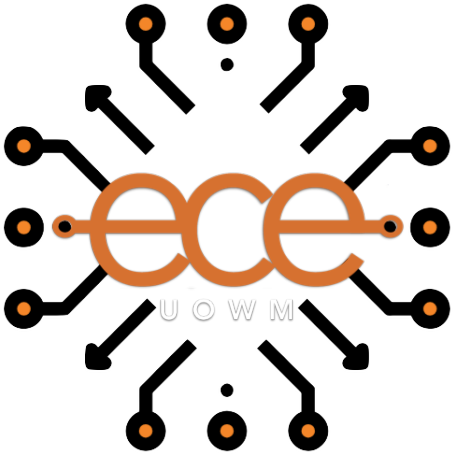


Πανεπιστήμιο Δυτικής Μακεδονίας

Πολυτεχνική Σχολή

Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

Μελέτη, ανάπτυξη και ενσωμάτωση ψηφιακού φωνητικού βοηθού στο έξυπνο σπίτι με χρήση opensource τεχνολογιών

Διπλωματική Εργασία

του

Κώστα Χατζόγλου

Επιβλέποντες:

Παντελής Αγγελίδης Καθηγητής

Νίκος Γκάλφας ΕΔΙΠ

Μάρτιος 2022,Κοζάνη

Ευχαριστίες

Με την περάτωση της παρούσας διπλωματικής εργασίας θα ήθελα να ευχαριστήσω θερμά τον επιβλέποντα καθηγητή μου κ.Αγγελίδη Παντελή,καθώς και τον κ.Γκάλφα Νικόλαο μέλος του ερευνητικού προσωπικού της σχολής.Τον πρώτο,για την εμπιστοσύνη που μου έδειξε στην εκπόνηση της παρούσας διπλωματικής εργασίας και τον δεύτερο για τις παραγωγικές υποδείξεις του,την παρότρυνσή και το πολύ καλό κλίμα συνεργασίας που διαμόρφωσε συμβάλλοντας τα μέγιστα για την κατάρτιση της διπλωματικής εργασίας.Επίσης θέλω να ευχαριστήσω την κ.Θωμαή Καραμήτσου για τις συμβουλές και υποδείξεις της στην συγγραφή του κειμένου. Ιδιαιτέρως ένα ευχαριστώ στην οικογένεια μου καθώς αποτελεί στήριγμα καθ'όλη τη διάρκεια της ζωής μου.

Δήλωση Πνευματικών Δικαιωμάτων

Δηλώνω ρητά ότι, σύμφωνα με το άρθρο 8 του Ν. 1599/1986 και τα άρθρα 2,4,6 παρ. 3 του Ν. 1256/1982, η παρούσα Διπλωματική Εργασία με τίτλο

”**Μελέτη, ανάπτυξη και ενσωμάτωση ψηφιακού φωνητικού βοηθού στο έξυπνο σπίτι με χρήση opensource τεχνολογιών** ”

καθώς και τα ηλεκτρονικά αρχεία και πηγαίοι κώδικες που αναπτύχθηκαν ή τροποποιήθηκαν στα πλαίσια αυτής της εργασίας και αναφέρονται ρητώς μέσα στο κείμενο που συνοδεύουν, και η οποία έχει εκπονηθεί στο Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών του Πανεπιστημίου Δυτικής Μακεδονίας, υπό την επίβλεψη του μέλους του Τμήματος κ. **Αγγελίδη Παντελή** αποτελεί αποκλειστικά προϊόν προσωπικής εργασίας και δεν προσβάλλει κάθε μορφής πνευματικά δικαιώματα τρίτων και δεν είναι προϊόν μερικής ή ολικής αντιγραφής, οι πηγές δε που χρησιμοποιήθηκαν περιορίζονται στις βιβλιογραφικές αναφορές και μόνον. Τα σημεία όπου έχω χρησιμοποιήσει ιδέες, κείμενο, αρχεία ή / και πηγές άλλων συγγραφέων, αναφέρονται ευδιάκριτα στο κείμενο με την κατάλληλη παραπομπή και η σχετική αναφορά περιλαμβάνεται στο τμήμα των βιβλιογραφικών αναφορών με πλήρη περιγραφή.

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς τον συγγραφέα. Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν τον συγγραφέα και μόνο.

Υπογραφή Φοιτητή:

Περίληψη

Σκοπός της εργασίας είναι η υλοποίηση ενός προσωπικού ψηφιακού φωνητικού βοηθού χρησιμοποιώντας open source εφαρμογές που μας δίνουν τη δυνατότητα χρήσης της ελληνικής γλώσσας ως μέσο αλληλεπίδρασης,χωρίς την ανάγκη σύνδεσης στο Internet ή κάποιας βάσης δεδομένων, προσφέροντας μια ενδεικτικά ασφαλέστερη,ανεξάρτητη και με περισσότερη παραμετροποίηση και προσαρμογή στον χρήστη λύση,σε σύγκριση με τις περισσότερες εμπορικές προτάσεις της αγοράς.

Στο πρώτο κεφάλαιο πραγματοποιήσαμε μια εισαγωγή στην λειτουργία του έξυπνου σπιτιού,κάνοντας αναφορά στις τεχνολογίες του Internet of Things. Αναλύσαμε τις λειτουργίες,τον τρόπο που δουλεύουν όσο και τις τεχνολογίες στις οποίες βασίζονται οι προσωπικοί φωνητικοί βοηθοί-voice assistants,ενώ καλύψαμε τις τρέχουσες εμπορικές λύσεις όσο και open source εφαρμογές. Κλείνοντας το πρώτο μέρος δίνουμε περισσότερη έμφαση στο λογισμικό ανοιχτού κώδικα που χρησιμοποιήσαμε στην δική μας υλοποίηση και παραθέτουμε τους λόγους για τους οποίους προβήκαμε σε αυτή την επιλογή. Στο δεύτερο μέρος επικεντρωθήκαμε στην υλοποίηση του δικού μας IoT δικτύου μέσω του προσωπικού ψηφιακού φωνητικού βοηθού. Αναλύσαμε το hardware και software που χρησιμοποιήσαμε και εξετάσαμε τον τρόπο όσο και τα πρωτόκολλα που χρησιμοποιούν οι συσκευές μας. Ωστόσο κυρίως έχει δοθεί έμφαση στις δυνατότητες και τον τρόπο λειτουργίας των λογισμικών Rhasspy και Node-Red.

Λέξεις κλειδια

Διαδίκτυο των πραγμάτων,Φωνητικός βοηθός,Έξυπνο σπίτι,MQTT,Rhasspy,Node-Red

Abstract

The purpose of this diploma thesis is the implementation of an offline opensource voice assistant that works with the Greek Language.The disengagement of the always online and cloud server features,offers a more secure,independent and more customizable solution in comparison with most commercial smart speakers. In the chapters followed we made an introduction about how IoT technologies and smart assistants implement in a smart home,analysing the functions,the way they operate and the technologies they rely upon. We overwiewed the closed box and opensource smart speakers that are available on the market,highlighting the technologies and protocols they use.

In advance we had a brief overwiew about the opensource softwares available and why we choose to use Rhasspy,an open source, fully offline set of voice assistant services for many human languages that works well with other programms and tools that a raspberry pi can run. In the second part of the project we focus on explaining the development of our IoT network using Rhasspy,Node-red and Tasmota firmwared devices,while also analysing the hardware,software,protocols and the way of communicaton among all these tools.

Key words

IoT,Voice Assistant,Smart home,MQTT,Rhasspy,Node-Red

Μελέτη, ανάπτυξη και ενσωμάτωση ψηφιακού φωνητικού βοηθού στο έξυπνο σπίτι με χρήση opensource τεχνολογιών

ΧΑΤΖΟΓΛΟΥ ΚΩΝΣΤΑΝΤΙΝΟΣ

Μάρτιος 2022

Περιεχόμενα

Κατάλογος Εικόνων

Μέρος Ι

Ανάλυση

Εισαγωγή

Ο όρος έξυπνο σπίτι ή smart home χρησιμοποιείται για να περιγράψει ένα σύνολο συσκευών οι οποίες οργανώνονται μέσω ενός συστήματος αυτόματου ελέγχου και ελέγχονται μέσω τηλεχειρισμού σε ένα σπίτι ή κτήριο.

Σκοπός του smart home είναι να προσφέρει μεγαλύτερη άνεση και ευχρηστία στον έλεγχο των συσκευών για την εκπλήρωση αναγκών του εκάστοτε χρήστη. Η αυξημένη ασφάλεια,μέσω καμερών,αισθητήρων κίνησης και άλλων συστημάτων ασφαλείας,όπως και η καλύτερη διαχείριση της ενέργειας στον χώρο,μέσω έξυπνων πριζών και ρελέ όπου καταγράφουν την κατανάλωση είναι μερικοί από τους λόγους που κάποιος θα θελήσει να επενδύσει στην λύση του έξυπνου σπιτιού,ενώ η εξοικονόμηση χρόνου και χρήματος μέσω των συστημάτων αυτών είναι κύριος λόγος. Για να επιτευχθεί κάτι τέτοιο οι αισθητήρες και άλλες συσκευές που χρησιμοποιούνται,είναι συνδεδεμένες μεταξύ τους,επικοινωνούν και ανταλλάσσουν δεδομένα μέσω του Internet ή άλλων ασύρματων δικτύων. Το δίκτυο επικοινωνίας πληθώρας συσκευών,όπως οικιακών συσκευών,ηλεκτρονικών καθώς και κάθε αντικειμένου που είναι προγραμματισμένο ώστε να καθιστά δυνατό το χειρισμός,την παρακολούθηση και την ανταλλαγή δεδομένων ονομάζεται διαδίκτυο των πραγμάτων(Internet of Things).

Αν και ο όρος (Internet of Things) υπάρχει από το 1990 και αποδόθηκε από τον Kevin Ashton,η αύξηση της τεχνολογίας των έξυπνων συσκευών και οι διαδεδομένες εμπορικές λύσεις είναι που έφεραν στο προσκήνιο το έξυπνο σπίτι την τελευταία δεκαετία. Οι πρώτες εταιρίες που χρησιμοποίησαν για εμπορική χρήση αυτή την τεχνολογία ήταν η LG με το πρώτο ψυγείο στον κόσμο με συνδεσιμότητα στο Διαδίκτυο. Το 2008 έγινε η συνεργασία εταίρων της βιομηχανίας που σχηματίστηκαν από την IPSO Alliance,η οποία ενδιαφερόταν για την προώθηση της συνδεσιμότητας των συσκευών. Σύμφωνα με το Cisco Internet Business Solutions Group(IBSG) το 2008 με 2009 ήταν η χρονιά στην οποία περισσότερες συσκευές ήταν συνδεδεμένες στο Internet από ότι άνθρωποι.

Με την αύξηση της εμπορικότητας και της χρήσης των συσκευών που συγκαταλέγονται στο διαδίκτυο των πραγμάτων σε όλο και περισσότερα σπίτια γινόταν απαραίτητη η χρήση ενός κεντρικού σημείου ελέγχου,κοινώς Smart Home Hub.

Πρόκειται για Graphic User Interfaces όπου μέσω αυτών ο χρήστης μπορεί να συνδέσει, ταξινομήσει και ελέγξει όλες τις έξυπνες συσκευές του. Η απλότητα έγκειται στο γεγονός ότι εάν ένας χρήστης έχει συσκευές από διαφορετικούς κατασκευαστές χρειάζεται να ανατρέχει στην εκάστοτε εφαρμογή για τον χειρισμό της συσκευής,ενώ μέσω του hub μπορεί να τις έχει όλες μαζί εφόσον τις συνδέσει,απαραίτητο ωστόσο είναι το Smart Hub να υποστηρίζει το πρωτόκολλο επικοινωνίας της κάθε συσκευής. Σε περίπτωση που θέλουμε οι δικές μας IoT συσκευές να τρέχουν ένα opensource firmware για την συνδεσιμότητά τους είναι πιθανόν να μην γίνουν αναγνωρίσιμες από closed-software hubs όπως αυτά των μεγάλων εταιριών(Google,Amazon,Samsung)αλλά να χρειάζεται μια opensource λύση όπως το Home Assistant ή το OpenHAB. Υπάρχουν ωστόσο cloud-based εφαρμογές που επιτρέπουν την σύνδεση των δυο αυτών κατηγοριών smart hub. Η αλληλεπίδραση με τον χρήστη μπορεί να γίνει είτε μέσω μιας φυσικής συσκευής smart hub,είτε μέσω εφαρμογής που συνήθως τρέχει στο κινητό. Η νέα γενιά εικονικών-φωνητικών βοηθών διαθέτουν πλέον εφαρμογές που μπορεί να λειτουργήσουν και σαν κέντρο ελέγχου όλων των υπόλοιπων έξυπνων συσκευών. Όσο αφορά την εμπορική απήχηση ο αριθμός των smart speaker παγκοσμίως φτάνει τις 320 εκατομμύρια μονάδες με τις πιο μεγάλες αγορές να είναι στην Κίνα και τις ΗΠΑ.Μέχρι το 2024 προβλέπεται ότι ο αριθμός αυτός θα φτάσει τα 640 εκατομμύρια. [7]

Επιπλέον μέχρι το 2021 με βάση τα τελευταία στοιχεία που έχουμε ο συνολικός αριθμός των smart home παγκοσμίως ξεπερνά τα 258 εκατομμύρια.Οι αγορές που καταναλώνουν περισσότερο αυτά τα προϊόντα σήμερα είναι χώρες της Άπω Ανατολής και κυρίως η Κίνα. Ενώ στο μέλλον φαίνεται ότι ο συνολικός αριθμός των συσκευών που αποστέλλονται παγκοσμίως θα αυξηθεί τα επόμενα χρόνια,η γεωγραφική διάταξη των αγορών αναμένεται να παραμείνει η ίδια. Η αγορά των smart home συσκευών περιλαμβάνει τέσσερα κυρίως τμήματα.

