"seconds","drop":false,"x":540,"y":260,"wires":[[
"1dd08ce.3a9c073"]]},{"id":"42d78505.06fccc","type":"link
in","z":"49c5eba8.eb31a4","name":"Measurements_flow_in","links":["3fcbbafa
.3f5186","5fac3116.b3eff8","f7fcf1bd.3bc268","783f92aa.17b0dc"],"x":35,"y"
:180,"wires":[["8ca65845.ea9138"]]},{"id":"3ca99234.dce32e","type":"link
in","z":"49c5eba8.eb31a4","name":"Measurements_stop_in","links":["3a66b7aa
.901188","d380d1bb.a3bf68","36e42c89.fd5a6c"],"x":35,"y":900,"wires":[["9c
86e24c.2611e"]]},{"id":"363d0e8d.adeaf2","type":"debug","z":"49c5eba8.eb31
a4","name":"Stop","active":false,"console":"false","complete":"payload","x
":370,"y":900,"wires":[]},{"id":"fe374e3.26bdeb","type":"function","z":"49
c5eba8.eb31a4","name":"Write Measurement to CSV","func":"var fin =
\"\";\nfor (i = 0; i < 10; i++){\n   fin =
fin.concat(msg.frequency).concat(\",\").concat(msg.volt[i]).concat(\"\\n\"
);\n}\nmsg.payload = fin;\nreturn
msg;","outputs":1,"noerr":0,"x":540,"y":800,"wires":[["1856a8db.f8f707"]]}
,{"id":"1856a8db.f8f707","type":"file","z":"49c5eba8.eb31a4","name":"","fi
lename":"/home/debian/measurements.csv","appendNewline":false,"createDir":
false,"overwriteFile":"false","x":820,"y":800,"wires":[]},{"id":"dbf2f2cc.
27a63","type":"function","z":"49c5eba8.eb31a4","name":"Log","func":"msg.pa
yload =\"frequency,measurement\".concat(\"\\n\");\nreturn
msg;","outputs":1,"noerr":0,"x":290,"y":140,"wires":[["2cd447d6.e6e1d8"]]}
,{"id":"2cd447d6.e6e1d8","type":"file","z":"49c5eba8.eb31a4","name":"","fi
lename":"/home/debian/measurements.csv","appendNewline":false,"createDir":
false,"overwriteFile":"true","x":500,"y":140,"wires":[]},{"id":"f6b111a6.e
baf4","type":"change","z":"49c5eba8.eb31a4","name":"Set_high","rules":[{"t
":"set","p":"payload","pt":"msg","to":"1","tot":"num"}],"action":"","prope
rty":"","from":"","to":"","reg":false,"x":300,"y":180,"wires":[["c28c9e26.
eb9f2"]]},{"id":"6ace6978.78b2a8","type":"change","z":"49c5eba8.eb31a4","n
ame":"Set_low","rules":[{"t":"set","p":"payload","pt":"msg","to":"0","tot"
:"num"}],"action":"","property":"","from":"","to":"","reg":false,"x":740,"
y":220,"wires":[["322fa07b.2de08"]]},{"id":"2422920a.83a78e","type":"funct
ion","z":"49c5eba8.eb31a4","name":"Get Starting
Frequency","func":"msg.fstart = flow.get(\"fstart\");\nreturn
msg;","outputs":1,"noerr":0,"x":350,"y":260,"wires":[["321e356e.b1cf6a"]]}
,{"id":"b5ed395c.545588","type":"link
out","z":"49c5eba8.eb31a4","name":"","links":["6398030.804147c"],"x":595,"
y":380,"wires":[]},{"id":"e13593ca.ed6ea8","type":"link
in","z":"49c5eba8.eb31a4","name":"Synth_response_A","links":["15aacf99.420
68"],"x":315,"y":420,"wires":[["225017e3.13b998"]]},{"id":"8f08f152.148398
","type":"link
in","z":"49c5eba8.eb31a4","name":"Synth_response_U","links":["be7aa3c0.3dc
618"],"x":315,"y":460,"wires":[["225017e3.13b998"]]},{"id":"a9783ed6.22283
","type":"link
in","z":"49c5eba8.eb31a4","name":"Synth_response_L","links":["e6bdff10.314
178"],"x":35,"y":740,"wires":[["21b3eb35.a4b1b4"]]},{"id":"b20ea6f5.a7fbc8
","type":"link
out","z":"49c5eba8.eb31a4","name":"Measurements_flow_out_not_ended","links
":["eacfb595.addf58"],"x":675,"y":700,"wires":[]},{"id":"2761e0e8.a02a","t
ype":"change","z":"49c5eba8.eb31a4","name":"Set_low & lock =
false","rules":[{"t":"set","p":"payload","pt":"msg","to":"0","tot":"num"},

{"t":"set","p":"lock","pt":"global","to":"false","tot":"bool"}],"action":"
","property":"","from":"","to":"","reg":false,"x":640,"y":480,"wires":[["c
27b9759.d5f258"]]},{"id":"83d95c06.aac22","type":"change","z":"49c5eba8.eb
31a4","name":"Set_low","rules":[{"t":"set","p":"payload","pt":"msg","to":"
0","tot":"num"}],"action":"","property":"","from":"","to":"","reg":false,"
x":940,"y":480,"wires":[["32861216.ebdd66"]]},{"id":"e036e66a.e5b1d","type
":"change","z":"49c5eba8.eb31a4","name":"Set_low","rules":[{"t":"set","p":
"payload","pt":"msg","to":"0","tot":"num"}],"action":"","property":"","fro
m":"","to":"","reg":false,"x":600,"y":560,"wires":[["4ab641cc.a37be"]]},{"
id":"9982f8e4.db2898","type":"change","z":"49c5eba8.eb31a4","name":"Set_lo
w","rules":[{"t":"set","p":"payload","pt":"msg","to":"0","tot":"num"}],"ac
tion":"","property":"","from":"","to":"","reg":false,"x":880,"y":560,"wire
s":[["4e640e25.860ce"]]},{"id":"9c86e24c.2611e","type":"change","z":"49c5e
ba8.eb31a4","name":"measurements_stop =
1","rules":[{"t":"set","p":"measurements_stop","pt":"flow","to":"1","tot":
"num"}],"action":"","property":"","from":"","to":"","reg":false,"x":190,"y
":900,"wires":[["363d0e8d.adeaf2"]]},{"id":"4197f962.eaeef8","type":"link
out","z":"49c5eba8.eb31a4","name":"","links":["4ca04501.54609c","860709d0.
c20358"],"x":935,"y":600,"wires":[]},{"id":"6babbfc6.cc92a8","type":"switc
h","z":"49c5eba8.eb31a4","name":"fcounter1 ==
5","property":"fcounter1","propertyType":"flow","rules":[{"t":"else"},{"t"
:"eq","v":"5","vt":"num"}],"checkall":"true","outputs":2,"x":120,"y":580,"
wires":[["f4163840.8e9c98"],["83f0386a.1f08d8"]]},{"id":"83f0386a.1f08d8",
"type":"function","z":"49c5eba8.eb31a4","name":"%
%","func":"flow.set(\"fcounter1\",0);\nmsg.payload =
flow.get(\"fcounter2\") / flow.get(\"famount\");\nmsg.payload =
Number(parseFloat(msg.payload).toFixed(2));\nreturn
msg;","outputs":1,"noerr":0,"x":270,"y":600,"wires":[["2d2e817d.496586"]]}
,{"id":"2d2e817d.496586","type":"template","z":"49c5eba8.eb31a4","name":"J
SON","field":"payload","fieldType":"msg","format":"handlebars","syntax":"m
ustache","template":"{\"state\":0,\"percentage\":{{payload}},\"new_data\":
[{\"frequencyNew\":\"13450.000\",\"average\":\"3617.6250000000000000000\"
},{\"frequencyNew\":\"13451.000\",\"average\":\"3606.3750000000000000000\
"}]}","x":390,"y":600,"wires":[["9c00a931.6107b"]]},{"id":"9c00a931.6107b"
,"type":"link
out","z":"49c5eba8.eb31a4","name":"","links":["8094e49f.d67768"],"x":475,"
y":600,"wires":[]},{"id":"b0497491.240c98","type":"comment","z":"49c5eba8.
eb31a4","name":"Measurements
Loop","info":"","x":470,"y":40,"wires":[]},{"id":"234c14c6.58bf2c","type":
"link
in","z":"49c5eba8.eb31a4","name":"Synth_response_R","links":["955e5cd1.f37
eb"],"x":475,"y":480,"wires":[["2761e0e8.a02a"]]},{"id":"4df870c0.3e21e","
type":"function","z":"49c5eba8.eb31a4","name":"Get 'fstart' if synth.
rejected","func":"if (global.get(\"synth_rejected\")==1){\n   msg.fstart
= flow.get(\"fstart\");\n   msg.topic = \"4\";\n}\nelse{\n   msg.topic =
\"3\";\n}\nreturn
msg;","outputs":1,"noerr":0,"x":780,"y":600,"wires":[["4197f962.eaeef8"]]}
,{"id":"83fb3296.01ad8","type":"comment","z":"49c5eba8.eb31a4","name":"Mai
n
Loop","info":"","x":80,"y":100,"wires":[]},{"id":"e0b4ba15.40c6c","type":"
comment","z":"49c5eba8.eb31a4","name":"Stop
case","info":"","x":80,"y":860,"wires":[]},{"id":"7b97c68c.38b42","type":"
delay","z":"49c5eba8.eb31a4","name":"","pauseType":"delay","timeout":"5","
timeoutUnits":"seconds","rate":"1","nbRateUnits":"1","rateUnits":"second",
"randomFirst":"1","randomLast":"5","randomUnits":"seconds","drop":false,"x
":840,"y":180,"wires":[["ac3761a7.490c9"]]},{"id":"ac3761a7.490c9","type":
"bbb-discrete-
out","z":"49c5eba8.eb31a4","pin":"P8_12","inverting":false,"toggle":false,
"defaultState":"0","name":"OC_OUT_1","x":310,"y":220,"wires":[["74e64d2b.8
7e44c"]]},{"id":"74e64d2b.87e44c","type":"delay","z":"49c5eba8.eb31a4","na
me":"","pauseType":"delay","timeout":"5","timeoutUnits":"seconds","rate":"