- Smart Home Automation:συσκευές ελέγχου και αυτοματισμού,όπως θερμοστάτες, λάμπες,κλειδαριές κτλ.
- Smart Metering:μετρητές κοινής ωφέλειας για την παρακολούθηση της κατανάλωσης και του κόστους.

- Smart appliances:μεγάλες ”λευκές συσκευές” οι οποίες χρησιμοποιούν συστήματα IoT.
- Smart Entertainment:συσκευές στις οποίες περιλαμβάνονται τα smart speakers,οι τηλεοράσεις και τα media players.

Κεφάλαιο 1

Ανάλυση φωνητικών βοηθών

Σε αυτό το κεφάλαιο γίνεται ιδιαίτερη ανάλυση στους φωνητικούς βοηθούς και στην μετέπειτα εξέλιξή τους μέσα από τους εικονικούς βοηθούς και τα έξυπνα ηχεία.Οι νέες τεχνολογίες που κάνουν πιο άμεση την αλληλεπίδραση με τον χρήστη,η ένταξή τους στα IoT δίκτυα και η χρήση τους ως κεντρικά σημεία ελέγχου είναι οι κύριοι λόγοι για την εμπορική τους επιτυχία και κατ' επέκταση την εγκατάστασή τους τόσο στα σπίτια όσο και σε κινητές συσκευές.Ωστόσο η φιλοσοφία τους περί ιδιωτικότητας του χρήστη έχει προκαλέσει κριτική και προβληματισμούς σε αρκετό ποσοστό των καταναλωτών.

1.1 Λειτουργία και Δυνατότητες Εμπορικών VA

Για την κατανόση της διπλωματική εργασίας είναι απαραίτητο να εξηγήσουμε τον όρο,την χρησιμότητα όσο και τον τρόπο λειτουργίας των ψηφιακών βοηθών.

Στην ουσία οι ψηφιακοί βοηθοί είναι εφαρμογές λογισμικού πράκτορα οι οποίοι μπορούν να εκτελέσουν εργασίες και υπηρεσίες μέσω εντολών ή ερωτήσεων. Με τον όρο πράκτορας συγκαταλέγεται μια πληθώρα λογισμικών που εκτελούν διαδικασίες ή υπηρεσίες για ένα άτομο. Το μέσο αλληλεπίδρασης με το λογισμικό μπορεί να είναι κείμενο,εικόνες και βίντεο όπως και η ομιλία.Με βάση αυτού και του τρόπου λειτουργίας τους διαχωρίζονται σε διάφορες κατηγορίες. Οι κατηγορίες που θα αναλύσουμε είναι οι Εικονικοί Ψηφιακοί Βοηθοί(Virtual Digital Assistants),Έξυπνοι Βοηθοί (Smart Assistants) και οι Φωνητικοί Βοηθοί (Voice Assistants) [8]. Αν και συχνά πρόκειται για όρους οι οποίοι συγχέονται,οι δυνατότητές τους μπορεί να διαφέρουν. ΣτουςVirtual Digital Assistant συγκαταλέγονται αυτο-

ματοποιημένες εφαρμογές λογισμικού ή πλατφόρμες που χρησιμοποιούν την φυσική ομιλία ή γραπτό κείμενο για την διεκπεραίωση απλών και επαναλαμβανόμενων εργασιών από τον χρήστη. Η φυσική εξέλιξη των ψηφιακών βοηθών (digital assistants) είναι οι voice assistants,ψηφιακοί βοηθοί που χρησιμοποιούν φωνητική αναγνώριση,σύνθεση ομιλίας και την επεξεργασία της φυσικής γλώσσας (NLP,natural language processsing) για να παρέχουν υπηρεσίες μέσω συγκεκριμένων εφαρμογών. Τεχνολογικά ωστόσο υπερτερούν οι Smart Assistans καθώς πέρα από το ότι "ακούνε" μέσω του μικροφώνου και επικοινωνούνε από τα ηχεία,η άμεση σύνδεση στο διαδίκτυο τους επιτρέπει να λειτουργούν μέσω αλγορίθμων και να έχουν πρόσβαση σε μεγάλες βάσεις δεδομένων ώστε να εκτελέσουν πιο περίπλοκες διεργασίες.Κυρίως αναφερόμαστε σε συσκευές μικρού μέγεθος οπού χρησιμοποιούν τα λεγόμενα smart speakers τα οποία αναμένουν για την λέξη αφύπνισης προτού ενεργοποιηθούν.

Αναδρομικά η τεχνολογία των voice assistant,των ψηφιακών βοηθών που δέχονται ως είσοδο/έξοδο την φυσική ομιλία έχει χρησιμοποιηθεί από το 1960 με το Shoebox,ενός υπολογιστή της IBM όπου μπορούσε να αναγνωρίσει 16 λέξεις και 10 ψηφία μέσω της ομιλίας και θεωρείται πρόδρομος των σημερινών συστημάτων αναγνώρισης φωνής. [9]
Τα τελευταία χρόνια παρακολουθούμε μεγάλη αύξηση στις ικανότητες και την χρήση των ψηφιακών βοηθών και πλέον απευθύνονται στον μέσο χρήστη.Η νέα εποχή των voice assistants άρχισε να ενσωματώνεται στα smartphone και να αξιοποιεί τεχνολογίες όπως μηχανική μάθηση και νευρωτικά δίκτυα οι οποίες συλλέγοντας τα δεδομένα του χρήστη προσφέρουν μια πιο εξατομικευμένη εμπειρία. Πρωτοπόρος σε αυτή την τεχνολογία ήταν η Apple με τον εικονικό βοηθό Siri το 2011,όπου αποτελούσε μέρος των λειτουργικών συστημάτων iOS της Apple τόσο στα κινητά όσο και στα macbook. Το 2012 ακολούθησε η Google με τον εικονικό βοηθό Google Now οπού βρισκόταν ενσωματωμένος στην μηχανή αναζήτηση της εταιρίας χωρίς ωστόσο να διαθέτει τεχνολογίες που να τις επιτρέπουν να επικοινωνεί αμφίδρομα με τον χρήστη μέσω φυσικής γλώσσας,ενώ μερικά χρόνια έπειτα,το 2016 είχαμε τον πλήρη επανδρωμένο με τεχνίτη νοημοσύνη εικονικό βοηθό Google Assistant,ο οποίος ήρθε να αντικαταστήσει το Google Now τόσο στις smartphone συσκευές και να ενσωματωθεί στο smart speaker Google Home.Το 2013 η Microsoft ενσωμάτωσε τα Windows με τον δικό της εικονικό βοηθό την Cortana,όπου μπορούσε να απαντήσει σε ερωτήσεις χρησιμοποιώντας πληροφορίες από τη μηχανή αναζήτησης Bing.Ενώ το 2014 η Amazon προσφέρει τη δική της πρόταση με την Alexa και το Amazon Echo μία συσκευή smart speaker όπου χρη-

σιμοποιεί τον δικό της τεχνολογικά ανεπτυγμένο εικονικό βοηθό. Μέσω της αναγνώρισης της φωνής των χρηστών της και της ενσωματωμένης τεχνητής νοημοσύνης οι εικονικοί βοηθοί μπορούν να εκτελέσουν περίπλοκες διεργασίες βασισμένοι σε αλγόριθμους μηχανικής μάθησης Κάποιες εργασίες που μπορεί να κάνει είναι:

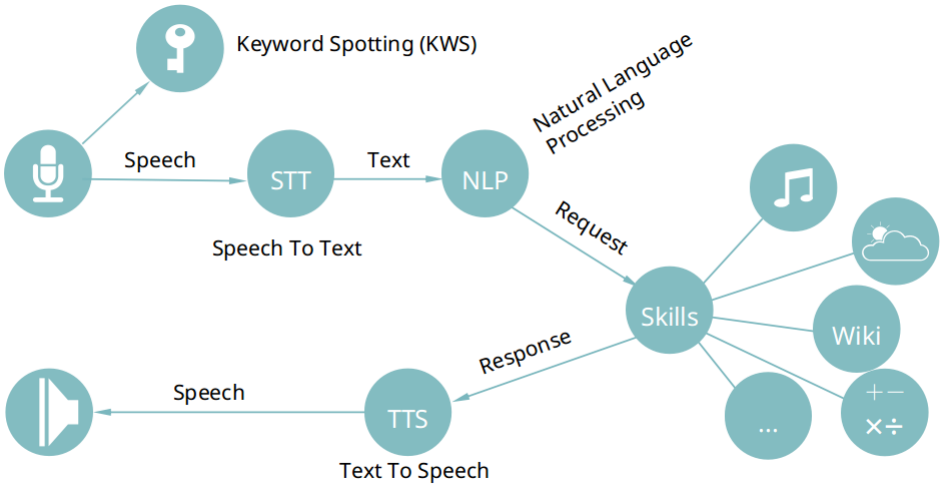
- κρατάει σημειώσεις
- δημιουργία ημερολογίου
- πραγματοποίησή online αγορών
- επικοινωνία με άλλες συσκευές
- αναζήτηση στον Ιστό
- δημιουργία υπενθύμισης
- ενημέρωση του καιρού

Οι εικονικοί βοηθοί,συχνά ονομάζονται και προσωπικοί βοηθοί,λόγω της παραμετροποιημένης λειτουργίας τους με βάση τον χρήστη. Ο κάθε εικονικός βοηθός διαφέρει με τους ανταγωνιστές σε διάφορους τομείς όπως η επιτυχής κατανόηση της ομιλίας αναλόγως την απόσταση και τις συνθήκες θορύβου,η αναγνώριση και ταυτοποίηση της φωνής του κύριου χρήστη,στο πόσο εύχρηστος και εξατομικευμένος είναι ο αλγόριθμός αναζήτησης και οι απαντήσεις που δίνει και πόσο γρήγορα ανταποκρίνεται στις φωνητικές εντολές. Ωστόσο ένα κοινό χαρακτηριστικό είναι ότι πρέπει να βρίσκονται σε σύνδεση με το Διαδίκτυο και κατά επέκταση στους servers και τις βάσεις δεδομένων της εταιρίας που έχει κατασκευάσει την συσκευή ώστε να πραγματοποιεί ορθά όλες του τις λειτουργίες,όπως αναζητήσεις,να βρίσκει απαντήσεις ή να επικοινωνεί με άλλες έξυπνες συσκευές. Οι συσκευές ακούνε συνέχεια το περιβάλλον τους αναμένοντας τη λέξη αφύπνισης ή κοινώς wake word ώστε να ξεκινήσουν τη φωνητική αλληλεπίδραση ανάμεσα στον χρήστη και την συσκευή.Όταν ‘‘ξυπνήσουν’’ και αναμένουν να λάβουν φωνητική εντολή,βρίσκονται σε standby mode για ένα χρονικό διάστημα και αν δεν υπάρχει ανταπόκριση,ξαναγυρνάνε στην προηγούμενη κατάσταση. Σε καθένα απο τους smart assistants ο χρόνος του standby mode όπως και η wake word διαφέρει.

Εφόσον έχουμε ενεργό τον φωνητικό βοηθό και “ακούει” η διαδικασία που ακολουθεί έπειτα είναι η εξής: Ο βοηθός στέλνει το φωνητικό μήνυμα που έλαβε στο cloud της αντίστοιχης εταιρίας. Έπειτα ένας αλγόριθμος NLP(επεξεργαστής φυσικής γλώσσας) μετατρέπει την

ομιλία σε κείμενο και αποφασίζει ποία "δεξιότητα" θα εκτελέσει. Η εφαρμογή στέλνει πίσω στο cloud την απάντηση και έπειτα μέσω ενός αλγορίθμου γίνεται μετατροπή του αρχείου κειμένου σε αρχείο φωνής γνωστό ως TTS(Text to Speach,κείμενο σε φωνή),αναπαράγοντας το αρχείο από τα ηχεία της συσκευής.

Η διαδικασία αυτή μπορεί να συνεχιστεί χωρίς την επανάληψη της wake word,μέχρι το πέρας κάποιας χρονικής περιόδου(λίγων δευτερολέπτων,αναλόγως την συσκευή) ή με την φωνητική εντολή "stop".



εικόνα 1.1: Οπτικοποίηση της διαδικασίας [1]

Αναφορά πρέπει να γίνει και στην συνεχή προσθήκη της ήδη μεγάλης γκάμας των δυνατοτήτων και των ενεργειών από προμηθευτές τρίτων κατασκευαστών(third party apps) ώστε να διευκολύνουν την συνδεσιμότητα των συσκευών ή των υπηρεσιών τους με τους έξυπνους εικονικούς βοηθούς. Κάποια χαρακτηριστικά παραδείγματα είναι η αναπαραγωγή και περιήγηση μουσικής μέσω του Spotify,η παραγγελία φαγητού από μια εφαρμογή delivery ακόμα και η βοήθεια στον ύπνο ή τον διαλογισμό με ασκήσεις μέσω της εφαρμογής headspace. Η υποστήριξη των υπηρεσιών διαφέρει αναλόγως τον φωνητικό βοηθό,καθώς δεν υποστηρίζονται όλες οι υπηρεσίες από όλους τους φωνητικούς βοηθούς,ωστόσο με τις πολύ γνωστές εφαρμογές αναπαραγωγής πολυμέσων όπως Youtube,Spotify,Netflix η συμβατότητα είναι καθολική.