1","nbRateUnits":"1","rateUnits":"second","randomFirst":"1","randomLast":"5","randomUnits":"seconds","drop":false,"x":460,"y":220,"wires":[["ce21b0f2.5b3f7"]]},{"id":"32861216.ebdd66","type":"bbb-discrete-out","z":"49c5eba8.eb31a4","pin":"P9_41","inverting":false,"toggle":false,"defaultState":"0","name":"5V_EN","x":1080,"y":480,"wires":[["2300ff49.2b299"]]},{"id":"4ab641cc.a37be","type":"bbb-discrete-out","z":"49c5eba8.eb31a4","pin":"P8_15","inverting":false,"toggle":false,"defaultState":"0","name":"12V_EN1","x":740,"y":560,"wires":[["9982f8e4.db2898"]]},{"id":"2300ff49.2b299","type":"delay","z":"49c5eba8.eb31a4","name":"","pauseType":"delay","timeout":"5","timeoutUnits":"seconds","rate":"1","nbRateUnits":"1","rateUnits":"second","randomFirst":"1","randomLast":"5","randomUnits":"seconds","drop":false,"x":600,"y":520,"wires":[["1ad4fcbf.8afbdb"]]},{"id":"c3122429.c9a548","type":"bbb-discrete-out","z":"49c5eba8.eb31a4","pin":"P8_12","inverting":false,"toggle":false,"defaultState":"0","name":"OC_OUT_1","x":890,"y":520,"wires":[["f9c48f9e.d0bb3"]]},{"id":"1ad4fcbf.8afbdb","type":"change","z":"49c5eba8.eb31a4","name":"Set_low","rules":[{"t":"set","p":"payload","pt":"msg","to":"0","tot":"num"}],"action":"","property":"","from":"","to":"","reg":false,"x":740,"y":520,"wires":[["c3122429.c9a548"]]},{"id":"f9c48f9e.d0bb3","type":"delay","z":"49c5eba8.eb31a4","name":"","pauseType":"delay","timeout":"5","timeoutUnits":"seconds","rate":"1","nbRateUnits":"1","rateUnits":"second","randomFirst":"1","randomLast":"5","randomUnits":"seconds","drop":false,"x":1040,"y":520,"wires":[["e036e66a.e5b1d"]]}]

Average (BBB)

[{"id":"3b610e8a.a0bc22","type":"tab","label":"Average (BBB)"},{"id":"eacfb595.addf58","type":"link in","z":"3b610e8a.a0bc22","name":"Averaging_flow_in_new","links":["3606e632.af057a","a78b9b8f.743b6","c5fe731.c7e209","2adcd9cc.d8ccd6","b20ea6f5.a7fbc8","3a59cd6b.b04fc2","32bf0d3f.8669a2","aa16de7b.2740a"],"x":35,"y":180,"wires":[["d6c01952.6dec48"]]},{"id":"d6c01952.6dec48","type":"function","z":"3b610e8a.a0bc22","name":"Average calc.","func":"var max=-Infinity, min=+Infinity, sum=0;\nvar volt = msg.volt;\nvar value;\nfor (i=0; i<30; i++){\n   value = Number(volt[i]);\n   if (value > max) {\n   max = value;\n   }if (value < min) {\n   min = value;\n   }\n   sum+=value;\n}\n\nvar ave = (sum - max -min)/28;\n\nmsg.payload={};\nmsg.payload.frequency=msg.frequency;\nmsg.payload.average=ave;\n\nvar msg2 = {};\nmsg2.frequencyNew = msg.payload.frequency;\nmsg2.average = msg.payload.average;\nvar buf = global.get(\"acc_buff\");\nbuf.array.push(msg2);\nglobal.set(\"acc_buff\", buf);\n\nreturn msg;\n\n\n","outputs":"1","noerr":0,"x":160,"y":180,"wires":[["b4c1639b.7f338","8e1e7c87.ed569","224371f5.8a837e"]]},{"id":"b4c1639b.7f338","type":"function","z":"3b610e8a.a0bc22","name":"INSERT INTO AverageBBB","func":"\nmsg.topic=\"INSERT INTO AverageBBB(frequencyNew,average,id) VALUES (\"+msg.payload.frequency+\",\"+msg.payload.average+\",1)\";\n\n\nreturn msg;","outputs":1,"noerr":0,"x":400,"y":180,"wires":[["62bde30e.3343bc"]]},{"id":"62bde30e.3343bc","type":"mysql","z":"3b610e8a.a0bc22","mydb":"f5e5a367.37581","name":"","x":600,"y":180,"wires":[["fcf58269.478f6"]]},{"id":"fcf58269.478f6","type":"switch","z":"3b610e8a.a0bc22","name":"measurements_ended ?","property":"measurements_ended","propertyType":"msg","rules":[{"t":"eq","v":"1","vt":"str"}],"checkall":"true","outputs":1,"x":790,"y":180,"wires":[["c59d66f.22fd998"]]},{"id":"c59d66f.22fd998","type":"link out","z":"3b610e8a.a0bc22","name":"Averaging_flow_out","links":["892445d3.153e78","fab854d4.15ba18","f465f174.c3f28"],"x":935,"y":180,"wires":[]},{"

id":"8e1e7c87.ed569","type":"link out","z":"3b610e8a.a0bc22","name":"","links":["f6905b85.aa2b68","1a45b3b4.0f7f24"],"x":295,"y":220,"wires":[]},{"id":"1ac2c2cf.275115","type":"comment","z":"3b610e8a.a0bc22","name":"Averaging Implementation","info":"","x":470,"y":40,"wires":[]},{"id":"224371f5.8a837e","type":"switch","z":"3b610e8a.a0bc22","name":"flush = true ?","property":"flush","propertyType":"msg","rules":[{"t":"true"}],"checkall":"true","outputs":1,"x":350,"y":140,"wires":[["713452b2.6b70fc"]]},{"id":"d0e09e95.e64f68","type":"template","z":"3b610e8a.a0bc22","name":"Message","field":"payload","fieldType":"msg","format":"handlebars","syntax":"mustache","template":"{\"state\":0,\"percentage\":-1,\"new_data\":{{{payload}}}}","x":840,"y":140,"wires":[["9c7eecc9.c31698"]]},{"id":"9c7eecc9.c31698","type":"link out","z":"3b610e8a.a0bc22","name":"","links":["8094e49f.d67768"],"x":935,"y":140,"wires":[]},{"id":"713452b2.6b70fc","type":"change","z":"3b610e8a.a0bc22","name":"payload = buff","rules":[{"t":"set","p":"payload","pt":"msg","to":"acc_buff.array","tot":"global"},{"t":"set","p":"acc_buff.array","pt":"global","to":"[]","tot":"json"}],"action":"","property":"","from":"","to":"","reg":false,"x":540,"y":140,"wires":[["c0070196.b89548"]]},{"id":"c0070196.b89548","type":"json","z":"3b610e8a.a0bc22","name":"=> JSON","x":700,"y":140,"wires":[["d0e09e95.e64f68"]]},{"id":"892445d3.153e78","type":"link in","z":"3b610e8a.a0bc22","name":"flush","links":["c59d66f.22fd998"],"x":395,"y":100,"wires":[["713452b2.6b70fc"]]},{"id":"f5e5a367.37581","type":"MySQLdatabase","z":"","host":"localhost","port":"3306","db":"beagledb","tz":""}]

Group Delay (BBB)

[{"id":"40698fdb.fbaef","type":"tab","label":"Group Delay (BBB)"},{"id":"1a45b3b4.0f7f24","type":"link in","z":"40698fdb.fbaef","name":"Group_Delay_in","links":["8e1e7c87.ed569"],"x":35,"y":140,"wires":[["729076d1.6f5568"]]},{"id":"b2170ce8.f4308","type":"function","z":"40698fdb.fbaef","name":"Group Delay calc.","func":"//var ic=global.get(\"icount\"),\n// icp=global.get(\"icountplus\");\n\nvar buf = JSON.parse(global.get(\"buffer\"));\nmsg.test = buf.freq + \" \" + buf.value + \" \" + msg.payload.frequency + \" \" + msg.payload.average;\nvar FN = buf.freq,\n   a = buf.value,\n   FNP = Number(parseFloat(msg.payload.frequency).toFixed(3)),\n   aP = parseFloat(msg.payload.average);\n//var FN=msg.payload[ic].frequencyNew,\n// a=msg.payload[ic].average,\n   //FNP=msg.payload[icp].frequencyNew,\n //aP=msg.payload[icp].average;\n \nvar ar=-(aP-a),\n   par=360*(FNP-FN),\n   gd=ar/par;\n   \ngd = gd / 1000000000; \n//msg.payload={};\n//msg.payload.frequency=FN;\n//msg.payload.GroupDelay=gd;\nmsg.frequency=FN;\nmsg.GroupDelay=gd;\n\nbuf.freq = FNP;\nbuf.value = aP;\nglobal.set(\"buffer\",JSON.stringify(buf));\n\nvar msg2 = {};\nmsg2.frequencyGD = msg.frequency;\nmsg2.GroupDelay = msg.GroupDelay;\nvar buf = global.get(\"acc_buff_2\");\nbuf.array.push(msg2);\nglobal.set(\"acc_buff_2\", buf);\n\nreturn msg;","outputs":1,"noerr":0,"x":350,"y":160,"wires":[["e7ae44.fecae1c","c5e431ed.2ca8","f2e59294.e72fe"]]},{"id":"729076d1.6f5568","type":"switch","z":"40698fdb.fbaef","name":"buffer empty?","property":"buffer.freq","propertyType":"global","rules":[{"t":"eq","v":"0","vt":"num"},{"t":"else"}],"checkall":"true","outputs":2,"x":160,"y":140,"wires":[["98357417.add0f"],["b2170ce8.f4308"]]},{"id":"98357417.add0f","type":"template","z":"40698fdb.fbaef","name":"Store in buffer","field":"buffer","fieldType":"global","format":"handlebars","syntax":"mustache","template":"{\"freq\":{{payload.frequency}},\"value\":{{payl oad.average}}}","x":340,"y":120,"wires":[[]]},{"id":"c5e431ed.2ca8","type":"switch","z":"40698fdb.fbaef","name":"measurements_ended ?","property":"measurements_ended","propertyType":"msg","rules":[{"t":"eq","v":"1","vt":"str"}],"checkall":"true","outputs":1,"x":590,"y":200,"wires":[["5c3ec5e9.4779fc"]]},{"id":"61a7f26.d6a780c","type":"link out","z":"40698fdb.fbaef","name":"Group_Delay_flow_out","links":["6a2022f9.6f2a54","b5c86e36.1c3a5"],"x":915,"y":200,"wires":[]},{"id":"a398ed94.f18728","type":"mysql","z":"40698fdb.fbaef","mydb":"ce7e5677.1e7618","name":"","x":820,"y":160,"wires":[["93cca059.cc72f"]]},{"id":"e7ae44.fecae1c","type":"function","z":"40698fdb.fbaef","name":"INSERT INTO GroupDelayBBB","func":"\nmsg.topic=\"INSERT INTO GroupDelayBBB(frequencyGD,GroupDelay,id) VALUES (\"+msg.frequency+\",\"+msg.GroupDelay+\",1)\";\n\n\nreturn msg;","outputs":1,"noerr":0,"x":610,"y":160,"wires":[["a398ed94.f18728"]]},{"id":"93cca059.cc72f","type":"debug","z":"40698fdb.fbaef","name":"GD","active":false,"console":"false","complete":"payload","x":950,"y":160,"wires":[]},{"id":"5c3ec5e9.4779fc","type":"change","z":"40698fdb.fbaef","name":"topic = GDend","rules":[{"t":"set","p":"topic","pt":"msg","to":"GDend","tot":"str"},{"t":"set","p":"buffer","pt":"global","to":"{\"freq\":0,\"value\":0}","tot":"json"}],"action":"","property":"","from":"","to":"","reg":false,"x":800,"y":200,"wires":[["61a7f26.d6a780c"]]},{"id":"9e626260.b7b0a8","type":"comment","z":"40698fdb.fbaef","name":"Group Delay Implementation","info":"","x":460,"y":40,"wires":[]},{"id":"f2e59294.e72fe","type":"switch","z":"40698fdb.fbaef","name":"flush = true ?","property":"flush","propertyType":"msg","rules":[{"t":"true"}],"checkall":"true","outputs":1,"x":550,"y":120,"wires":[["3d327be0.29e1ac"]]},{"id":"e1e54b11.102698","type":"template","z":"40698fdb.fbaef","name":"Message","field":"payload","fieldType":"msg","format":"handlebars","syntax":"mustache","template":"{\"state\":0,\"percentage\":-2,\"new_data\":{{{payload}}}}","x":1040,"y":120,"wires":[["a4966b8f.b183"]]},{"id":"a4966b8f.b183","type":"link out","z":"40698fdb.fbaef","name":"","links":["8094e49f.d67768"],"x":1135,"y":120,"wires":[]},{"id":"3d327be0.29e1ac","type":"change","z":"40698fdb.fbaef","name":"payload = buff","rules":[{"t":"set","p":"payload","pt":"msg","to":"acc_buff_2.array","tot":"global"},{"t":"set","p":"acc_buff_2.array","pt":"global","to":"[]","tot":"json"}],"action":"","property":"","from":"","to":"","reg":false,"x":740,"y":120,"wires":[["60981820.23c928"]]},{"id":"60981820.23c928","type":"json","z":"40698fdb.fbaef","name":"=> JSON","x":900,"y":120,"wires":[["e1e54b11.102698"]]},{"id":"6a2022f9.6f2a54","type":"link in","z":"40698fdb.fbaef","name":"flush_GD","links":["61a7f26.d6a780c"],"x":595,"y":80,"wires":[["3d327be0.29e1ac"]]},{"id":"ce7e5677.1e7618","type":"MySQLdatabase","z":"","host":"localhost","port":"3306","db":"beagledb","tz":""}]

Κώδικας Διαδικτυακής Πλατφόρμας

ShowGraphs.php

```
<!DOCTYPE html>
<html lang="en">
<head>
<link rel="stylesheet" type="text/css" href="style.css">
<font color="white">
<div id="nav-bar">
BITE for Satellite Transponder Calibration -  Measurements UI
<button class="btn btn-default pull-right nav-btn"
onclick="window.location.href='login.php?logout=&quot;1&quot;'"><span
class="glyphicon glyphicon-log-out"></span> Logout</button>
<button class="btn btn-default pull-right nav-btn"
onclick="window.location.href='dashboard.php'"><span class="glyphicon glyphicon-
arrow-left"></span> Home</button>
<button class="btn btn-default pull-right nav-btn"
onClick="window.location.href='/BeagleSite_v3/results.csv'"><span class="glyphicon
glyphicon-save"></span> CSV</button>
</div>
</font>
<!--- TAB ------>
<h4 align="right"></h4>
<head>
<title>Satellite Transponder Measurements</title>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css">
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>
<script
src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>
</head>
<div class="container">
<br>
</div>
<!--- END of TAB --------------------------->
<title><font color="white">BeagleBoneMeasurements</title>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css">
<style>
.topright {
position: absolute;
top: 40px;
right: 100px;
font-size: 18px;
}
```

```
</style>
<div align="left|right|center|justify">
</head>
<body>
<! ------------------------------CHART --------------------------------------------
---------->
<center>
<style>
path.domain {
stroke: aquamarine;
}
.tick text {
fill:white
}
body {
background-color: #262f39;
}
.axis path,
.axis line{
fill: none;
stroke: black;
}
.line{
fill: none;
stroke: black;
stroke-width: 2px;
}
.tick text{
font-size: 12px;
}
.tick line{
opacity: 0.5;
}
.grid.tick {
stroke: black;
opacity: 0.7;
}
.grid line {
stroke: aquamarine;
stroke-opacity: 0.6;
shape-rendering: crispEdges;
}
.grid path {
stroke-width: 0;
}
```

```
circle {
fill: yellow;
}
path{
stroke: blue;
}
div.tooltip {
position: absolute;
text-align: center;
width: 100px;
height: 30px;
padding: 2px;
font: 12px sans-serif;
background: lightsteelblue;
border: 0px;
border-radius: 8px;
pointer-events: none;
}
</style>
<svg id="svg1" width="1100" height="400"></svg>
<svg id="svg2" width="1100" height="400"></svg>
<script src="https://d3js.org/d3.v4.min.js"></script>
<script>
// set the dimensions and margins of the graph
var svg1 = d3.select("#svg1")
margin = {top: 5, right: 0, bottom: 30, left: 55},
width = 1000- margin.left - margin.right,
height = 400 - margin.top - margin.bottom,
padding = 55;
g1 = svg1.append("g").attr("transform", "translate(" + margin.left + "," +
margin.top + ")");
// Set the ranges
var x = d3.scaleLinear()
.domain([13000, 14000])
.range([0, 1000])
.nice(10);
var y = d3.scaleLinear()
.rangeRound([height, 0]);
// Define the line
var line = d3.line()
.x(function(d) { return x(d.frequency); })
.y(function(d) { return y(d.output); });
// Define the div for the tooltip
var div = d3.select("body").append("div")
.attr("class", "tooltip")
.style("opacity", 0);
```

```
// Define the axes
var xAxis = d3.axisBottom(x)
.scale(x);

var yAxis = d3.axisLeft(y)
.scale(y);

// Define the div for the tooltip
//var div = d3.select("body").append("div")
// .attr("class", "tooltip")
// .style("opacity", 0);

// Get the data
//d3.csv("OUTPUT1.csv", function(d) {
// d.frequency = +d.frequency;
// d.output = +d.output;
// return d;
//},
// function(error, data) {
// if (error) throw error;

// Get the data
d3.json("php/Average.php", function(error, data) {
data.forEach(function(d) {
d.frequency = +d.frequencyNew;
d.output = +d.average;
});
if (error) throw error;

// Scale the range of the data
x.domain(d3.extent(data, function(d) { return d.frequency; })).nice();
y.domain(d3.extent(data, function(d) { return d.output; }));

// Add the X Axis
g1.append("g")
.attr("transform", "translate(0," + height + ")")
.call(xAxis)
.append("text")
.attr("fill", "#e5e5e5")
.attr("y", 27)
.attr("dx", "0.91em")
.attr("text-anchor", "end")
.text("Frequency");

// Add the Y Axis
g1.append("g")
.call(yAxis)
.append("text")
.attr("fill", "#e5e5e5")
.attr("transform", "rotate(-90)")
.attr("y", 17)
.attr("dy", "100em")
.attr("text-anchor", "end")
```

```
.text("Average Values");
// Add the valueline path.
g1.append("path")
.datum(data)
.attr("fill", "none")
.attr("stroke", "aquamarine")
.attr("stroke-linejoin", "round")
.attr("stroke-linecap", "round")
.attr("stroke-width", 4)
.attr("d", line)
;

// gridlines in x axis function
function make_x_gridlines() {
return d3.axisBottom(x)
.scale(x)
.ticks(10)
}
// gridlines in y axis function
function make_y_gridlines() {
return d3.axisLeft(y)
.scale(y)
.ticks(10)
}

// add the X gridlines
svg1.append("g")
.attr("class", "grid")
.attr("transform", "translate(55," + height + ")")
.call(make_x_gridlines()
.tickSize(-400)
.tickFormat("")
)
// add the Y gridlines
svg1.append("g")
.attr("class", "grid")
.attr("transform", "translate(" + 55 + "+5)" )
.call(make_y_gridlines()
.tickSize(-1000)
.tickFormat("")
)

// Add the scatterplot
svg1.selectAll("dot")
.data(data)
.enter().append("circle")
.attr("r", 2)
.attr("cx", function(d) { return x(d.frequency)+55; })
.attr("cy", function(d) { return y(d.output)+5; })
```

```
.on("mouseover", function(d) {
div.transition()
.duration(200)
.style("opacity", .9);
div .html(d.frequency + "<br/>" + d.output)
.style("left", (d3.event.pageX) + "px")
.style("top", (d3.event.pageY ) + "px");
})
.on("mouseout", function(d) {
div.transition()
.duration(400)
.style("opacity", 0);
});
});

/////////////////////////////////////////////////////SECOND
GRAPH///////////////////////////////////////////////////////

// set the dimensions and margins of the second graph
var svg2 = d3.select("#svg2"),
margin = {top: 5, right: 0, bottom: 30, left: 55},
width = 100- margin.left - margin.right,
height = 400 - margin.top - margin.bottom,
padding = 40;
g2 = svg2.append("g").attr("transform", "translate(" + margin.left + "," +
margin.top + ")");

// Get the data
//d3.csv("aaa.csv", function(d) {
// d.frequency = +d.frequency;
// d.output = +d.output;
// return d;
//},
// function(error, data) {
// if (error) throw error;

d3.json("php/GroupDelay.php", function(error, data) {
data.forEach(function(d) {
d.frequency = +d.frequencyGD;
d.output = +d.GroupDelay;
});
if (error) throw error;

// Scale the range of the data
x.domain(d3.extent(data, function(d) { return d.frequency; })).nice();
y.domain(d3.extent(data, function(d) { return d.output; }));

// Define the line
var line1 = d3.line()
.x(function(d) { return x(d.frequency); })
.y(function(d) { return y(d.output); });

// Define the axes
var xAxis = d3.axisBottom(x)
```

```
.scale(x);
var yAxis = d3.axisLeft(y)
.scale(y)
.tickFormat(d3.format(".1e"));
// Add the X Axis
g2.append("g")
.attr("transform", "translate(0," + height + ")")
.call(xAxis)
.append("text")
.attr("fill", "#e5e5e5")
.attr("y", 27)
.attr("dx", "0.91em")
.attr("text-anchor", "end")
.text("Frequency");
// Add the Y Axis
g2.append("g")
.call(yAxis)
.append("text")
.attr("fill", "#e5e5e5")
.attr("transform", "rotate(-90)")
.attr("y", 17)
.attr("dy", "100em")
.attr("text-anchor", "end")
.text("Delay Variance Indication");
// Add the valueline path.
g2.append("path")
.datum(data)
.attr("fill", "none")
.attr("stroke", "aquamarine")
.attr("stroke-linejoin", "round")
.attr("stroke-linecap", "round")
.attr("stroke-width", 4)
.attr("d", line1);

// gridlines in x axis function
function make_x_gridlines() {
return d3.axisBottom(x)
.scale(x)
.ticks(10)
}
// gridlines in y axis function
function make_y_gridlines() {
return d3.axisLeft(y)
.scale(y)
.ticks(20)
}
// add the X gridlines
```

```
svg2.append("g")
.attr("class", "grid")
.attr("transform", "translate(55," + 370 + ")")
.call(make_x_gridlines()
.tickSize(-500)
.tickFormat("")
)
// add the Y gridlines
svg2.append("g")
.attr("class", "grid")
.attr("transform", "translate(" + 55 + "+5)" )
.call(make_y_gridlines()
.tickSize(-1000)
.tickFormat("")
)
// Add the scatterplot
svg2.selectAll("dot")
.data(data)
.enter().append("circle")
.attr("r", 2)
.attr("cx", function(d) { return x(d.frequency)+55; })
.attr("cy", function(d) { return y(d.output)+5; })
.on("mouseover", function(d) {
div.transition()
.duration(200)
.style("opacity", .9);
div .html(d.frequency + "<br/>" + d.output)
.style("left", (d3.event.pageX) + "px")
.style("top", (d3.event.pageY ) + "px");
})
.on("mouseout", function(d) {
div.transition()
.duration(400)
.style("opacity", 0);
});


});
</script>
</center>
<hr/>
</body>
</html>
```