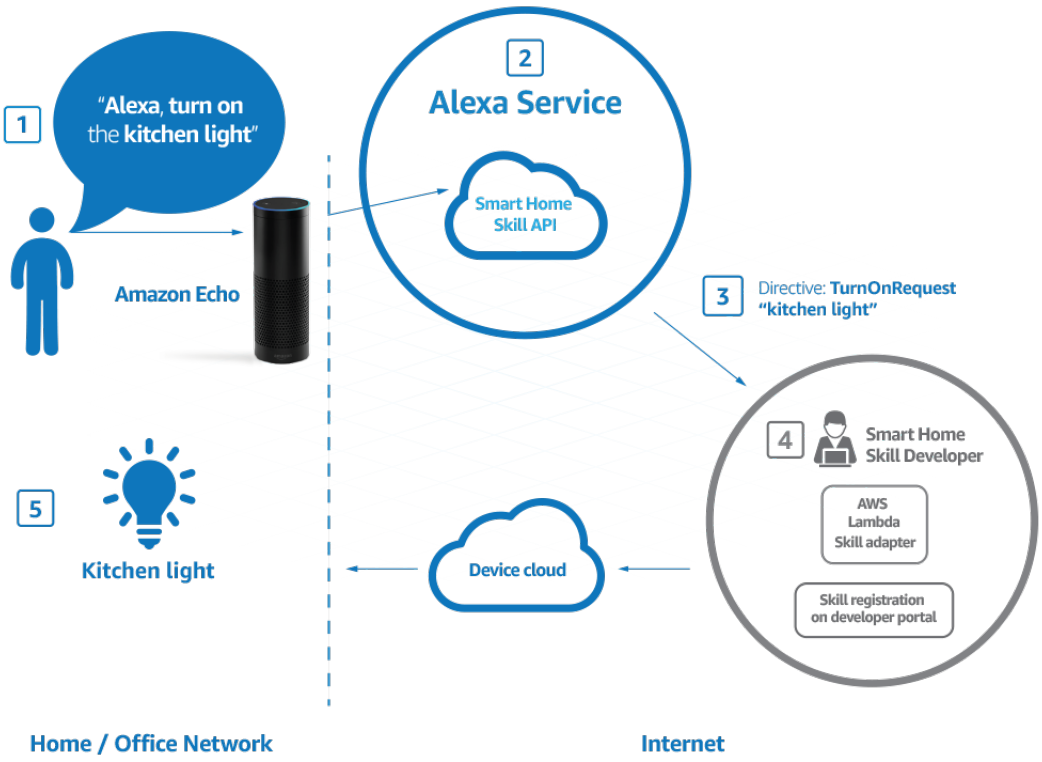
1.1.1 Speaker Recognition

Η διαδικασία αναγνώρισης ενός ατόμου με βάση τη φωνή ονομάζεται speaker recognition. Κάθε άτομο έχει διαφορετικά χαρακτηριστικά στα όργανα που συνθέτουν την ομιλία όπως η φωνητική οδός και ο λάρυγγας,με αποτέλεσμα το φωνητικό σήμα που αποτελεί την κωδικοποίηση της φωνής,να είναι μοναδικό για τον καθένα. Για την αναγνώριση των βιομετρικών αυτών δεδομένων γίνετε χρήση συστημάτων μηχανικής μάθησης.Τα συστήματα αυτά αποτελούνται από διάφορα στάδια,τα οποία δεν θα τα αναλύσουμε διεξοδικά αλλά θα τα αναφέρουμε περιληπτικά. Το πρώτο στάδιο είναι η βάση δεδομένων των ηχητικών εγγραφών,εδώ αξίζει να αναφέρουμε ότι στην αρχική ενεργοποίησή στους εικονικούς βοηθούς των μεγάλων εταιριών,μας ζητείται η ανάγνωση και επανάληψη της λέξης αφύπνισης μέσα σε μια πρόταση αρκετές φορές ώστε να συλλεχθούν τα δεδομένα που θα γίνεται η ταυτοποίηση. Τα δυο πιο διαδεδομένα format ήχου είναι το mp3 και wav,με την δεύτερη μορφή να επιλέγεται περισσότερο σε αυτές τις περιπτώσεις καθώς το mp3 είναι ελλείπείς σε πληροφορίες απέναντι στον τύπο wav,ο οποίος δεν υπόκειται σε συμπίεση. Έπειτα στο στάδιο της προ-επεξεργασίας γίνεται πιο εύκολη η οργάνωση του αρχείου του ήχου με την βοήθεια εξισώσεων και αφαιρώντας τον θόρυβο. Προχωράμε στην εξαγωγή χαρακτηριστικών,στην διαδικασία αυτή γίνεται υπολογισμός μιας συλλογής διανυσμάτων τα οποία παρέχουν μια συμπαγής αναπαράσταση ενός σήματος ομιλίας μέσω αλγορίθμων,χαρακτηριστικό παράδειγμα ο αλγόριθμος παλινδρόμησης τυχαίων δασών.Εξαγωγή αυτής της διαδικασίας είναι η ψηφιακή αναπαράσταση του σήματος. Στο επόμενο στάδιο χρησιμοποιούνται αλγόριθμοι ταξινόμησης για να διαχωρίσουν την training phase με την testing phase. Τέλος παιρνόντας μέσα απο εφτά αλγορίθμους τεχνικές μηχανικής μάθησης στην φάση της εκπαίδευσης το σύστημα καταλήγει στο αποτέλεσμα της πρόβλεψης. [10]

1.1.2 Smart Home Hub και συνδεσιμότητα

Smart Home Hub μπορούμε να ονομάσουμε οποιαδήποτε συσκευή ή λογισμικό γεφυρώνει τις υπόλοιπες συσκευές σε ένα IoT δίκτυο και συντονίζει την επικοινωνία τους,στην ουσία πρόκειται για ένα κέντρο ελέγχου ενός smart home.Μπορεί να είναι μια υπολογιστική μονάδα ή απλώς ένα λογισμικό,στο οποίο ο χρήστης έχει πρόσβαση είτε βρίσκοντας εντός του ίδιου δικτύου,είτε μέσω ενός server που επικοινωνεί με το δίκτυο του σπιτιού. Οι συσκευές των smartspeakers που αναφέραμε παραπάνω μπορούν να χρησιμοποιηθούν και για αυτό το σκοπό.

Ένα παράδειγμα χρήσης ενός εικονικού βοηθού σαν smart hub είναι εάν χρησιμοποιούμε μια έξυπνη λάμπα της Smart Life και έχουμε στον χώρο μας μια συσκευή Alexa, να πληκτρολογήσουμε από το menu skills(αντίστοιχα actions για το Google) της εφαρμογής Alexa το όνομα της εταιρίας,στην προκειμένη περίπτωση Smart Life. Κατεβάζοντας το skill της Smart Life στην Alexa,θα μπορούμε πλεον μέσω φωνητικών εντολών που δίνουμε στην ψηφιακό βοηθό να ελέγχουμε τις έξυπνες συσκευές της Smart Life.



εικόνα 1.2: Λειτουργία του έξυπνου ηχείου ως smart hub [2]

Ωστόσο υπάρχουν πολλοί περισσότερες εμπορικές λύσεις που επικεντρώνονται κυρίως σε αυτό το σκοπό και όχι στην ενσωμάτωση των εικονικών βοηθών.

Εταιρίες όπως οι Sonoff,Philips,Broadlink πέρα απο τις έξυπνες λάμπες,πρίζες,αισθητήρες και άλλες έξυπνες συσκευές έχουν και δικό τους smart hub,τα οποία χρησιμοποιούν διάφορα πρωτόκολλα επικοινωνίας απο bluetooth,rf,zigbee μέχρι wi-fi ενώ είναι δυνατή η συνδεσή τους με τους εικονικούς βοηθούς των μεγάλων εταιριών (google-amazon-apple). Επίσης το OpenHub και Home Assistant αποτελούν opensource εφαρμογές των smart hubs τα οποία προσφέρουν μια μεγαλύτερη παραμετροποίηση και απευθύνονται τόσο σε απλούς χρήστες,όσο και σε προγραμματιστές.

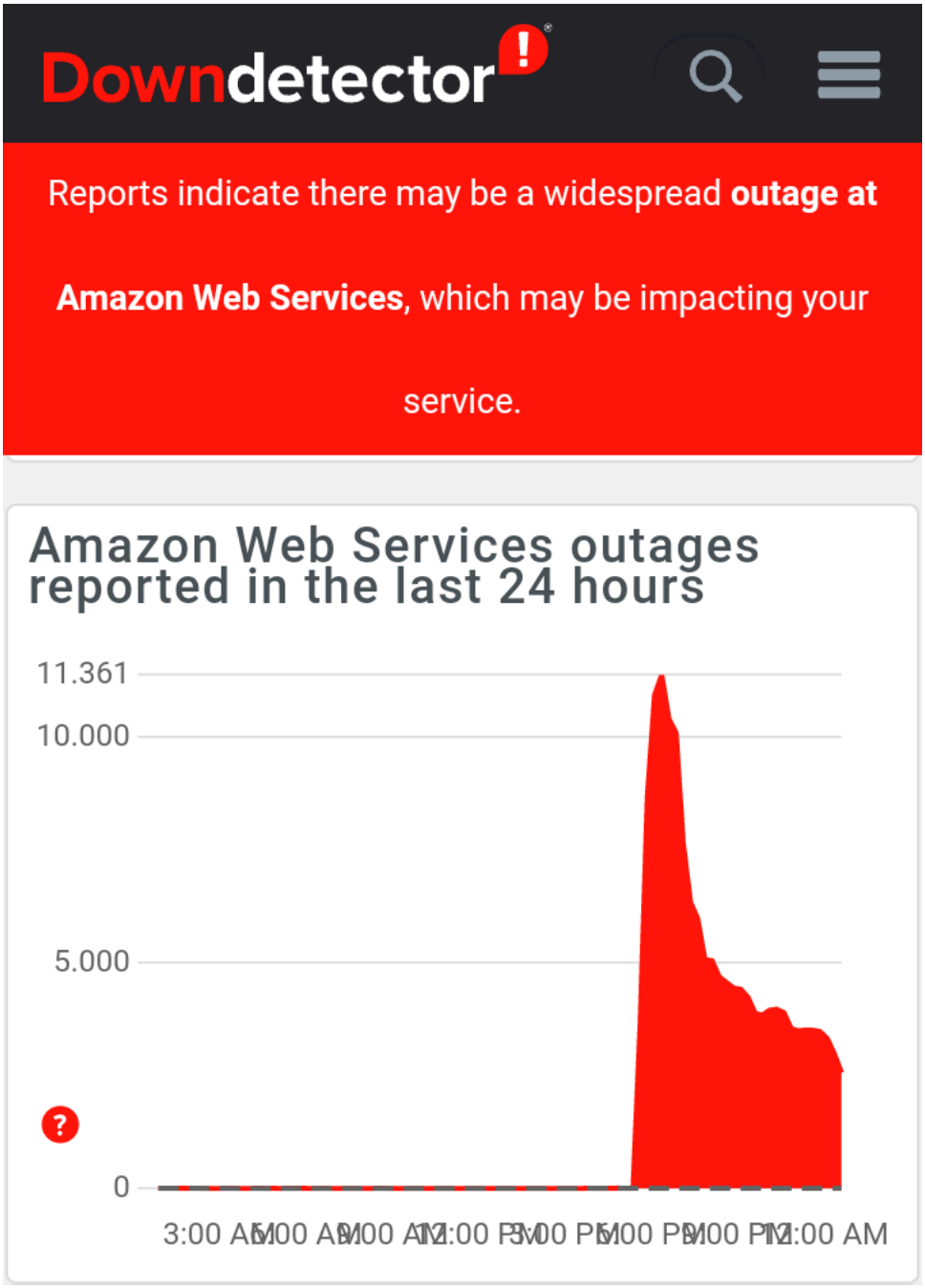
1.1.3 Ασφάλεια και Ιδιωτικότητα

Οι λειτουργίες των προσωπικών βοηθών όπως είναι η ενεργοποίηση μέσω της φωνής η οποία προϋποθέτει ότι η συσκευή ακούει συνεχώς,έχουν δημιουργήσει πολλές ανησυχίες όσο αφορά την ιδιωτικότητα του χρήστη. Μελέτη που παρουσιάστηκε το 2019 από το Michigan State University [11] δείχνει ότι οι Home Digital Voice Assistants(HDVAs) είναι αρκετά ευάλωτοι όσο αφορά την ασφάλεια τους. Αναγνώρισαν τρία τρωτά σημεία στην ασφάλεια και σχεδίασαν δυο κακόβουλες επιθέσεις οι οποίες πέτυχαν.

Σε περίπτωση διάρρηξης στο σπίτι,ο διαρρήκτης μπορεί να ελέγξει τον Smart Assistant και μέσω φωνητικών εντολών να έχει άμεση πρόσβαση στις υπόλοιπες έξυπνες συσκευές.Εάν η είσοδος κάποιων χώρων του σπιτιού ελέγχονται μέσω τέτοιων συσκευών(όπως το γκαράζ) είναι δυνατή η πρόσβαση σε αυτούς.

Η άλλη περίπτωση είναι η πραγματοποίηση παραγγελιών από τρίτους χωρίς την έγκριση του πραγματικού χρήστη. Για την συγκεκριμένη μελέτη χρησιμοποίησαν την Amazon Alexa καθώς αποτελεί την πιο διαδεδομένη συσκευή οικιακής χρήσης.Οι υπηρεσίες της Amazon μπορούν να έχουν άμεση πρόσβαση στα στοιχεία που χρησιμοποιεί κάποιος για να πραγματοποιήσει παραγγελία από τον ομώνυμο site. Όπως φάνηκε λόγω του ελέγχου ταυτότητας με ένα παράγοντα και την μη ενεργοποίησή του speaker recognition,η πρόσβαση στη συσκευή ήταν πολύ εύκολη να παρακαμφθεί από τον καθένα που ήξερε την λέξη αφύπνισης ώστε να ξεκινήσει την αλληλεπίδραση με τον έξυπνο βοηθό.Η συσκευή ενεργοποιούνταν ανεξάρτητα εάν βρισκόταν άτομο στον χώρο και εφόσον έπιανε τον ήχο στην επιθυμητή ένταση(τουλάχιστον 60dB). Όσο αφορά τις τρίτες υπηρεσίες και συγκεκριμένα για όσες ενεργοποιόντουσαν μέσω φωνητικών εντολών της Alexa δεν έχει αναπτυχθεί κανένας έλεγχος πρόσβασης,καθώς θεωρούν ότι οι φωνητικές εντολές από τις υπηρεσίες της Alexa είναι πάντα καλόβουλες.

Το κυριότερο ωστόσο μειονέκτημά των voice assistants είναι τα τυχόν προβλήματα στη σύνδεση.Είτε του δικού μας δικτύου,είτε των ίδιων των εταιριών.Λόγω της άμεσης εξάρτησης με τους servers είναι αδύνατο να πραγματοποιήσουν τις περισσότερες λειτουργίες τους εφόσον δεν βρίσκονται σε σύνδεση με τον online server. Ένα τέτοιο περιστατικό μεγάλης κλίμακας συνέβη στις 7 Δεκεμβρίου 2021.Σε πολλά μέρη του πλανήτη οι συσκευές της Amazon σταμάτησαν να λειτουργούν καθώς οι Amazon Web Services είχαν πέσει.1.3



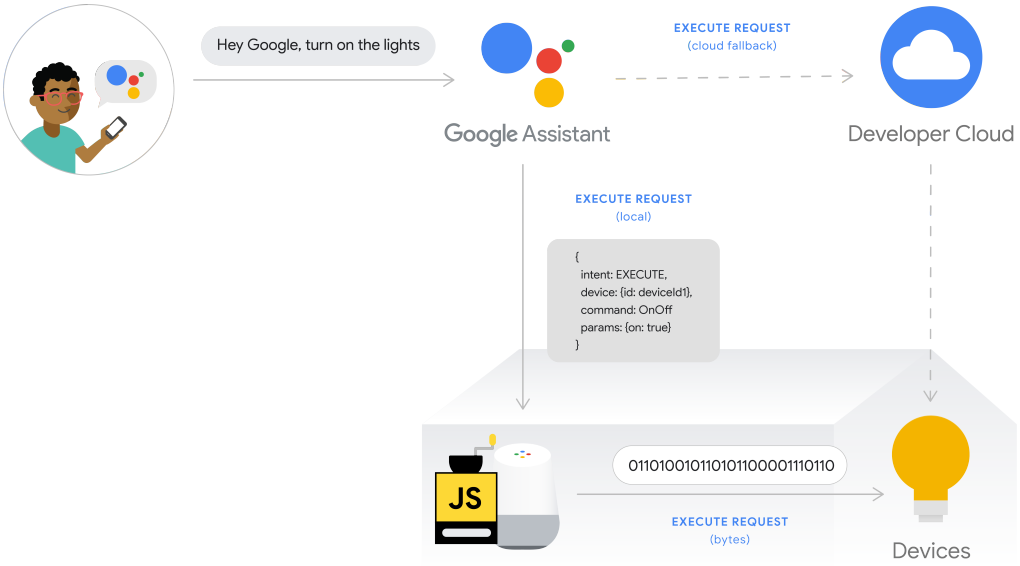
εικόνα 1.3: Διάγραμμα με τις διακοπές του AWS [3]

Στις περιπτώσεις όπου ο έξυπνος βοηθός είναι συνδεδεμένος με κάποιο smart hub,η αποσύνδεση του με τις συσκευές του χώρου στον οποίο βρίσκεται μπορεί να προκαλέσει μεγάλα προβλήματα.Στην περίπτωση της Amazon [12] με την ενεργοποίηση του Local Voice Control αυτό είναι δυνατό να αντιμετωπιστεί.Μέσω του ασύρματου πρωτοκόλλου Zigbee είναι δυνατή η επικοινωνία των συσκευών που βρίσκονται σε κοντινή απόσταση χωρίς την απαραίτητη πρόσβαση των cloud servers.Έτσι είναι εφικτή η χρήση του φωνητικού βοηθού

για κάποιες λειτουργίες όπως :

- Ο έλεγχος συμβατών συσκευών όπως διακόπτες,λάμπες και πρίζες οι οποίες είναι απευθείας συνδεδεμένες με τις Echo συσκευές
- Η ακύρωση υπενθυμίσεων οι οποίες είχαν προγραμματιστεί πριν ο βοηθός μπεί εκτός σύνδεσης
- Η ανάγνωση της ώρας και ημέρας

Αντίστοιχα η Google χρησιμοποιεί το Local Home SDK.Πρόκειται για μια παρόμοια υπηρεσία όπου παρακάμπτει την διεπαφή με τον Server και επικοινωνεί με τις συσκευές απευθείας μέσω του LAN στο ρούτερ.Μέσω αυτού του τρόπου είναι εμφανής και η βελτίωση στο χρόνο απόκρισης. [13]



εικόνα 1.4: Λειτουργία Google Home παρακάμπτοντας το cloud [4]

1.1.4 Πολιτικές Προστασίας Διακεκριμένων VA

Έρευνα που διεξάχθηκε απο το Reviews.org [14] και αφορά τη χρήση των smart assistants έδειξε ότι το 56% των χρηστών έχουν ανησυχίες όσο αφορά την συλλογή των δεδομένων τους. Μελετώντας τους όρους χρήσης και προϋποθέσεις των Alexa, Google Assistant, Siri, Bixby και Cortana,φαίνεται ότι και οι πέντε υπηρεσίες συλλέγουν αρκετά στοιχεία,όπως το όνομα,τηλεφωνικό αριθμό,την τοποθεσία της συσκευής και την IP διεύθυνση.Επίσης μπορούν να έχουν πρόσβαση στις επαφές του χρήστη και στο ιστορικό,όπως και να γνωρίζουν

τις εφαρμογές που χρησιμοποιεί. Στην έρευνα το 60% των ερωτηθέντων τρέφουν ανησυχίες ότι οι συζητήσεις τους ακούγονται από κάποιον άλλον.Ενώ η Siri και το Google assistant χρειάζονται άδεια για να ηχογραφήσουν τις αλληλεπιδράσεις του χρήστη,οι υπόλοιποι voice assistants το κάνουν αυτό από προεπιλογή. Οι τύποι δεδομένων που μπορούν να συλλέξουν και εξέτασαν στην έρευνα είναι 48 και ποικίλουν από τα δεδομένα των επαφών,τα αρχεία και τις δραστηριότητες εώς και τις συσκευές που συνδέονται στο δίκτυο,την ώρα και τα ονόματα της συσκευής,μέχρι και τον πάροχο του Ίντερνετ.

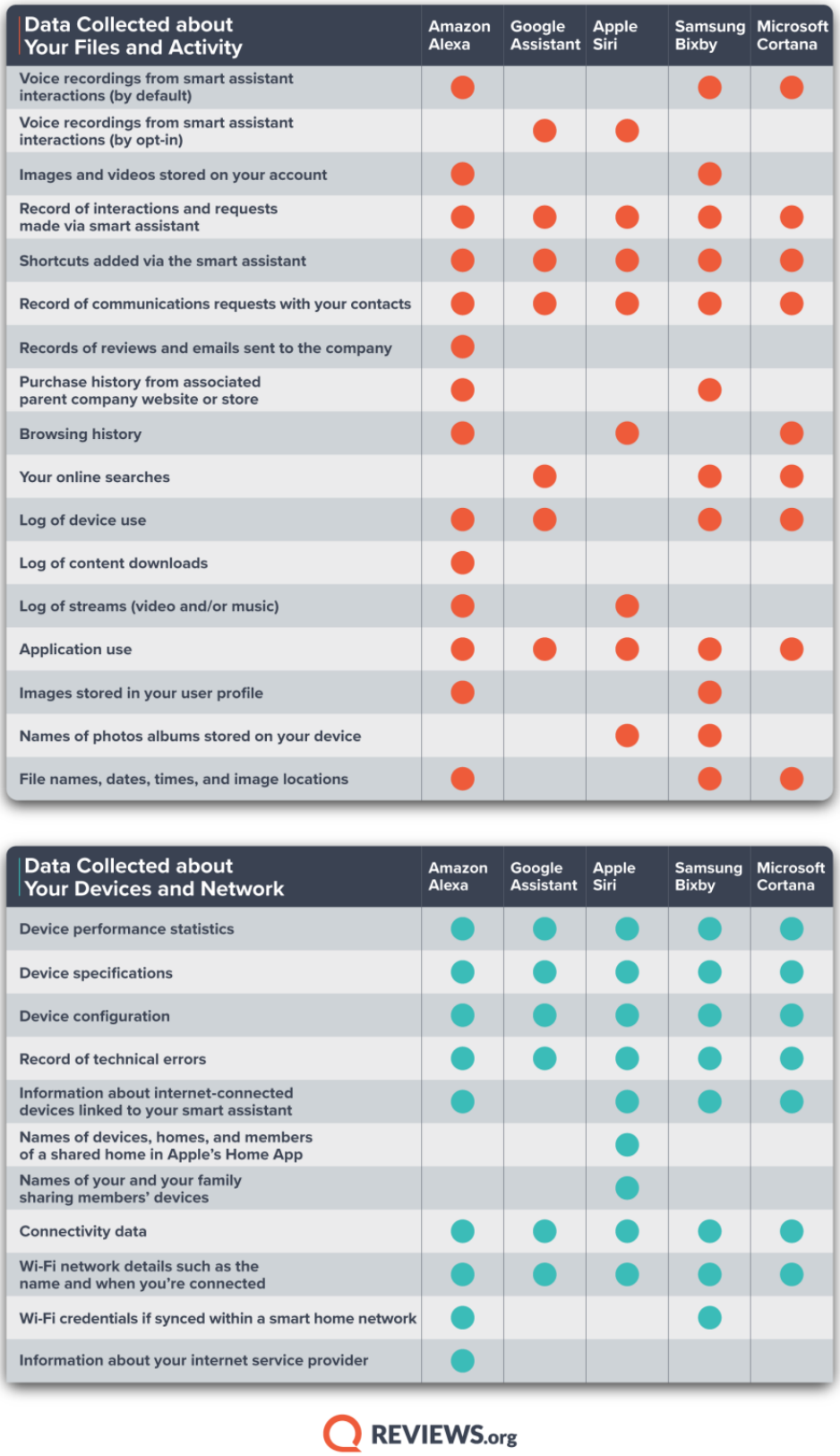
Στα διαγράμματα που ακολουθούν μπορούμε να δούμε αναλυτικά τα δεδομένα για τον εκάστοτε βοηθό.

Data Smart Assistants Collect about You

Data Collected about You	Amazon Alexa	Google Assistant	Apple Siri	Samsung Bixby	Microsoft Cortana
Your name	●	●	●	●	●
Your time zone	●	●	●	●	●
Address	●		●	●	
Phone number(s)	●	●	●	●	●
Payment information	●		●	●	
Your age	●			●	●
Personal interests as stored in your user profile	●	●		●	●
Personal description as stored in your user profile	●			●	
The location of your device or computer	●	●	●	●	●
Location history, places, and routes		●			●
Your IP address	●	●	●	●	●
Your synced email					●
Your calendar				●	●
Acoustic model of voice characteristics	●	●	●		●

Data Collected about Your Contacts	Amazon Alexa	Google Assistant	Apple Siri	Samsung Bixby	Microsoft Cortana
Names for stored contacts	●	●	●	●	●
Nicknames for stored contacts		●	●		●
Relationships for stored contacts		●	●		
Phone numbers for stored contacts	●	●	●	●	●
Addresses for stored contacts	●	●		●	
Email addresses of stored contacts	●	●		●	●

εικόνα 1.5: Δεδομένα που συλλέγει ο κάθε φωνητικός βοηθός [5]



Data Collected about Your Files and Activity	Amazon Alexa	Google Assistant	Apple Siri	Samsung Bixby	Microsoft Cortana
Voice recordings from smart assistant interactions (by default)	●			●	●
Voice recordings from smart assistant interactions (by opt-in)		●	●		
Images and videos stored on your account	●			●	
Record of interactions and requests made via smart assistant	●	●	●	●	●
Shortcuts added via the smart assistant	●	●	●	●	●
Record of communications requests with your contacts	●	●	●	●	●
Records of reviews and emails sent to the company	●				
Purchase history from associated parent company website or store	●			●	
Browsing history	●		●		●
Your online searches		●		●	●
Log of device use	●	●		●	●
Log of content downloads	●				
Log of streams (video and/or music)	●		●		
Application use	●	●	●	●	●
Images stored in your user profile	●			●	
Names of photos albums stored on your device			●	●	
File names, dates, times, and image locations	●			●	●

Data Collected about Your Devices and Network	Amazon Alexa	Google Assistant	Apple Siri	Samsung Bixby	Microsoft Cortana
Device performance statistics	●	●	●	●	●
Device specifications	●	●	●	●	●
Device configuration	●	●	●	●	●
Record of technical errors	●	●	●	●	●
Information about internet-connected devices linked to your smart assistant	●		●	●	●
Names of devices, homes, and members of a shared home in Apple's Home App			●		
Names of your and your family sharing members' devices			●		
Connectivity data	●	●	●	●	●
Wi-Fi network details such as the name and when you're connected	●	●	●	●	●
Wi-Fi credentials if synced within a smart home network	●			●	
Information about your internet service provider	●				

εικόνα 1.6: Δεδομένα που συλλέγει ο κάθε φωνητικός βοηθός II

Για να διασφαλίσουν την προστασία των προσωπικών δεδομένων από τους πελάτες τους οι διακεκριμένες εταιρίες στον χώρο έχουν πάρει κάποια μέτρα.