ShowGraphsUser.php

```
<!DOCTYPE html>
<html lang="en">
```

```
<head>
<link rel="stylesheet" type="text/css" href="style.css">
<font color="white">
<div id="nav-bar">
BITE for Satellite Transponder Calibration -  Measurements UI
<button class="btn btn-default pull-right nav-btn"
onclick="window.location.href='login.php'"><span class="glyphicon glyphicon-arrow-
left"></span> Back to login</button>
</div>
</font>
<!--- TAB ------>
<head>
<title>Satellite Transponder Measurements</title>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css">
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>
<script
src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>
</head>
<div class="container">
<br>
</div>
<!--- END of TAB -------------------------->
<body>
<! ------------------------------CHART -------------------------------------------
---------->
<center>
<style>
path.domain {
stroke: aquamarine;
}
.tick text {
fill:white
}
body {
background-color: #262f39;
}
.axis path,
.axis line{
fill: none;
stroke: black;
}
.line{
fill: none;
stroke: black;
```

```
stroke-width: 2px;
}
.tick text{
font-size: 12px;
}
.tick line{
opacity: 0.5;
}
.grid.tick {
stroke: black;
opacity: 0.7;
}
.grid line {
stroke: aquamarine;
stroke-opacity: 0.6;
shape-rendering: crispEdges;
}
.grid path {
stroke-width: 0;
}
circle {
fill: yellow;
}
path{
stroke: blue;
}
div.tooltip {
position: absolute;
text-align: center;
width: 100px;
height: 30px;
padding: 2px;
font: 12px sans-serif;
background: lightsteelblue;
border: 0px;
border-radius: 8px;
pointer-events: none;
}
</style>
<svg id="svg1" width="1100" height="400"></svg>
<svg id="svg2" width="1100" height="400"></svg>
<script src="https://d3js.org/d3.v4.min.js"></script>
<script>
// set the dimensions and margins of the graph
var svg1 = d3.select("#svg1")
```

```
margin = {top: 5, right: 0, bottom: 30, left: 55},
width = 1000- margin.left - margin.right,
height = 400 - margin.top - margin.bottom,
padding = 55;
g1 = svg1.append("g").attr("transform", "translate(" + margin.left + "," +
margin.top + ")");
// Set the ranges
var x = d3.scaleLinear()
.domain([13000, 14000])
.range([0, 1000])
.nice(10);

var y = d3.scaleLinear()
.rangeRound([height, 0]);
// Define the line
var line = d3.line()
.x(function(d) { return x(d.frequency); })
.y(function(d) { return y(d.output); });
// Define the div for the tooltip
var div = d3.select("body").append("div")
.attr("class", "tooltip")
.style("opacity", 0);
// Define the axes
var xAxis = d3.axisBottom(x)
.scale(x);

var yAxis = d3.axisLeft(y)
.scale(y);

// Define the div for the tooltip
//var div = d3.select("body").append("div")
// .attr("class", "tooltip")
// .style("opacity", 0);
// Get the data
//d3.csv("OUTPUT1.csv", function(d) {
// d.frequency = +d.frequency;
// d.output = +d.output;
// return d;
//},
// function(error, data) {
// if (error) throw error;
// Get the data
d3.json("php/Average.php", function(error, data) {
data.forEach(function(d) {
d.frequency = +d.frequencyNew;
d.output = +d.average;
});
if (error) throw error;
// Scale the range of the data
```

```
x.domain(d3.extent(data, function(d) { return d.frequency; })).nice();
y.domain(d3.extent(data, function(d) { return d.output; }));
// Add the X Axis
g1.append("g")
.attr("transform", "translate(0," + height + ")")
.call(xAxis)
.append("text")
.attr("fill", "#e5e5e5")
.attr("y", 27)
.attr("dx", "0.91em")
.attr("text-anchor", "end")
.text("Frequency");
// Add the Y Axis
g1.append("g")
.call(yAxis)
.append("text")
.attr("fill", "#e5e5e5")
.attr("transform", "rotate(-90)")
.attr("y", 17)
.attr("dy", "100em")
.attr("text-anchor", "end")
.text("Average Values");
// Add the valueline path.
g1.append("path")
.datum(data)
.attr("fill", "none")
.attr("stroke", "aquamarine")
.attr("stroke-linejoin", "round")
.attr("stroke-linecap", "round")
.attr("stroke-width", 4)
.attr("d", line)
;

// gridlines in x axis function
function make_x_gridlines() {
return d3.axisBottom(x)
.scale(x)
.ticks(10)
}
// gridlines in y axis function
function make_y_gridlines() {
return d3.axisLeft(y)
.scale(y)
.ticks(10)
}
// add the X gridlines
svg1.append("g")
```

```
.attr("class", "grid")
.attr("transform", "translate(55," + height + ")")
.call(make_x_gridlines()
.tickSize(-400)
.tickFormat("")
)
// add the Y gridlines
svg1.append("g")
.attr("class", "grid")
.attr("transform", "translate(" + 55 + "+5)" )
.call(make_y_gridlines()
.tickSize(-1000)
.tickFormat("")
)
// Add the scatterplot
svg1.selectAll("dot")
.data(data)
.enter().append("circle")
.attr("r", 2)
.attr("cx", function(d) { return x(d.frequency)+55; })
.attr("cy", function(d) { return y(d.output)+5; })
.on("mouseover", function(d) {
div.transition()
.duration(200)
.style("opacity", .9);
div .html(d.frequency + "<br/>" + d.output)
.style("left", (d3.event.pageX) + "px")
.style("top", (d3.event.pageY ) + "px");
})
.on("mouseout", function(d) {
div.transition()
.duration(400)
.style("opacity", 0);
});
});
////////////////////////////////////////////////////////SECOND
GRAPH/////////////////////////////////////////////////////////
// set the dimensions and margins of the second graph
var svg2 = d3.select("#svg2"),
margin = {top: 5, right: 0, bottom: 30, left: 55},
width = 100- margin.left - margin.right,
height = 400 - margin.top - margin.bottom,
padding = 40;
g2 = svg2.append("g").attr("transform", "translate(" + margin.left + "," +
margin.top + ")");
// Get the data
//d3.csv("aaa.csv", function(d) {
```

```
// d.frequency = +d.frequency;
// d.output = +d.output;
// return d;
//},
// function(error, data) {
// if (error) throw error;
d3.json("php/GroupDelay.php", function(error, data) {
data.forEach(function(d) {
d.frequency = +d.frequencyGD;
d.output = +d.GroupDelay;
});
if (error) throw error;
// Scale the range of the data
x.domain(d3.extent(data, function(d) { return d.frequency; })).nice();
y.domain(d3.extent(data, function(d) { return d.output; }));
// Define the line
var line1 = d3.line()
.x(function(d) { return x(d.frequency); })
.y(function(d) { return y(d.output); });
// Define the axes
var xAxis = d3.axisBottom(x)
.scale(x);
var yAxis = d3.axisLeft(y)
.scale(y)
.tickFormat(d3.format(".1e"));
// Add the X Axis
g2.append("g")
.attr("transform", "translate(0," + height + ")")
.call(xAxis)
.append("text")
.attr("fill", "#e5e5e5")
.attr("y", 27)
.attr("dx", "0.91em")
.attr("text-anchor", "end")
.text("Frequency");
// Add the Y Axis
g2.append("g")
.call(yAxis)
.append("text")
.attr("fill", "#e5e5e5")
.attr("transform", "rotate(-90)")
.attr("y", 17)
.attr("dy", "100em")
.attr("text-anchor", "end")
.text("Delay Variance Indication");
// Add the valueline path.
```

```
g2.append("path")
.datum(data)
.attr("fill", "none")
.attr("stroke", "aquamarine")
.attr("stroke-linejoin", "round")
.attr("stroke-linecap", "round")
.attr("stroke-width", 4)
.attr("d", line1);

// gridlines in x axis function
function make_x_gridlines() {
return d3.axisBottom(x)
.scale(x)
.ticks(10)
}
// gridlines in y axis function
function make_y_gridlines() {
return d3.axisLeft(y)
.scale(y)
.ticks(20)
}
// add the X gridlines
svg2.append("g")
.attr("class", "grid")
.attr("transform", "translate(55," + 370 + ")")
.call(make_x_gridlines()
.tickSize(-500)
.tickFormat("")
)
// add the Y gridlines
svg2.append("g")
.attr("class", "grid")
.attr("transform", "translate(" + 55 + "+5)" )
.call(make_y_gridlines()
.tickSize(-1000)
.tickFormat("")
)
// Add the scatterplot
svg2.selectAll("dot")
.data(data)
.enter().append("circle")
.attr("r", 2)
.attr("cx", function(d) { return x(d.frequency)+55; })
.attr("cy", function(d) { return y(d.output)+5; })
.on("mouseover", function(d) {
div.transition()
.duration(200)
.style("opacity", .9);
```

```
div .html(d.frequency + "<br/>" + d.output)
.style("left", (d3.event.pageX) + "px")
.style("top", (d3.event.pageY ) + "px");
})
.on("mouseout", function(d) {
div.transition()
.duration(400)
.style("opacity", 0);
});


});
</script>
</center>
<hr/>
</body>
</html>
```