- Alexa

  Η Alexa αρχίζει την καταγραφή ήχου μόνο μετά την λέξη αφύπνισης [15]

- Google

  Η διατήρηση των αρχείων του ήχου είναι απενεργοποιημένη εξ'αρχής και θα πρέπει ο χρήστης μέσω του Voice & Audio Activity (VAA) να δώσει άδεια στη συσκευή ώστε να καταγράφει τη φωνή του,σε αυτή την περίπτωση είναι δυνατό να αξιολογηθεί από ανθρώπινο παράγοντα ώστε η συσκευή να αναγνωρίζει καλύτερα τη φωνή του χρήστη [16]

- Siri

  Σε παρόμοιο πλαίσιο κινείται και η apple αφού μας εξασφαλίζει ότι πλέον δεν θα γίνεται διατήρηση των ηχητικών μας δεδομένων,ενώ σε περίπτωση που οι χρήστες επιλέξουν να συμμετέχουν στην βελτίωση της ποιότητας της συσκευής(customers opt in),η εταιρεία εγγυάται ότι μόνο υπάλληλοι της θα έχουν πρόσβαση σε αυτά τα δεδομένα και όχι τρίτες εταιρίες. [17]

Όσο αφορά το hardware των συσκευών,όταν κάνουμε mute την συσκευή Google Home μέσω του διακόπτη στο πίσω μέρος,το μικρόφωνο αποσυνδέεται από το ρεύμα.Διαφορετική προσέγγιση έχει το Amazon echo dot όπου το mute της συσκευής γίνεται μέσω του λογισμικού χωρίς ποτέ το μικρόφωνο να αποσυνδεθεί απο την παροχή ρεύματος.

Κεφάλαιο 2

Open Source Λύσεις

Στον τομέα των εικονικών βοηθών όπως αναφέραμε και παραπάνω,πέρα από τις εμπορικές λύσεις,υπάρχουν και opensource επιλογές,δηλαδή λογισμικά ανοιχτού κώδικα τα οποία διατίθενται δωρεάν. Με την τεχνολογία των voice assistant να βρίσκεται ακόμη σε αναπτυσσόμενο στάδιο, πολλές επιχειρήσεις που θέλουν να εξοπλιστούν με ψηφιακούς βοηθούς,βρίσκουν πιο λογική μια λύση ανοιχτού κώδικα καθώς τους επιτρέπει να διαμορφώσουν και να πειραματιστούν περισσότερο με βάση τις δικές τους απαιτήσεις. ‘Ενα ακόμα κριτήριο που κάνει πολύ ταιριαστή μια τέτοια επιλογή είναι η ασφάλεια. Οι περισσότερες opensource επιλογές λειτουργούν χωρίς να βασίζουν την λειτουργία τους σε εξωτερικούς παράγοντες όπως είναι οι servers των εκάστοσε εταιριών,αλλά οι φωνητικές εντολές του χρήστη περιορίζονται μόνο εντός του δικτύου του,αυξάνοντας και με αυτό τον τρόπο την ταχύτητα που ανταποκρίνονται. Επίσης σε πειραματικό στάδιο μας βοηθάει να κατανοήσουμε τον τρόπο που λειτουργούν και επικοινωνούν τα κεντρικά συστήματα ελέγχου δηλαδή τα hubs με τις υπόλοιπες συσκευές.

2.1 Mycroft

Πρόκειται για τον πιο γνωστό opensource ψηφιακό βοηθό ο οποίος χτίστηκε πάνω στην γλώσσα προγραμματισμού Python.Πέρα απο το λογισμικό υπάρχει διαθέσιμη και εμπορική hardware συσκευή.Οι Mycroft Mark συσκευές έχουν τις δυνατότητες των έξυπνων ηχείων τα οποία λειτουργούν με παρόμοιο τρόπο όπως τα Alexa Echo Dot και Google Nest με την διαφορά ότι δεν πρόκειται για “μαύρα” κουτιά αλλά ο κώδικας μπορεί να παραμετροποιη-

θεί,να αντιγραφεί και να διανεμηθεί σε ολόκληρη την κοινότητα του Mycroft. [18] Πυρήνας του Mycroft Mark I είναι το Raspberry Pi 3,το οποίο βρίσκεται μέσα στη συσκευή,μαζί με το ενσωματομένο ηχείο.Ενώ κάποια από τα υπόλοιπα τεχνικά χαρακτηριστικά είναι οι εξωτερικές θύρες ήχου RCA,η LED οθόνη,το ενσωματομένο Wi-Fi και ένα Arduino Mini. Το συγκεκριμένο πρόγραμμα μπορεί να τρέξει ανεξάρτητα τόσο σε υπολογιστές με Linux όσο και σε Raspberry Pi(απο 3B και νεότερα) με την υποστήριξη των κατάλληλων περιφερειακών,δηλαδή μια συσκευή εισόδου και εξόδου ήχου. Επιπρόσθετα μέσω της ανοιχτής opensource κοινότητας έχουν κυκλοφορήσει και άλλοι τρόποι όπου μπορεί να εγκατασταθεί το Mycroft.Χτίζοντας το Mycroft μέσω του Android Studio,μπορούμε να το τρέξουμε στις συσκευές Android,ενώ μέσω του KDE Plasmoid και του Docker μας δίνονται παραπάνω επιλογές και δυνατότητες για την εγκαταστασή του φωνητικού βοηθού. Αναφορά πρέπει να γίνει και στο νέο μοντέλο Mark II που βρίσκεται υπό ανάπτυξη και θα κυκλοφορήσει τον Σεπτέβριο του 2022.Εξοπλισμένο με ενσωματωμένη οθόνη και κάμερα,πιστό στην φιλοσοφία του opensource δίνει την δυνατότητα χρήσης ακόμα και offline σε συγκεκριμένα skills.

εικόνα 2.1: Mycroft Mark I [6]

2.1.1 Picroft

Η έκδοση του Mycroft που μπορούμε να ενσωματώσουμε σε ένα Raspberry Pi ονομάζεται Picroft και βασίζεται στο λογισμικό Raspbian Buster Lite. Για την εγκατάσταση θα χρειαστεί να περάσουμε το λειτουργικό σύστημα σε μία sd card.'Επειτα αφού γίνει εγκατάσταση,μας ζητείται να κάνουμε σύζευξη της συσκευής μας με την Mycroft AI.Με τον μοναδικό κωδικό που εμφανίζεται στην οθόνη πρέπει να κάνουμε register στην επίσημη σελίδα της Mycroft,σε

αυτή την διαδικασία μας ζητάει απαραίτητα να κατοχυρώσουμε και την τοποθεσία μας.

2.2 Jasper

Το Jasper κυκλοφόρησε το 2014 με την άδεια του MIT απο τούς Charles Marsh & Shubhro Saha και πρόκειται για ένα opensource project το οποίο είναι ειδικά σχεδιασμένο για να λειτουργεί κυρίως με το Raspberry Pi. Ο συγκεκριμένος ψηφιακός φωνητικός βοηθός χρησιμοποιεί κάποιες βιβλιοθήκες της Python και έχει μεγάλη ποικιλία στα εργαλεία που μπορεί να επιλέξει ο χρήστης για την εκάστοτε λειτουργία. Για την μετατροπή της ομιλίας σε κείμενο,το γνωστό STT(speech to text),έχουμε να επιλέξουμε ανάμεσα στις εξής μηχανές: Google STT,Pocketshinx,AT T STT i,Wit.ai STT ,Julius. Πέρα απο την Pocketshinx και την Julius οι υπόλοιπες μηχανές βασίζονται στη μεταφορά των δεδομένων στο Internet. [19]

2.3 Rhasspy

Επηρεασμένο απο τον φωνητικό βοηθό Jasper πρόκειται για τον φυσικό του διάδοχο,το Rhasspy συνδυάζει όλα τα προηγούμενα εργαλεία καθώς προσθέτει και καινούργια. Διαθέσιμο απο το 2019,ύπο την άδεια του MIT,ο προγραμματιστής του Rhasspy,Michael Hansen, παραμετροποιεί,προσθέτει και αναβαθμίζει τις λειτουργίες του φωνητικού βοηθού τα τελευταία τρία χρόνια. Βασισμένο στην opensource φιλοσοφία είναι ο φωνητικός βοηθός που επιλέξαμε για να την υλοποίηση του σεναρίου μας. Για αυτό το λόγο αναλύεται διεξοδικά σε παρακάτω κεφάλαιο. Τα κύρια χαρακτηριστικά που το έκαναν να ξεχωρίζει είναι η δυνατότητα υποστήριξης πολλών φυσικών γλωσσών,το γραφικό του περιβάλλον(GUI),η άμεση υποστήριξή του και η μακροπρόθεσμη λειτουργία του στο μέλλον,το αναλυτικό και κατατοπιστικό documentation όπως και ο συνδυασμός του με τα άλλα προγράμματα που είμαστε εξοικειωμένοι ή μας είναι χρήσιμα(Home Assistant,Node-Red) [20]

Κεφάλαιο 3

Ενσωμάτωση στο έξυπνο σπίτι

Σε αυτό το κεφάλαιο εξηγούμε γραπτώς και μέσω διαγράμματος την υλοποίηση του δικού μας σεναρίου που πραγματοποιήσαμε,πριν προχωρήσουμε στο 2ο μέρος όπου παρουσιάζουμε την εγκατάσταση,τον προγραμματισμό και αναλύουμε περαιτέρω την λειτουργία των software και hardware εμβαθύνοντας σε πιο τεχνικούς όρους.

Στόχος του σεναρίου μας είναι η χρήση και λειτουργία των έξυπνων συσκευών του σπιτιού μέσω φωνητικού βοηθού με την χρήση και αναγνώριση της ελληνικής γλώσσας,διασφαλίζοντας ότι η επικοινωνία και σύνδεση των συστημάτων γίνεται εντός του δικτύου μας,χωρίς την χρήση cloud ή server υπηρεσιών που βασίζονται στην σύνδεση του διαδικτύου.

3.1 Σενάριο για συνεργασία προγραμμάτων

Για την υλοποίηση του προσωπικού ψηφιακού βοηθού που αναπτύξαμε χρησιμοποιήσαμε το λογισμικό ανοιχτού κώδικα Rhasspy. Πρόκειται για ένα λογισμικό το οποίο παρέχει ένα ευρύ φάσμα υπηρεσιών απαραίτητων για την υλοποίηση ενός voice assistant χωρίς να χρειάζεται σύνδεση στο διαδίκτυο,ενώ υποστηρίζει πολλές φυσικές γλώσσες.

Το λειτουργήσαμε σε συνδυασμό με τις παρακάτω υπηρεσίες

- Node-RED
- MQTT

- HTTP
- Websockets

Οι προγραμματιστικές του λειτουργίες γίνονται μέσω του node-red ενώ η επικοινωνία των έξυπνων συσκευών γίνεται μέσω ενός mosquitto broker.

Η εγκατάσταση των προαναφερθέντων λογισμικών έγινε μέσω του docker [21].Το Docker είναι μια πλατφόρμα λογισμικού ανοιχτού κώδικα με την οποία μπορούμε να κάνουμε virtualization,δηλαδή εικονικοποίηση πλατφόρμας σε επίπεδο λειτουργικού συστήματος,στην ουσία τρέχουμε το container για την κάθε υπηρεσία που θέλουμε χωρίς να γίνει εγκατάσταση του προγράμματος κατευθείαν στον επεξεργαστή, αλλά με την λήψη μίας εικόνας-image ο υπολογιστής έχει τις απαραίτητες εξαρτήσεις και τα πακέτα κώδικα για να τρέξει την εφαρμογή γρήγορα και αξιόπιστα σε οποιοδήποτε περιβάλλον υπολογιστή.Η φιλοσοφία των container στοχεύει στην καλή εξοικονόμηση των υπολογιστικών πόρων.

- **Node-Red**

  Πρόκειται για ένα προγραμματιστικό εργαλείο διασύνδεσης συσκευών IoT,API και υπηρεσιών νέφους όπου λειτουργεί σε γραφικό περιβάλλον.Βασίζεται σε browser based γραφικό περιβάλλον εργασίας μέσω εντολών και διασυνδέσεων (flows), ακολουθόντας τη φιλοσοφία του event-driven πργραμματισμού. Χτίστηκε πάνω στο Node.js και χρησιμοποιεί τη Javascript για τον προγραμματισμό συναρτήσεων. Η αρχιτεκτονική του,το καθιστά κατάλληλο για να τρέχει σε μικρής ισχύος hardware όπως το Raspberry Pi,ακόμα και σε cloud. Οι ροές που δημιουργούνται στο Node-Red αποθηκεύονται ως JSON και με αυτό τον τρόπο είναι εύκολο να αντιγραφούν και να διανεμηθούν σε άλλους.

- **Mosquitto broker**

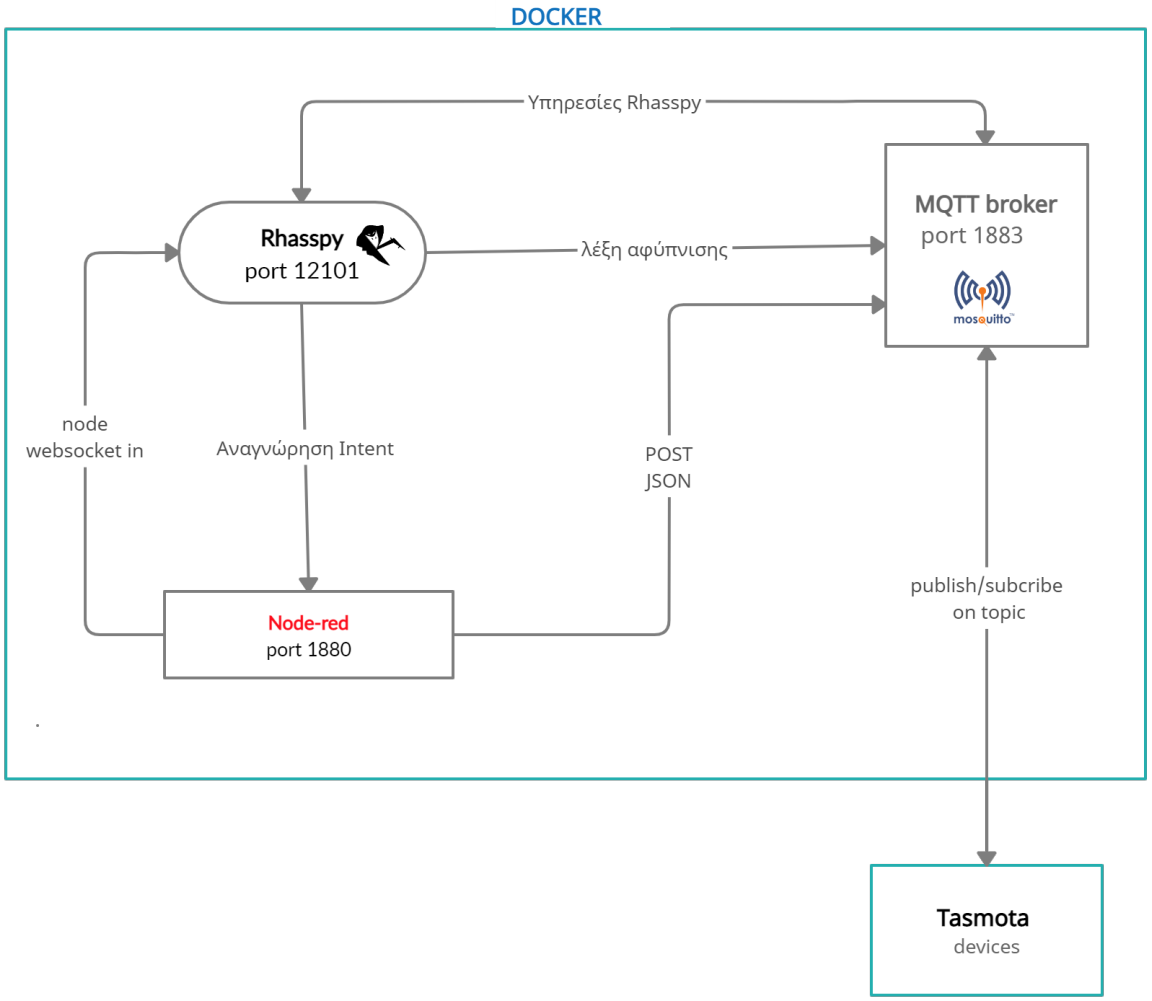
  Για την επικοινωνία μεταξύ των συστημάτων,χρησιμοποιείται ο broker mosquitto,ο οποίος επικοινωνεί μέσω του πρωτόκολλου Mqtt. Πρόκειται για ένα ελαφρύ συμπαγές πρωτόκολλο ανταλλαγής μηνυμάτων για τα IoT-διαδίκτυο των πραγμάτων. Έχει σχεδιαστεί με γνώμονα την ασύγχρονη επικοινωνία και λειτουργεί με τοπολογία server-client. Κύριο ρόλο στην λειτουργία του είναι ο publisher,broker,και subscriber. Publishers είναι οι συσκευές ή τα λογισμικά ακόμα και τα websockets που στέλνουν τις πληροφορίες στον server. brokers ονομάζουμε τους server του συστήματος όπου λαμβάνονται και διαχειρίζονται οι πληροφορίες που στέλνονται από τους publishers.

Οι πληροφορίες λαμβάνονται μέσω των subscribers,όσων είναι εγγεγραμμένοι στον broker. Οι clients του συστήματος οι οποίοι έχουν τον ρόλο των publisher απαιτούν ελάχιστους πόρους ώστε να μπορούν να χρησιμοποιηθούν και σε μικροελεγκτές.

Στο δικό μας παράδειγμα έχουμε συνδέσει στις έξυπνες συσκευές δυο φωτιστικά,το ένα στο σαλόνι και το δεύτερο στο γραφείο.Οι έξυπνες συσκευές οι οποίες είναι εγγεγραμμένες στον broker έχουν διακόπτη ενώ στη μια εξ' αυτών βρίσκεται αισθητήριας θερμοκρασίας και υγρασίας. Ο χρήστης για αρχή θα πρέπει να ''ξυπνήσει'' τον φωνητικό βοηθό με την λέξη *blueberry*. Όταν ακούσει τον χαρακτηριστικό ήχο θα μπορεί να προβεί έπειτα στην εντολή που θέλει να πραγματοποιήσει. Ο χρήστης στο δικό μας σενάριο θα μπορεί να ανοίγει ή να κλίνει τα φώτα που βρίσκονται συνδεδεμένες οι έξυπνες συσκευές,να ενημερώνεται για την θερμοκρασία και υγρασία του χώρου,όπως και για την ώρα. Τέλος έχουμε φωνητική ανταπόκριση από τον βοηθό με την ολοκλήρωση της ενέργειας.Οι εντολές πραγματοποιούνται μέσω της ομιλίας,χρησιμοποιώντας την ελληνική γλώσσα και η απάντηση δίνεται πάλι στα ελληνικά μέσω ενός TextoSpeach εργαλείου.

3.2 Διάγραμμα

Το παρακάτω μπλοκ-διάγραμμα αποτελεί την οπτικοποίηση της επικοινωνίας των διάφορων λογισμικών που χρησιμοποιούμε,ώστε να γίνει καλύτερη η κατανόηση της διαδικασίας. Φαίνεται η περίπτωση που έχουμε μια ολοκλήρωση του σεναρίου.Τα μπλοκ συμβολίζουν τα λογισμικά και οι γραμμές την ενδιάμεση επικοινωνία τους.



εικόνα 3.1: Μπλοκ διάγραμμα του σεναρίου μας

Μέρος ΙΙ

Υλοποίηση

Κεφάλαιο 4

Υλικό-Λογισμικό

Αποτελούμενο απο το raspberry pi και έξυπνες συσκευές της εταιρίας sonoff,το υλικό που χρησιμοποιήσαμε για να στήσουμε το δίκτυο,στο δικό μας σενάριο αναλύεται παρακάτω. Όσο αφορά τα παρελκόμενα του Raspberry Pi τα χρησιμοποιούμε για την καταγραφή και αναπαραγωγή του ήχου,είναι θέμα προτίμησης,παρομοίως και ο αποθηκευτικός χώρος που θα μπορούσε να είναι λιγότερος.Συνιστάται ωστόσο όχι λιγότερο απο 32 GB και τουλάχιστον speed class 10.

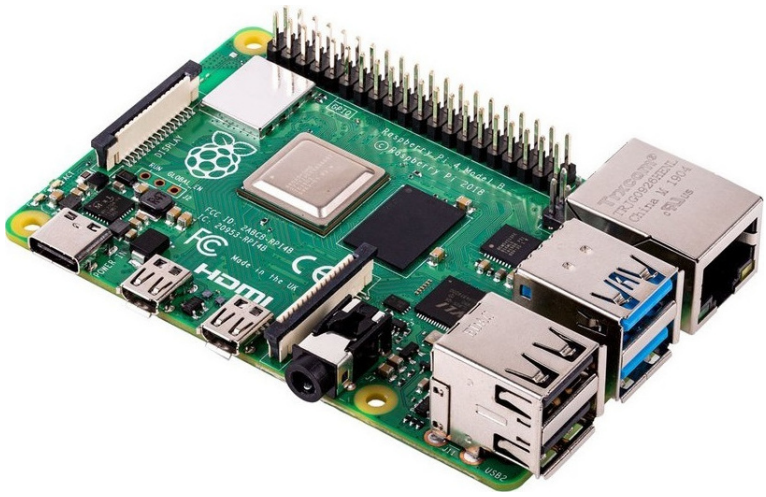
4.1 Raspberry Pi

Το απαραίτητο λογισμικό και υλισμικό που χρησιμοποιήσαμε είναι :

- Raspberry Pi 4 B
- 64 GB micro SD card
- Raspberry Pi OS
- ReSpeaker 2mic HAT
- 3.5 mm speakers

Τα Raspberry Pi μια σειρά μικρών υπολογιστών μονής μητρικής με μικρό όγκο,αναπτύχθηκαν από την Raspberry Pi Foundation σε συνεργασία με το πανεπιστήμιο του Cambridge και κυκλοφόρησαν το πρώτο μοντέλο τον Φεβρουάριο του 2012. Χαρακτηριστικά: [22]

- 4GB LPDDR4-3200 SDRAM
- Broadcom BCM2711, Quad core Cortex-A72 (ARM v8) 64-bit SoC @ 1.5GHz
- 2.4 GHz and 5.0 GHz IEEE 802.11ac wireless, Bluetooth 5.0, BLEGigabit Ethernet
- Gigabit Ethernet
- Raspberry Pi standard 40 pin GPIO header
- 2 micro-HDMI θύρες
- Είσοδος Micro-SD card για φόρτωση λειτουργικού συστήματος και αποθηκευτικού χώρου
- στερεοφωνικό ήχο και μικτή θύρα βίντεο
- είσοδος ρεύματος 5V DC μέσω USB-C



εικόνα 4.1: Raspberry Pi 4

Το λειτουργικό σύστημα Raspberry Pi OS βασίζεται πάνω στην αρχιτεκτονική της Linux διαμονής Debian.Εμείς χρησιμοποιήσαμε την Lite έκδοση που βασίζεται στην αρχιτεκτονική kernell 5.1. [23]

Για την καταγραφή του ήχου χρησιμοποιήσαμε το ReSpeaker 2mic HAT,το οποίο διαθέτει 2 συστοιχίες μικροφώνου και τοποθετείται στο πάνω μέρος απο τις καρφίτσες(40-pin headers) του Raspberry . Η πλακέτα του βασίζεται στον WM8960,ενός χαμηλού κατανάλωσης στερεοφωνικό αποκωδικοποιητή. [24] Για την έξοδο του ήχου διαθέτει ένα 3.5mm

Audio Jack και ένα βύσμα JST2.0.Στην δική μας περίπτωση συνδέσαμε ένα μαγνητικό ηχείο 18mm,ισχύς 0.5W στην θύρα JST

εικόνα 4.2: ReSpeaker 2mic HAT

Ολοκληρώνοντας το Raspberry Pi χρησιμοποιήσαμε μια θήκη για την καλύτερη προστασία και φορητότητα του hardware.

4.2 Έξυπνες Συσκευές

Σε αυτή την ενότητα παραθέσαμε τις συσκευές που χρησιμοποιήσαμε για να στήσουμε το δικό μας IoT δίκτυο.

4.2.1 Sonoff TH10

Πρόκειται για έναν διακόπτη της Sonoff [25] που λειτουργεί μέσω wifi για απομακρυσμένη σύνδεση. Τον χρησιμοποιούμε μαζί με τον αισθητήρα Si7021 ο οποίος συνδέεται στην συσκευή ώστε να έχουμε μετρήσεις θερμοκρασίας και υγρασίας.

εικόνα 4.3: Sonoff TH10 and Si7021 sensor

Χαρακτηριστικά του Sonoff TH10:

- Απομακρυσμένος έλεγχος
- Δυνατότητα μέτρησης Θερμοκρασίας και υγρασίας
- Χρονοδιακόπτης
- Μέγιστο ρεύμα 10 A

4.2.2 Sonoff S26 Wi-Fi Smart Plug

Η S26 είναι μια έξυπνη Wifi Wireless Τηλεχειριζόμενη Πρίζα Ρεύματος 10A και 2200W. Διαθέτει ένα κουμπί on/off στο μπροστά μέρος και πληρεί τα ευρωπαϊκά στάνταρ. [26] Μέσω της σύνδεσης της απομακρυσμένης σύνδεσης υπάρχει δυνατότητα λειτουργίας χρονοδιακόπτη.

εικόνα 4.4: Sonoff s26 smart socket

4.3 ESP8266

Η πλακέτα που χρησιμοποιεί η sonoff και για τις δυο συσκευές είναι η esp8266. [27]

Πρόκειται για ένα χαμηλού κόστους wi-fi μικροτσίπ με υποστήριξη TCP/IP πρωτοκόλλου και δυνατότητες ενός 32-bit μικροελεγκτή,κατασκευασμένος απο τη Espressif Systems.Ο μικροεπεξεργαστής που είναι ενσωματομένος λειτουργεί στα 80 MHz και βασίζεται στον πυρήνα της εταιρίας Tensilica.Λειτουργεί με συνεχές ρεύμα στα 3.3V και για πηγές εισόδου υπάρχουν 16 πινάκια GPIO.