dashboard.php

```
<!DOCTYPE HTML>
<html>
<head>
<title>Satellite Transponder Measurements</title>
<script
src="https://code.jquery.com/jquery-3.3.1.min.js"
integrity="sha256-FgpCb/KJQlLNfOu91ta32o/NMZxltwRo8QtmkMRdAu8="
crossorigin="anonymous">
</script>
<link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css">
<script src='http://91.138.190.103:8080/BeagleSite_v3/jim_script.js
'></script>
<script src='http://91.138.190.103:8080/BeagleSite_v3/real_time_graph.js
'></script>
<link rel="stylesheet" type="text/css"
href='http://91.138.190.103:8080/BeagleSite_v3/jim_style.css '>
</head>
<body onunload="wsDisconnect();">
<font color="white">
<div id="nav-bar">
BITE for Satellite Transponder Calibration -  Measurements UI
<button class="btn btn-default pull-right nav-btn"
onclick="window.location.href='login.php?logout=&quot;1&quot;'"><span
class="glyphicon glyphicon-log-out"></span> Logout</button>
<button class="btn btn-default pull-right nav-btn"
onclick="window.location.href='ShowGraphs.php'"><span class="glyphicon
glyphicon-signal"></span> Graphs</button>
</div>
```

```
<div class="container-fluid">
<div class="row">
<div class="col-lg-10 col-lg-offset-1" style="margin-top: 48px;">
<img src=http://91.138.190.103:8080/BeagleSite_v3/ilogo.png ' id="logo" />
<h1 style="margin: 0px;">Welcome</h1>
<hr/>
<span style="color: red;">Last Experiment's Result: The last experiment was
successful. </span>
<hr/>
<div class="row">
<div class="col-lg-6">
<h5>Status window</h5>
<div id="messages" class="msg-class">No connection</div>
<h5>Commands</h5>
<div class="msg-class">
<form method="post" action="" id="freqin" name="myform">
<button type="button" disabled name="submitted_start" class="button button1
btn btn-primary btn-lg" onclick='startEXP($(this).parent());'>Start</button>
<button type="button" disabled name="submitted_stop" class="button button1
btn btn-danger btn-lg" onclick='stopEXP();'>Stop</button>
<button type="button" disabled name="submitted_restart" class="button button1
btn btn-info btn-lg" onclick='restartCU();'><span class="glyphicon glyphicon-
repeat"></span></button>
<button type="button" name="submitted_refresh" class="button button1 btn btn-
warning btn-lg" onclick='refreshAll();'>Refresh status/LED's</button>
<div class="form-group row">
<div class="col-xs-4 ">
<label for="ex1"><font color="white">Start Frequency (MHz)</label>
<input disabled class="form-control" id="frstart" type="number" step="5"
min="13000" max="14000" required name="frstart" value="13325" />
</div>

<div class="col-xs-4 ">
<label for="ex2">Stop Frequency (MHz)</label>
<input disabled class="form-control" id="frstop" type="number" step="5"
min="13000" max="14000" required name="frstop" value="13775"/>
</div>

<div class="col-xs-4 ">
<label for="ex2">Frequency Step (MHz)</label>
<input disabled class="form-control" id="step" type="number" step="0.1"
min="0.1" required name="step" value="1"/>
</div>
</div>
</form>
</div>
</div>
<div class="col-lg-6" align="center">
<h5 style="margin-bottom: 2%;">Real-Time System Status</h5>
<div class="led-container">
<div class="leds">POWER</div>
<div class="leds">RF</div>
```

```
<div class="leds">TS</div>
<div class="leds">LD</div>
</div>
<h5 style="margin-top: 4%;margin-bottom: 2%;">Transponder Activation</h5>
<div class="progress progress-striped progress-striped">
<div class="progress-bar" role="progressbar" style="width: 100%" aria-
valuenow="25" aria-valuemin="0" aria-valuemax="100">
OFF
</div>
</div>
</div>
</div>
<hr/>
<h5>Progress</h5>
<div id="progress" class="msg-class full-width" style="margin-bottom: 2vh;">
<span style="display: flex;">
<img id="rot_iconn" style="width: 5vh; height: 5vh; display: none;"
src="loading.gif" alt="Loading Spinner">
 
<div id="rot_per">0%</div>
</span>
<div class="container" style="margin-top: 1.5vh;">
<div class="row">
<div class="col-md-2">
<div class="col-md-12 bar-not-loading"></div>
</div>
<div class="col-md-2">
<div class="col-md-12 bar-not-loading"></div>
</div>
<div class="col-md-2">
<div class="col-md-12 bar-not-loading"></div>
</div>
<div class="col-md-2">
<div class="col-md-12 bar-not-loading"></div>
</div>
<div class="col-md-2">
<div class="col-md-12 bar-not-loading"></div>
</div>
<div class="col-md-2">
<div class="col-md-12 bar-not-loading"></div>
</div>
</div>
</div>
</div>
<! ------------------------------CHART --------------------------------------
-------------- -->
<center>
<style>
path.domain {
```

```
stroke: aquamarine;
}
.tick text {
fill:white
}
body {
background-color: #262f39;
}
.axis path,
.axis line{
fill: none;
stroke: black;
}
.line{
fill: none;
stroke: black;
stroke-width: 2px;
}
.tick text{
font-size: 12px;
}
.tick line{
opacity: 0.5;
}
.grid.tick {
stroke: black;
opacity: 0.7;
}
.grid line {
stroke: aquamarine;
stroke-opacity: 0.6;
shape-rendering: crispEdges;
}
.grid path {
stroke-width: 0;
}
circle {
fill: yellow;
}
path{
stroke: blue;
}
div.tooltip {
position: absolute;
text-align: center;
```

```
width: 100px;
height: 30px;
padding: 2px;
font: 12px sans-serif;
background: lightsteelblue;
border: 0px;
border-radius: 8px;
pointer-events: none;
}
</style>
<svg id="svg1" width="1100" height="400"></svg>
<svg id="svg2" width="1100" height="400"></svg>
<script src="https://d3js.org/d3.v4.min.js"></script>
<script>
plot_now();
</script>
</center>
</div>
</div>
</font>
</body>
</html>
```

```
errors.php
<?php  if (count($errors) > 0) : ?>
<div class="error">
<?php foreach ($errors as $error) : ?>
<p><?php echo $error ?></p>
<?php endforeach ?>
</div>
<?php  endif ?>
```