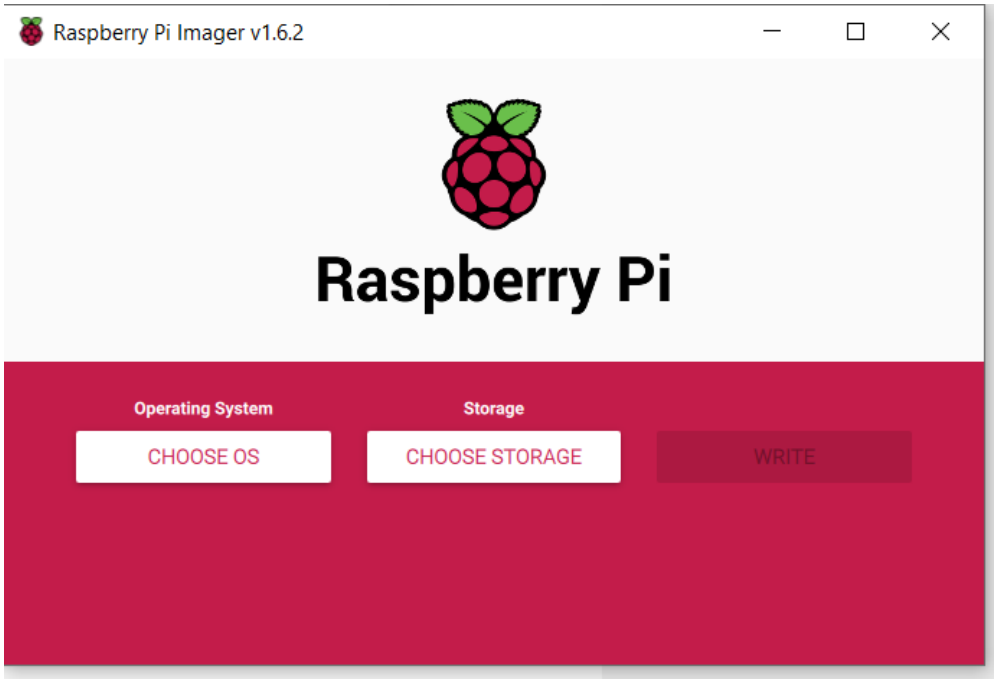
Κεφάλαιο 5

Εγκατάσταση λειτουργικού και λογισμικών

Σε αυτό το κεφάλαιο δίνουμε αναλυτικές οδηγίες,τόσο με εικόνες όσο και με κώδικα για την εγκατάσταση των απαραίτητων προγραμμάτων που τρέχουν στο Raspberry Pi ώστε να το ενσωματώσουμε στο έξυπνο σπίτι τον φωνητικό βοηθό.

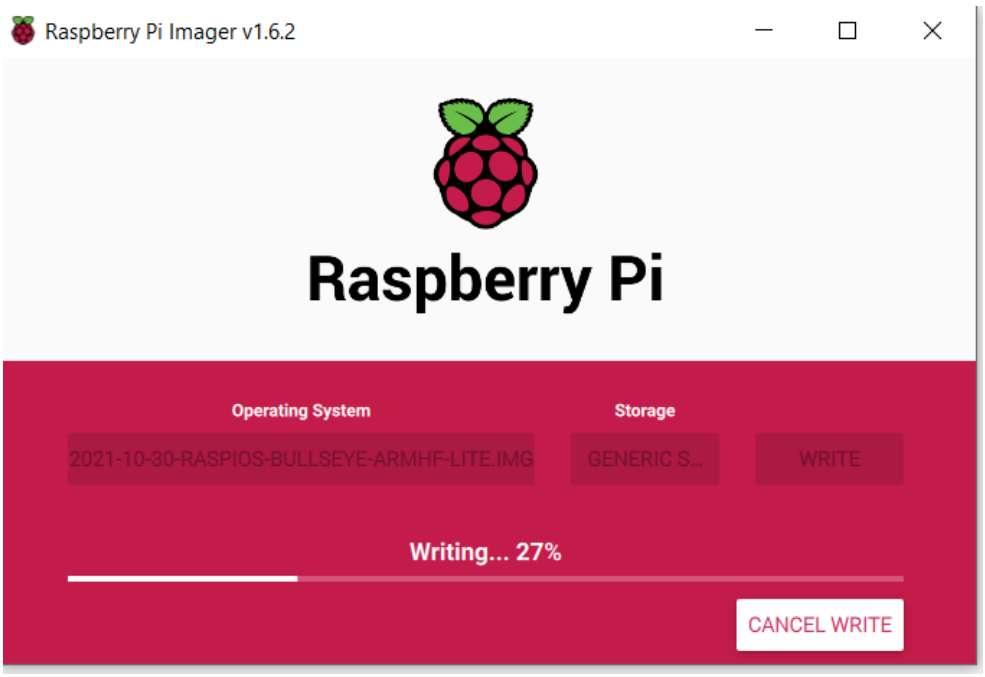
5.1 Εγκατάσταση και ρυθμίσεις λειτουργικού στο Raspberry Pi

Για αρχή θα χρειαστούμε μία micro sd κάρτα όπου θα περάσουμε το λειτουργικό σύστημα που θα τρέχει στο Raspberry.Η micro sd θα πρέπει να έχει το ελάχιστο 10MB/s ταχύτητα εγγραφής. Επισκεφθήκαμε την επίσημη σελίδα του Raspberry [28] και κατεβάσαμε το Raspberry Pi Imager ώστε να εκτελέσει αυτή τη διαδικασία. Σαν λειτουργικό σύστημα επιλέξαμε το Raspberry Pi OS Lite,το οποίο είναι αρκετά μικρό σε μέγεθος και το πιο οικονομικό στους πόρους που καταναλώνει,διότι δεν διαθέτει γραφικό περιβάλλον.Ωστόσο η σύνδεση στην συσκευή μας θα γίνεται μέσω κονσόλας άρα η έλλειψη επιφάνειας εργασίας δεν μας απασχολεί.



εικόνα 5.1: Πρόγραμμα Pi Imager

Πατόντας στο choose OS και στο drop down μενού διαλέξαμε το custom image. Στην αναζήτηση βρήκαμε το λειτουργικό σύστημα που κατεβάσαμε και στην επιλογή choose storage την εξωτερική συσκευή αποθήκευσης που έχουμε συνδεδεμένη.Έπειτα μέσω του write έγινε η εγγραφή.

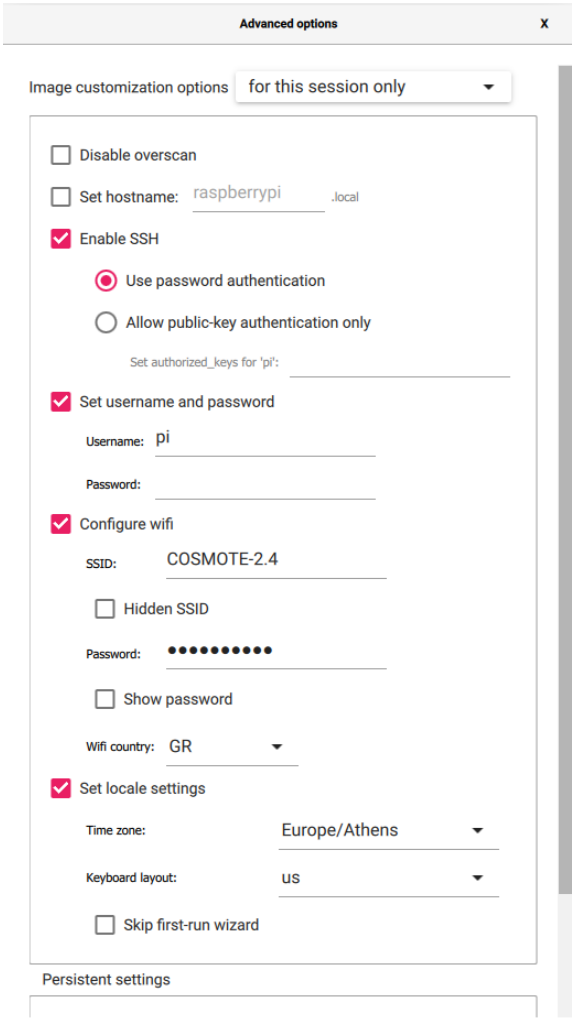


εικόνα 5.2: Πρόγραμμα Pi Imager II

Πριν εισάγουμε την sd κάρτα στο raspberry πρέπει να ενεργοποιήσουμε το SSH και να κάνουμε τις κατάλληλες ρυθμίσεις για την σύνδεση στο δίκτυο του Wi-Fi. Ένας τρόπος είναι να αναζητήσαμε μέσω ενός card reader τα αρχεία στον φάκελο boot και να δημιουργήσουμε ένα νέο αρχείο με όνομα ssh στην πιο πάνω διεύθυνση του φακέλου.Η παρουσία αυτού του αρχείου θα δώσει την εντολή στο Raspberry Pi να ενεργοποιήσει το SSH κατά την εκκίνηση.Για την ενεργοποίση του Wi-Fi η δημιουργία ενός αρχείου κειμένου στο φάκελο boot θα πρέπει να περιέχει τα πσρακάτω στοιχεία:

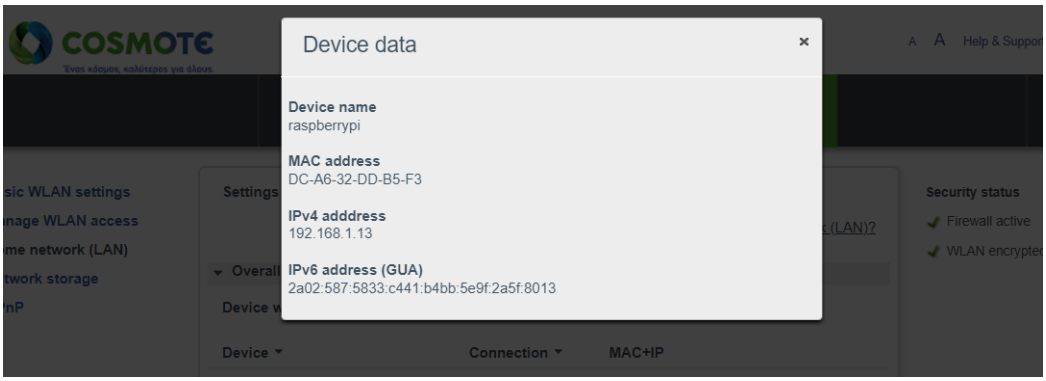
ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev update_config=1 country=GR network= ssid=”MyWiFi” psk=”wi-fi password”

Ωστόσο η ίδια διαδικασία μπορεί να γίνει πιο γρήγορα και πρακτικά μέσω των ρυθμίσεων του Pi Imager,όπως φαίνεται στην παρακάτω εικόνα



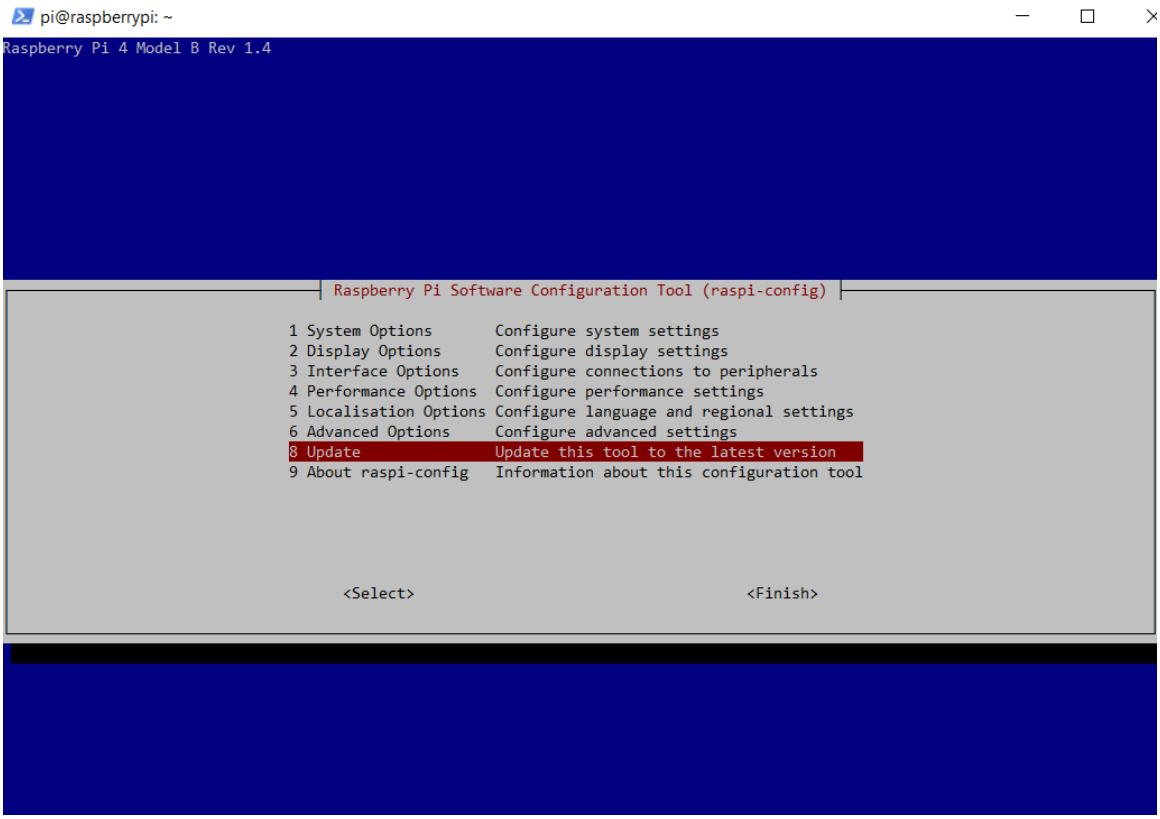
εικόνα 5.3: Ρυθμίσεις Δικτύου

Με αυτό τον τρόπο η συσκευή μας συνδέθηκε στο δίκτυο μας και μπορούμε να βρούμε την IP της,μπαίνοντας στην αρχική σελίδα του router μας ή με την εντολή **ifconfig** εάν έχουμ έξοδο εικόνας στη συσκεύη.