jim_script.js

```
/**
        jim_script.js
**/

/*-----------------------------*/
/* Code executed on page load */
/* Setting up code variables  */
/*-----------------------------*/
var ws;
var wsUri = "ws:";
var loc = window.location;
console.log(loc);
if (loc.protocol === "https:") { wsUri = "wss:"; }
// This needs to point to the web sockets in the Node-RED flow
// ... in this case it's ws/simple, ws/status and ws/leds
var n = loc.host.indexOf(":");
wsUri += "//" + loc.host.substr(0,n+1) + "1880/ws/simple";
statusUri = "http://" + loc.host.substr(0,n+1) + "1880/status";
ledsUri = "http://" + loc.host.substr(0,n+1) + "1880/leds";
// 1.0 / 6 bars = 0.1666.... per bar
var incr = 0.1666;
```

```
// The current state
var APP_STATE = 0;
// Buffer for real-time plots.
var buffer = {"fstart":0 , "fstop": 0, "data":[], "data2":[]};

/*----------------------------*/
/* Connect to Web Service.    */
/*----------------------------*/
function wsConnect() {
    console.log("connect",wsUri);
    ws = new WebSocket(wsUri);
    ws.onmessage = function(msg) {
                var incoming = JSON.parse(msg.data);
                if (incoming.state > 0){
                changeState(incoming.state);
        }
        else{
                if (APP_STATE == 3){
                        // ..if the experiment is running
                 if (incoming.percentage ==-1){

updateProgress(incoming.percentage,incoming.new_data,null);
                 }
                 else{

updateProgress(incoming.percentage,null,incoming.new_data);
                 }
                }
        }
    }
    ws.onopen = function() {
        console.log("connected");
    }
    ws.onclose = function() {
        // Nothing, as per Stack Overflow
    }
}

/*-------------------------------*/
/* Disconnect from Web Service. */
/*-------------------------------*/
function wsDisconnect() {
    if (ws){
        ws.close();
    }
}

/*----------------------------------*/
/* FSM client-side implementation. */
/*----------------------------------*/
function changeState(state){
        switch (state){
        case 1:
             document.getElementById('messages').innerHTML = "Powered up!";
                document.getElementsByName('submitted_start')[0].disabled
= true;

        document.getElementsByName('submitted_restart')[0].disabled =
true;
                document.getElementsByName('submitted_stop')[0].disabled =
true;
```

```
            document.getElementsByName('frstart')[0].disabled = true;
            document.getElementsByName('frstop')[0].disabled = true;
            document.getElementsByName('step')[0].disabled = true;
            document.getElementById('rot_per').innerHTML = "0%";
            document.getElementById('rot_iconn').style.display = "none";
            $(".col-md-12").removeClass("bar-loading");
            $(".col-md-12").addClass("bar-not-loading");
               break;

        case 2:
            document.getElementById('messages').innerHTML = "Device is
ready and responsive.";
               document.getElementsByName('submitted_start')[0].disabled
= false;

        document.getElementsByName('submitted_restart')[0].disabled =
true;
               document.getElementsByName('submitted_stop')[0].disabled =
true;
            document.getElementsByName('frstart')[0].disabled = false;
            document.getElementsByName('frstop')[0].disabled = false;
            document.getElementsByName('step')[0].disabled = false;
            document.getElementById('rot_per').innerHTML = "0%";
            document.getElementById('rot_iconn').style.display = "none";
            $(".col-md-12").removeClass("bar-loading");
            $(".col-md-12").addClass("bar-not-loading");
               break;

        case 3:
            document.getElementById('messages').innerHTML = "Measurement
mode has started.";
               document.getElementsByName('submitted_start')[0].disabled
= true;

        document.getElementsByName('submitted_restart')[0].disabled =
true;
               document.getElementsByName('submitted_stop')[0].disabled =
false;
            document.getElementsByName('frstart')[0].disabled = true;
            document.getElementsByName('frstop')[0].disabled = true;
            document.getElementsByName('step')[0].disabled = true;
            document.getElementById('rot_per').innerHTML = "0%";
            document.getElementById('rot_iconn').style.display = "block";
            $(".col-md-12").removeClass("bar-loading");
            $(".col-md-12").addClass("bar-not-loading");
               break;

        case 4:
            document.getElementById('messages').innerHTML = "Measurements
have been aqcuired.";
               document.getElementsByName('submitted_start')[0].disabled
= true;

        document.getElementsByName('submitted_restart')[0].disabled =
true;
               document.getElementsByName('submitted_stop')[0].disabled =
true;
            document.getElementsByName('frstart')[0].disabled = true;
            document.getElementsByName('frstop')[0].disabled = true;
            document.getElementsByName('step')[0].disabled = true;
            document.getElementById('rot_per').innerHTML = "100%";
```

```
            document.getElementById('rot_iconn').style.display = "none";
            $(".col-md-12").removeClass("bar-not-loading");
            $(".col-md-12").addClass("bar-loading");
            buffer.data = [];
            buffer.data2 = [];
               break;

        case 5:
            document.getElementById('messages').innerHTML = "Averages
calculation is completed.";
               document.getElementsByName('submitted_start')[0].disabled
= true;

        document.getElementsByName('submitted_restart')[0].disabled =
true;
               document.getElementsByName('submitted_stop')[0].disabled =
true;
            document.getElementsByName('frstart')[0].disabled = true;
            document.getElementsByName('frstop')[0].disabled = true;
            document.getElementsByName('step')[0].disabled = true;
            document.getElementById('rot_per').innerHTML = "100%";
            document.getElementById('rot_iconn').style.display = "none";
            $(".col-md-12").removeClass("bar-not-loading");
            $(".col-md-12").addClass("bar-loading");
            buffer.data = [];
            buffer.data2 = [];
               break;

        case 6:
            document.getElementById('messages').innerHTML = "Experiment
complete!";
               document.getElementsByName('submitted_start')[0].disabled
= true;

        document.getElementsByName('submitted_restart')[0].disabled =
false;
               document.getElementsByName('submitted_stop')[0].disabled =
true;
            document.getElementsByName('frstart')[0].disabled = true;
            document.getElementsByName('frstop')[0].disabled = true;
            document.getElementsByName('step')[0].disabled = true;
            document.getElementById('rot_per').innerHTML = "100%";
            document.getElementById('rot_iconn').style.display = "none";
            $(".col-md-12").removeClass("bar-not-loading");
            $(".col-md-12").addClass("bar-loading");
            buffer.data = [];
            buffer.data2 = [];
               break;

        case 7:
            document.getElementById('messages').innerHTML = "Delay
Variance has been calculated.";
               document.getElementsByName('submitted_start')[0].disabled
= true;

        document.getElementsByName('submitted_restart')[0].disabled =
true;
               document.getElementsByName('submitted_stop')[0].disabled =
true;
            document.getElementsByName('frstart')[0].disabled = true;
            document.getElementsByName('frstop')[0].disabled = true;
```

```
            document.getElementsByName('step')[0].disabled = true;
            document.getElementById('rot_per').innerHTML = "100%";
            document.getElementById('rot_iconn').style.display = "none";
            $(".col-md-12").removeClass("bar-not-loading");
            $(".col-md-12").addClass("bar-loading");
            buffer.data = [];
            buffer.data2 = [];
               break;

               case 8:
            document.getElementById('messages').innerHTML = "Synthesizer
Command Rejected!";
            document.getElementsByName('submitted_start')[0].disabled =
true;
            document.getElementsByName('submitted_restart')[0].disabled =
false;
            document.getElementsByName('submitted_stop')[0].disabled =
true;
            document.getElementsByName('frstart')[0].disabled = true;
            document.getElementsByName('frstop')[0].disabled = true;
            document.getElementsByName('step')[0].disabled = true;
            document.getElementById('rot_per').innerHTML = "100%";
            document.getElementById('rot_iconn').style.display = "none";
            $(".col-md-12").removeClass("bar-not-loading");
            $(".col-md-12").addClass("bar-loading");
            buffer.data = [];
            buffer.data2 = [];
            break;

    }
    APP_STATE = state;
}

/*--------------------------*/
/* Send START message back  */
/* to the Web Service.      */
/*--------------------------*/
function startEXP(m) {
    // Validate input
    var forma = document.querySelector('[name="myform"]');
    if (!forma.checkValidity()) {
        // Form is invalid!
        // Create the temporary button, click and remove it.
        const tmpSubmit = document.createElement('button');
        forma.appendChild(tmpSubmit);
        tmpSubmit.click();
        forma.removeChild(tmpSubmit);
        return;
    }
        // If connection open...
    if (ws) {
        buffer.fstart = m.find('[name=frstart]').val();
        buffer.fstop = m.find('[name=frstop]').val();
        var data = JSON.stringify({
            "mode": "START",
            "fstart": buffer.fstart,
            "fstop": buffer.fstop,
            "fstep": m.find('[name=step]').val()
        });
        ws.send(data);
        jQuery.ajax({
```

```
            type: "POST",
            url: 'php/StoredValues.php',
            data: {fstart: buffer.fstart, fstop: buffer.fstop}
        });
    }
}

/*--------------------------*/
/* Send STOP message back   */
/* to the Web Service.      */
/*--------------------------*/
function stopEXP() {
    // If connection open...
    if (ws) {
        var data = JSON.stringify({
            "mode": "STOP"
        });
        ws.send(data);
    }
}

/*---------------------------*/
/* Calculation of the number */
/* of active progress bars   */
/* based on the given per.   */
/* INPUT RANGE: 0.00 - 1.00  */
/*---------------------------*/
function updateProgress(per, data, dataGD) {
    if (per >= 0){
        var count = Math.floor(per/incr);
        $(".col-md-12").slice(0,count).removeClass("bar-not-loading");
        $(".col-md-12").slice(0,count).addClass("bar-loading");
        document.getElementById('rot_per').innerHTML = Math.floor(per*100)
+ "%";
    }
    else{
        d3.select("#svg1").selectAll("*").remove();
        d3.select("#svg2").selectAll("*").remove();
        if (data != null){
            // Plot 1
            buffer.data = data.reduce( function(coll,item){
                coll.push( item );
                return coll;
            }, buffer.data );
        }
        if (dataGD != null){
            // Plot 2
            buffer.data2 = dataGD.reduce( function(coll,item){
                coll.push( item );
                return coll;
            }, buffer.data2 );
        }
        plot_now(buffer.data, buffer.data2, buffer.fstart, buffer.fstop);
    }
}

/*--------------------------------*/
/* AJAX request of the status of */
/* the server.                    */
/*--------------------------------*/
function requestStatus() {
```

```
    var form_data = "Random data";
    var form_url = statusUri;
    var form_method = "GET";

    // Check to see if the flag has been initialized
    if ( typeof requestStatus.connected == 'undefined' ) {
        // It has not... perform the initialization
        requestStatus.connected = false;
    }

    $.ajax({
        url: form_url,
        type: form_method,
        data: form_data,
        cache: false,
        success: function(return_obj){
            if (return_obj.msg.includes('READY')){
                clear_inter();
                if (requestStatus.connected == false){
                    wsConnect();
                    requestStatus.connected = true;
                }
                changeState(return_obj.state);
            }
        },
        timeout: 2000
    });
}

/*--------------------------------*/
/* Every 4s request the status   */
/* of the server until it is up. */
/*--------------------------------*/
var result_interval = setInterval(requestStatus, 4000);

/*--------------------------------*/
/* Clear the repeating requests  */
/* by disabling the interval.    */
/*--------------------------------*/
function clear_inter(){
    clearInterval(result_interval);
}

/*--------------------------------*/
/* AJAX request of the status of */
/* the GPIO pins and the LED's.  */
/*--------------------------------*/
function request_LED() {
    var form_data = "Random data";
    var form_url = ledsUri;
    var form_method = "GET";

    $.ajax({
        url: form_url,
        type: form_method,
        data: form_data,
        cache: false,
        success: function(return_obj){
            if (return_obj.led1){
                $(".leds")[0].style.background = "#6666ff";
            }
```

```
            else{
                $(".leds")[0].style.background = "grey";
            }
            if (return_obj.led2){
                $(".leds")[1].style.background = "#6666ff";
            }
            else{
                $(".leds")[1].style.background = "grey";
            }
            if (return_obj.led3){
                $(".leds")[2].style.background = "#6666ff";
            }
            else{
                $(".leds")[2].style.background = "grey";
            }
            if (return_obj.led4){
                $(".leds")[3].style.background = "red";
            }
            else{
                $(".leds")[3].style.background = "grey";
            }
            if (return_obj.led5){
                $(".progress-bar")[0].style.backgroundColor = "#dc3545";
                $(".progress-bar")[0].innerHTML = "ON";
                $(".progress").addClass("active");
            }
            else{
                $(".progress-bar")[0].style.backgroundColor = "grey";
                $(".progress-bar")[0].innerHTML = "OFF";
                $(".progress").removeClass("active");
            }
        }
    });
}

/*-------------------------------*/
/* Every 4s request the status   */
/* of the LED's non-stop.        */
/*-------------------------------*/
setInterval(request_LED, 4000);

/*-----------------------------------------*/
/* Immediate request for the current status */
/* of the server and the LED's / GPIO pins. */
/*-----------------------------------------*/
function refreshAll(){
    requestStatus();
    request_LED();
}

/*--------------------------*/
/* Send RESTART message     */
/* back to the Web Service. */
/*--------------------------*/
function restartCU(){
        // If connection open...
    if (ws) {
        var data = JSON.stringify({
            "mode": "RESTART"
        });
        ws.send(data);
```

```
    }
}
```

jim_style.css

```
/* CSS style*/
table {
  font-family: arial, sans-serif;
  border-collapse: collapse;
  width: 50%;
  text-align: center;
}

td, th {
  border: 1px solid black;
  text-align: center;
  color: red;
  padding: 8px;
}

tr {
  background-color: #dddddd;
}

.topright {
  position: absolute;
  top: 40px;
  right: 100px;
  font-size: 18px;
}

.button {
  border: none;
  color: white;
  padding: 15px 32px;
  text-align: center;
  text-decoration: none;
  display: inline-block;
  font-size: 16px;
  margin: 4px 2px;
  cursor: pointer;
}

body {
  background-color: #262f39;
}

.wrapper {
  text-align: center;
}

.error {
  color: #FF0000;
}

h5 {
  margin: 3px;
}

.msg-class{
  border-width: 1px;
  border-style: solid;
  max-width: 500px;
```

```
  padding: 10px;
  border-radius: 5px;
}

.leds{
  border-width: 2px;
  border-radius: 50px;
  border-style: solid;
  width: 80px;
  height: 80px;
  background: grey;
  line-height: 80px;
  text-align: center;
  margin-right: 1%;
  font-weight: bold;
}

.leds:not(:first-child){
  width: 60px;
  height: 60px;
  line-height: 60px;
}

#nav-bar {
    background-color: #dddddd;
    color: #555;
    border-bottom-right-radius: 5px;
    border-bottom-left-radius: 5px;
    padding: 10px;
    margin-left: 2%;
    margin-right: 2%;
    font-weight: bold;
    font-size: 18px;
    padding-bottom: 1.6%;
    text-shadow: 4px 2px 6px #b3a3a3;
}

h1 {
  text-align: center;
}

.nav-btn{
  margin-left: 1%;
}

.led-container{
  display: flex;
  align-items: center;
  justify-content: center;
}

.bar-loading{
  border-style: solid;
  border-width: 1px;
  padding: 5px;
  background-color: lime;
  border-radius: 8px;
}

.bar-not-loading{
  border-style: solid;
```

```
  border-width: 1px;
  padding: 5px;
  background-color: grey;
  border-radius: 8px;
}

hr{
  height: 2px;
  background-color: blueviolet;
  border: none;
}

.full-width{
  max-width: none;
}

#progress{
  display: flex;
  line-height: 5vh;
}

.gly-spin {
  padding-top: 4px;
  margin-top: -3px;
  margin-right: 20%;
  font-size: 18px;
  -animation: spin 2s infinite linear;
  -webkit-animation: spinWebkit 2s infinite linear;
  -moz-animation: spinMoz 2s infinite linear;
}

@-webkit-keyframes spinWebkit {
    from { -webkit-transform: rotate(0deg);}
    to { -webkit-transform: rotate(360deg);}
}

@keyframes spinMoz {
    from { transform: scale(1) rotate(0deg);}
    to { transform: scale(1) rotate(360deg);}
}

@keyframes spin {
    from {  rotate(0deg);}
    to { rotate(360deg);}
}

.glyphicon-signal{
  color: #5F9EA0;
}

.btn-info{
  padding-left: 2.5%;
  padding-right: 2.5%;
}

.button[disabled]{
  opacity: .15;
}

#logo{
  max-width: 12%;
```

```
  position: absolute;
}

.progress{
  max-width: 50%;
}

.progress .progress-bar{
  background-color: grey;
  font-weight: bold;
}
```

login.php

```
<!DOCTYPE html>
<html>
<head>
<title>Satellite Transponder Measurements</title>
<link rel="stylesheet" type="text/css" href="style.css">
<link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css">
</head>
<body>
<font color="white">
<div id="nav-bar">
BITE for Satellite Transponder Calibration -  Measurements UI
<button class="btn btn-default pull-right nav-btn"
onclick="window.location.href='ShowGraphsUser.php'"><span class="glyphicon
glyphicon-signal"></span> Graphs</button>
</div>
</font>
<style>
body {
background-color: #262f39;
/*background-color: white; */
}
h3 {
text-align: center;
}
</style>
<div class="header">
<h2>Login as Admin</h2>
</div>
<form method="post" action="login.php">
<div class="input-group">
<label>Username</label>
```

```
<input type="text" name="username" autofocus>
</div>
<div class="input-group">
<label>Password</label>
<input type="password" name="password">
</div>
<div class="input-group">
<button type="submit" class="btn" name="login_user">Login</button>
</div>
</form>
</body>
</html>
```

php/Average.php

```
<?php
$username = "beagleuser";
$password = "beagleuser";
$host = "localhost";
$database="beagledb";
$server = mysql_connect($host, $username, $password);
$connection = mysql_select_db($database, $server);
$myquery = "SELECT * FROM AverageBBB WHERE id=2";
$query = mysql_query($myquery);
if ( ! $query ) {
echo mysql_error();
die;
}
$data = array();
for ($x = 0; $x < mysql_num_rows($query); $x++) {
$data[] = mysql_fetch_assoc($query);
}
echo json_encode($data);
mysql_close($server);
?>
```

php/GroupDelay.php

```
<?php
$username = "beagleuser";
$password = "beagleuser";
$host = "localhost";
$database="beagledb";
$server = mysql_connect($host, $username, $password);
$connection = mysql_select_db($database, $server);
$myquery = "SELECT  frequencyGD, GroupDelay  FROM  GroupDelayBBB WHERE
id=2";
$query = mysql_query($myquery);
if ( ! $query ) {
echo mysql_error();
```

```
die;
}
$data = array();
for ($x = 0; $x < mysql_num_rows($query); $x++) {
$data[] = mysql_fetch_assoc($query);
}
echo json_encode($data);
mysql_close($server);
?>
```

php/StoredValues.php

```
<?php
if( isset($_POST['fstart']) && isset($_POST['fstop'])) {
$f1 = $_POST['fstart'];
$f2 = $_POST['fstop'];
$username = "beagleuser";
$password = "beagleuser";
$host = "localhost";
$database="beagledb";
$server = mysql_connect($host, $username, $password);
$connection = mysql_select_db($database, $server);
$myquery = "UPDATE `ExperimentResult` SET `id`=2,`message`='" . $f1 . "'
WHERE `id`=2;";
$query = mysql_query($myquery);
if ( ! $query ) {
echo mysql_error();
die;
}
$myquery = "UPDATE `ExperimentResult` SET `id`=3,`message`='" . $f2 . "'
WHERE `id`=3;";
$query = mysql_query($myquery);
if ( ! $query ) {
echo mysql_error();
die;}
mysql_close($server);
}
?>
```

real_time_graph.js

```
/*
  Real time graphs - JS
  ----------------------
*/

function plot_now(RTdata, GDdata, fsr, fst){

  // set the dimensions and margins of the graph
  var svg1 = d3.select("#svg1"),
      margin = {top: 5, right: 0, bottom: 30, left: 55},
      width = 1000- margin.left - margin.right,
      height = 400 - margin.top - margin.bottom,
      padding = 55;
  g1 = svg1.append("g").attr("transform", "translate(" + margin.left + ","
+ margin.top + ")");
```

```
  // Set the ranges
  var x = d3.scaleLinear()
          .domain([13000, 14000])
          .range([0, 1000])
          .nice(10);


  var y = d3.scaleLinear()
          .rangeRound([height, 0]);

  // Define the line
  var line = d3.line()
          .x(function(d) { return x(d.frequency); })
          .y(function(d) { return y(d.output); });

  // Define the div for the tooltip
  var div = d3.select("body").append("div")
          .attr("class", "tooltip")
          .style("opacity", 0);

  // Define the axes
  var xAxis = d3.axisBottom(x)
          .scale(x);


  var yAxis = d3.axisLeft(y)
          .scale(y);

  function consume_data(error, data, fstart, fstop) {
    data.forEach(function(d) {
      d.frequency = +d.frequencyNew;
      d.output = +d.average;
    });
    if (error) throw error;

    // Scale the range of the data
    if (fstart != null){
      x.domain([fstart,fstop]);
    }
    else{
      x.domain(d3.extent(data, function(d) { return d.frequency;
})).nice();
    }
    y.domain(d3.extent(data, function(d) { return d.output; }));

    // Add the X Axis
    g1.append("g")
        .attr("transform", "translate(0," + height + ")")
        .call(xAxis)
        .append("text")
        .attr("fill", "#e5e5e5")
        .attr("y", 27)
        .attr("dx", "0.91em")
        .attr("text-anchor", "end")
        .text("Frequency");

    // Add the Y Axis
    g1.append("g")
        .call(yAxis)
        .append("text")
```

```
    .attr("fill", "#e5e5e5")
    .attr("transform", "rotate(-90)")
    .attr("y", 17)
    .attr("dy", "100em")
    .attr("text-anchor", "end")
    .text("Average Values");

// Add the valueline path.
g1.append("path")
    .datum(data)
    .attr("fill", "none")
    .attr("stroke", "aquamarine")
    .attr("stroke-linejoin", "round")
    .attr("stroke-linecap", "round")
    .attr("stroke-width", 4)
    .attr("d", line);

// gridlines in x axis function
function make_x_gridlines() {
  return d3.axisBottom(x)
      .scale(x)
      .ticks(10)
}

// gridlines in y axis function
function make_y_gridlines() {
  return d3.axisLeft(y)
      .scale(y)
      .ticks(10)
}

// add the X gridlines
svg1.append("g")
   .attr("class", "grid")
   .attr("transform", "translate(55," + height + ")")
   .call(make_x_gridlines()
   .tickSize(-400)
   .tickFormat("")
)

// add the Y gridlines
svg1.append("g")
   .attr("class", "grid")
   .attr("transform", "translate(" + 55 + "+5)"       )
   .call(make_y_gridlines()
   .tickSize(-1000)
   .tickFormat("")
)

// Add the scatterplot
svg1.selectAll("dot")
    .data(data)
    .enter().append("circle")
    .attr("r", 2)
    .attr("cx", function(d) { return x(d.frequency)+55; })
    .attr("cy", function(d) { return y(d.output)+5; })
    .on("mouseover", function(d) {
      div.transition()
        .duration(200)
        .style("opacity", .9);
      div  .html(d.frequency + "<br/>"  + d.output)
```

```
              .style("left", (d3.event.pageX) + "px")
              .style("top", (d3.event.pageY ) + "px");
        })
        .on("mouseout", function(d) {
          div.transition()
            .duration(400)
            .style("opacity", 0);
        });
  }

  // Get the data
  if (RTdata == null){
    d3.json("php/Average.php", consume_data);
  }
  else{
    consume_data(false, RTdata, fsr, fst);
  }

  /////////////SECOND GRAPH//////////

  // set the dimensions and margins of the second graph
  var svg2 = d3.select("#svg2"),
    margin = {top: 5, right: 0, bottom: 30, left: 55},
    width = 100- margin.left - margin.right,
    height = 400 - margin.top - margin.bottom,
    padding = 40;
  g2 = svg2.append("g").attr("transform", "translate(" + margin.left + ","
+ margin.top + ")");

  function consume_data_GD(error, data, fstart, fstop) {
    data.forEach(function(d) {
      d.frequency = +d.frequencyGD;
      d.output = +d.GroupDelay;
    });
    if (error) throw error;

    // Scale the range of the data
    if (fstart != null){
      x.domain([fstart,fstop]);
    }
    else{
      x.domain(d3.extent(data, function(d) { return d.frequency;
})).nice();
    }
    y.domain(d3.extent(data, function(d) { return d.output; }));

    // Define the line
    var line1 = d3.line()
            .x(function(d) { return x(d.frequency); })
            .y(function(d) { return y(d.output); });

    // Define the axes
    var xAxis = d3.axisBottom(x)
            .scale(x);

    var yAxis = d3.axisLeft(y)
            .scale(y)
            .tickFormat(d3.format(".1e"));

    // Add the X Axis
    g2.append("g")
```

```
    .attr("transform", "translate(0," + height + ")")
    .call(xAxis)
    .append("text")
    .attr("fill", "#e5e5e5")
    .attr("y", 27)
    .attr("dx", "0.91em")
    .attr("text-anchor", "end")
          .text("Frequency");

// Add the Y Axis
g2.append("g")
    .call(yAxis)
    .append("text")
    .attr("fill", "#e5e5e5")
    .attr("transform", "rotate(-90)")
    .attr("y", 17)
    .attr("dy", "100em")
    .attr("text-anchor", "end")
    .text("Delay Variance Indication");

// Add the valueline path.
g2.append("path")
    .datum(data)
    .attr("fill", "none")
    .attr("stroke", "aquamarine")
    .attr("stroke-linejoin", "round")
    .attr("stroke-linecap", "round")
    .attr("stroke-width", 4)
    .attr("d", line1);

// gridlines in x axis function
function make_x_gridlines() {
  return d3.axisBottom(x)
      .scale(x)
      .ticks(10)
}

// gridlines in y axis function
function make_y_gridlines() {
  return d3.axisLeft(y)
      .scale(y)
      .ticks(20)
}

// add the X gridlines
svg2.append("g")
    .attr("class", "grid")
    .attr("transform", "translate(55," + 370 + ")")
    .call(make_x_gridlines()
    .tickSize(-500)
    .tickFormat(""))

// add the Y gridlines
svg2.append("g")
    .attr("class", "grid")
    .attr("transform", "translate(" + 55 + "+5)"        )
    .call(make_y_gridlines()
    .tickSize(-1000)
    .tickFormat(""))

// Add the scatterplot
```

```
    svg2.selectAll("dot")
        .data(data)
        .enter().append("circle")
        .attr("r", 2)
        .attr("cx", function(d) { return x(d.frequency)+55; })
        .attr("cy", function(d) { return y(d.output)+5; })
        .on("mouseover", function(d) {
          div.transition()
              .duration(200)
              .style("opacity", .9);
          div  .html(d.frequency + "<br/>"  + d.output)
              .style("left", (d3.event.pageX) + "px")
              .style("top", (d3.event.pageY ) + "px");
        })
        .on("mouseout", function(d) {
          div.transition()
              .duration(400)
              .style("opacity", 0);
        });
  }

  // Get the data
  if (GDdata == null){
    d3.json("php/GroupDelay.php", consume_data_GD);
  }
  else{
    consume_data_GD(false, GDdata, fsr, fst);
  }
}
```

register.php

```
<!DOCTYPE html>
<html>
<head>
<title>Registration system</title>
<link rel="stylesheet" type="text/css" href="style.css">
<link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css">
</head>
<body style="background-color: #262f39;">
<font color="white">
<div id="nav-bar">
BITE for Satellite Transponder Calibration -  Measurements UI
<button class="btn btn-default pull-right nav-btn"
onclick="window.location.href='ShowGraphsUser.php'"><span class="glyphicon
glyphicon-signal"></span> Graphs</button>
</div>
</font>
<div class="header">
<h2>Register</h2>
</div>
<form method="post" action="register.php">
```

```
<div class="input-group">
<label>Username</label>
<input type="text" name="username" value="">
</div>
<div class="input-group">
<label>Email</label>
<input type="email" name="email" value="">
</div>
<div class="input-group">
<label>Password</label>
<input type="password" name="password_1">
</div>
<div class="input-group">
<label>Confirm password</label>
<input type="password" name="password_2">
</div>
<div class="input-group">
<button type="submit" class="btn" name="reg_user">Register</button>
</div>
<p>
Already a member? <a href="login.php">Sign in</a>
</p>
<p>
<a href="ShowGraphsUser.php"><font color="">Continue to graphs</font></a>
</p>
</form>
<div style="height: 10vh;"></div>
</body>
</html>
```

server.php

```
<?php
session_start();
// variable declaration
$username = "";
$email    = "";
$errors = array();
$_SESSION['success'] = "";
// connect to database
$db = mysqli_connect('localhost', 'beagleuser', 'beagleuser',
'registration');
// REGISTER USER
if (isset($_POST['reg_user'])) {
// receive all input values from the form
$username = mysqli_real_escape_string($db, $_POST['username']);
$email = mysqli_real_escape_string($db, $_POST['email']);
$password_1 = mysqli_real_escape_string($db, $_POST['password_1']);
$password_2 = mysqli_real_escape_string($db, $_POST['password_2']);
// form validation: ensure that the form is correctly filled
if (empty($username)) { array_push($errors, "Username is required"); }
if (empty($email)) { array_push($errors, "Email is required"); }
```

```
if (empty($password_1)) { array_push($errors, "Password is required"); }
if ($password_1 != $password_2) {
array_push($errors, "The two passwords do not match");
}
// register user if there are no errors in the form
if (count($errors) == 0) {
$password = md5($password_1);//encrypt the password before saving in the
database $query = "INSERT INTO users (username, email, password)
VALUES('$username', '$email', '$password')";
mysqli_query($db, $query);
$_SESSION['username'] = $username;
$_SESSION['success'] = "You are now logged in";
header('location: dashboard.php');
}
}
// ...
// LOGIN USER
if (isset($_POST['login_user'])) {
$username = mysqli_real_escape_string($db, $_POST['username']);
$password = mysqli_real_escape_string($db, $_POST['password']);
if (empty($username)) {
array_push($errors, "Username is required");
}
if (empty($password)) {
array_push($errors, "Password is required");
}
if (count($errors) == 0) {
$password = md5($password);
$query = "SELECT * FROM users WHERE username='$username' AND
password='$password'";
$results = mysqli_query($db, $query);
if (mysqli_num_rows($results) == 1) {
$_SESSION['username'] = $username;
$_SESSION['success'] = "You are now logged in";
header('location: dashboard.php');
}else {
array_push($errors, "Wrong username/password combination");
}
}
}
?>
```

style.css

```
* {
	margin: 0px;
	padding: 0px;
}

body {
	font-size: 120%;
	background: #F8F8FF;
}

.header {
	width: 30%;
	margin: 50px auto 0px;
	color: white;
	background: #5F9EA0;
	text-align: center;
	border: 1px solid #B0C4DE;
```

```
        border-bottom: none;
        border-radius: 10px 10px 0px 0px;
        padding: 20px;
}

form, .content {
        width: 30%;
        margin: 0px auto;
        padding: 20px;
        border: 1px solid #B0C4DE;
        background: white;
        border-radius: 0px 0px 10px 10px;
}

.input-group {
        margin: 10px 0px 10px 0px;
        width: 100%;
}

.input-group label {
        display: block;
        text-align: left;
        margin: 3px;
}

.input-group input {
        height: 30px;
        width: 93%;
        padding: 5px 10px;
        font-size: 16px;
        border-radius: 5px;
        border: 1px solid gray;
}

.btn {
        padding: 10px;
        font-size: 15px;
        color: white;
        background: #5F9EA0;
        border: none;
        border-radius: 5px;
}

.error {
        width: 92%;
        padding: 10px;
        border: 1px solid #a94442;
        color: #a94442;
        background: #f2dede;
        border-radius: 5px;
        text-align: left;
}

.success {
        color: #3c763d;
        background: #dff0d8;
        border: 1px solid #3c763d;
        margin-bottom: 20px;
}

#nav-bar {
```

```
    background-color: #dddddd;
    color: #555;
    border-bottom-right-radius: 5px;
    border-bottom-left-radius: 5px;
    padding: 10px;
    margin-left: 2%;
    margin-right: 2%;
    font-weight: bold;
    font-size: 18px;
    padding-bottom: 1.6%;
    text-shadow: 4px 2px 6px #b3a3a3;
}

.nav-btn{
        margin-left: 1%;
}

.glyphicon-signal{
        color: #5F9EA0;
}
```

Κώδικας εξαγωγής τιμών από μετατροπέα ADC

```
#include <stdlib.h>
#include <stdio.h>
#include <string.h>
#include <unistd.h>     //close()
#include <fcntl.h>     //define O_WONLY and O_RDONLY
#define MAX_BUF 64     //This is plenty large
int Freq = 13000;
//Function declarations
int readADC(unsigned int pin);
void itoa(int n, char s[]);
void reverse(char s[]);
FILE *fp;
//main program
int main(int argc, char **argv)
{
int i;
//Enable ADC pins within code
//system("echo BB-ADC > /sys/devices/platform/bone_capemgr/slots");
char sMeas[5];
for (i = 0; i < 10; i++)
{
int adcx = readADC(0);
printf("%d", adcx);
```

```
if (i<9)
printf(",");
usleep(10000); //delay 10 milliseconds
}
//printf("\n");
return 0;
}//end main
//Function definitions
int readADC(unsigned int pin)
{
int fd
char buf[MAX_BUF];     //file buffer
char val[4];     //holds up to 4 digits for ADC value
//Create the file path by concatenating the ADC pin number to the end of the
string
//Stores the file path name string into "buf"
snprintf(buf, sizeof(buf), "/sys/bus/iio/devices/iio:device0/in_voltage%d_raw",
pin);     //Concatenate ADC file name
fd = open(buf, O_RDONLY);     //open ADC as read only
//Will trigger if the ADC is not enabled
if (fd < 0) {
perror("ADC - problem opening ADC");
}//end if
read(fd, &val, 4);     //read ADC ing val (up to 4 digits 0-1799)
close(fd);     //close file and stop reading
return atoi(val);     //returns an integer value (rather than ascii)
}//end read ADC()
/* itoa:  convert n to characters in s */
void itoa(int n, char s[])
{
int i, sign;
if ((sign = n) < 0)  /* record sign */
n = -n;          /* make n positive */
i = 0;
do {       /* generate digits in reverse order */
s[i++] = n % 10 + '0';   /* get next digit */
} while ((n /= 10) > 0);     /* delete it */
```

```
if (sign < 0)
s[i++] = '-';
s[i] = '\0';
reverse(s);
}
/* reverse:  reverse string s in place */
void reverse(char s[])
{
int i, j;
char c;
for (i = 0, j = strlen(s) - 1; i<j; i++, j--) {
c = s[i];
s[i] = s[j];
s[j] = c;
}
}
```