εικόνα 5.4: Ερεύση διεύθυνσης

Το πρώτο εργαλείο που πρέπει να χρησιμοποιήσουμε πριν από όλα είναι το raspi-configuration,ώστε να κάνουμε αναβάθμιση των πακέτων στις τελευταίες εκδόσεις. Πατάμε την εντολή **sudo raspi-config** και έπειτα με το πληκτρολόγιο πατάμε enter στο update.



εικόνα 5.5: Εργαλείο ρυθμίσεων Raspberry

5.1.1 Εγκατάσταση Μικροφώνου

Με την σύνδεση και λειτουργία του Raspberry περνάμε στην εγκατάσταση των οδηγών-drivers. Το πρώτο λογισμικό που χρειαζόμαστε είναι οι drives του μικροφώνου. Όπως αναφέραμε και στο κεφάλαιο του hardware χρησιμοποιήσαμε το ReSpeaker 2-Mics Pi HAT για την καταγραφή του ήχου.

Τα λειτουργικά συστήματα Linux και οι διανομές τους χρησιμοποιούν το framework ALSA,πρόκειται για ένα πλαίσιο λογισμικού που χρησιμοποιεί διεπαφή χρήστη (API) και παρέχει drivers για την κάρτα ήχου. [29]

Η εγκατάσταση των οδηγών έγινε με τις εξής εντολές:

sudo apt-get install –yes git

git clone https://github.com/respeaker/seeed-voicecard

cd seeed-voicecard

sudo ./install.sh

sudo reboot

Μετά την επανεκκίνηση δίνοντας την εντολή **arecord -l** βλέπουμε τις συσκευές εισόδου που είναι διαθέσιμες,εάν αναγνωρίζουμε το μικροφωνό μας τότε η εγκατάσταση έγινε σωστά.

Επίσης ένας πρακτικός τρόπος ώστε να ελέγξουμε αν δουλεύουν σωστά οι συσκευές εισόδου/εξόδου είναι μέσω της παρακάτω εντολής η οποία κάνει stream τον ήχο του μικροφώνου στο ηχείο.

arecord –format=S16_LE –rate=16000 | aplay –format=S16_LE –rate=16000

5.1.2 Εγκατάσταση του λογισμικού Rhasspy

Σε αυτό το κεφάλαιο θα δείξουμε τη διαδικασία που ακολουθήσαμε για την εγκατάσταση του λογισμικού **Rhasspy** στον υπολογιστή Raspberry Pi 4. [30] ΚαΘώς είναι το πρώτο λογισμικό που θα λειτουργούμε σε container κάνουμε πρώτα εγκατάσταση του Docker.

Οι εντολές που χρησιμοποιήσαμε είναι:

curl -sSL https://get.docker.com | sh

Κάναμε εγκατάσταση του docker στο raspberry

sudo usermod -aG docker pi

Θέσαμε τον χρήστη pi σαν admin στο πρόγραμμα docker ώστε να έχουμε δικαιώματα

sudo reboot

κάναμε επαννεκίνηση του συστήματος

docker pull rhasspy/rhasspy

Κατεβάσαμε το image του Rhasspy μέσω του docker.

Έπειτα εκτελέσαμε τις εντολές που φαίνονται στην εικόνα ώστε να τρέξει το image του Rhasspy υπο την μορφή container.

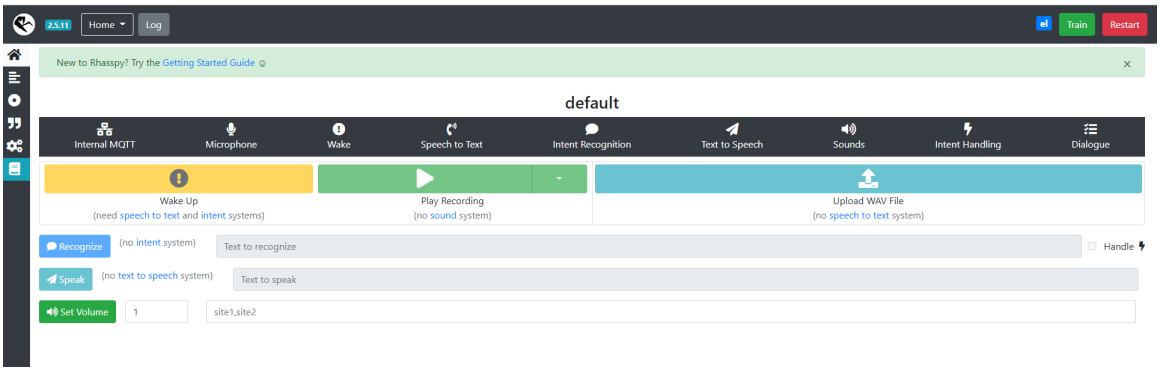
```
pi@raspberrypi:~ $ docker run -d \
>       -p 12101:12101 \
 "/etc/localtime:/etc/localtime:ro" \
      --device /dev/snd:/dev/snd \
      rhasspy/rhasspy \
      --user-profiles /profiles \
      -->       --name rhasspyel \
>       --restart unless-stopped \
>       -v "$HOME/.config/rhasspy/profiles:/profiles" \
>       -v "/etc/localtime:/etc/localtime:ro" \
>       --device /dev/snd:/dev/snd \
>       rhasspy/rhasspy \
>       --user-profiles /profiles \
>       --profile el
```

εικόνα 5.6: Εγκατάσταση Rhasspy

Στις εντολές εγκατάστασης ρυθμίζουμε ώστε το Rhasspy να εκτελείται αυτόματα σε κάθε επανεκκίνηση.Ενώ η τελευταία εντολή καθορίζει το προφιλ,το el αντιστοιχεί στο ελληνικό. Στο τέλος της εγκατάστασης βλέπουμε την διεύθυνση όπου τρέχει η υπηρεσία Rhasspy. Εμφανίζεται ένα μήνυμα σαν το παρακάτω

Running on http://0.0.0.0:12101 (CTRL + C to quit)

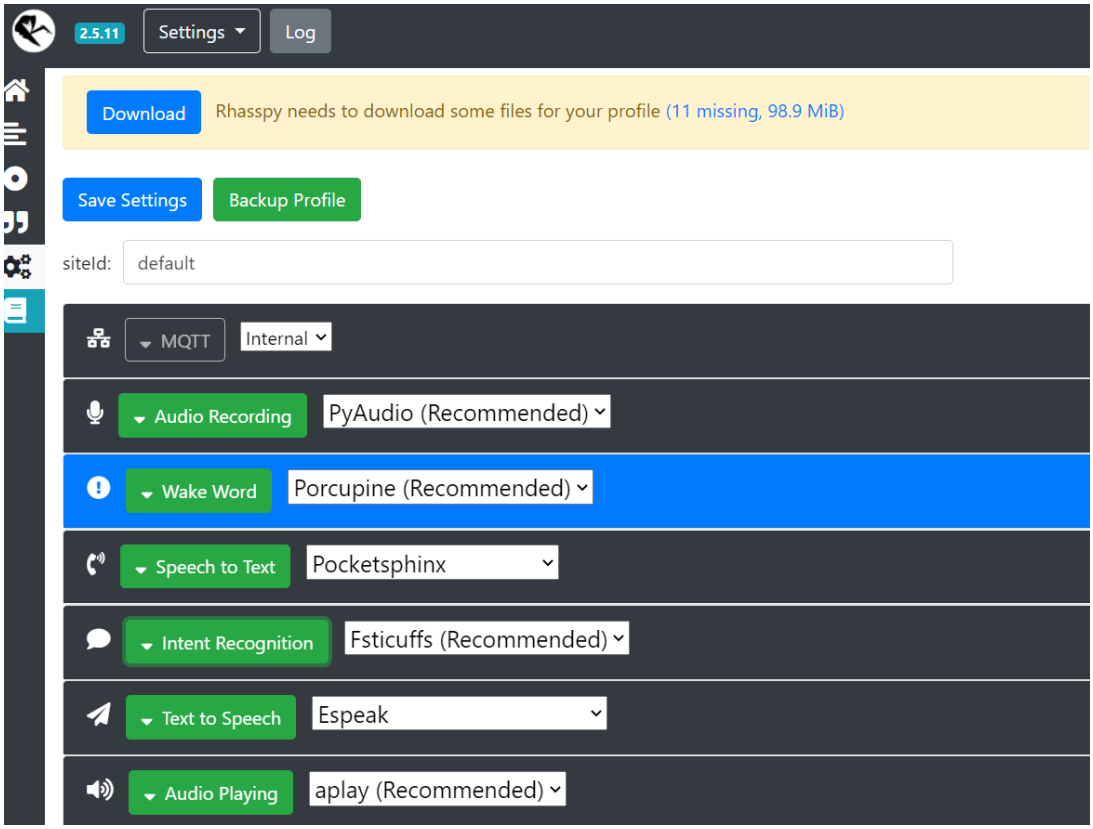
Όταν ολοκληρώσαμε την εγκατάσταση μπορούμε να επισκεφθούμε τον server όπου έχουμε εικόνα της διεπαφής χρήστη.



εικόνα 5.7: Πρώτη διεπαφή του Rhasspy

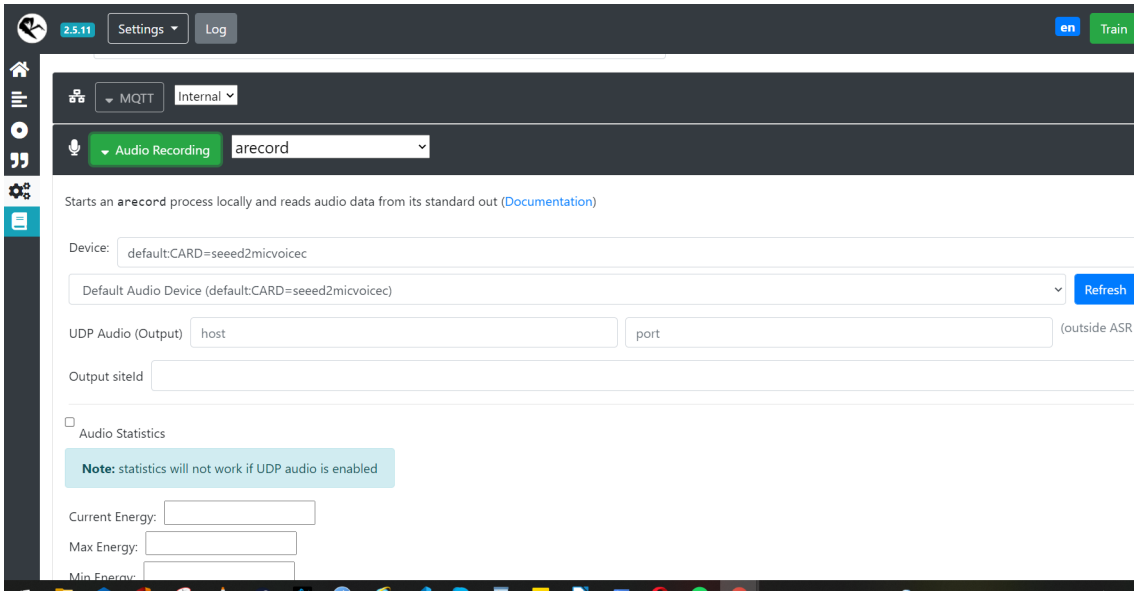
Ξεκινώντας κάναμε τις παρακάτω επιλογές.

Τα εργαλεία που επιλέξαμε για την κάθε υπηρεσία του Rhasspy κατέβηκαν ώστε να προστεθούν στο profile μας. Η ανάλυσή του καθένα από αυτά γίνεται στη συνέχεια.



εικόνα 5.8: Επιλογή Εργαλείων Rhasspy

Αν και προτείνεται η χρήση του PyAudio απο τον οδηγό εγκατάστασης,έχοντας αντιμετωπίσει προβλήματα με την ηχογράφηση αλλάξαμε σε arecord,πατώντας test επιβεβαιώσαμε ότι το μικρόφωνο ακούει.

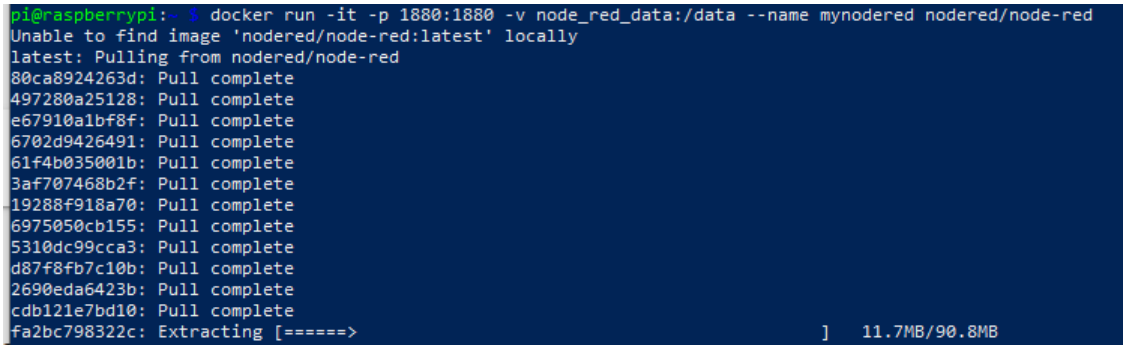


εικόνα 5.9: Επιλογή Εργαλείων Rhasspy II

5.1.3 Εγκατάσταση Node-red και Mosquitto

Με την ολοκλήρωση της εγκατάστασης του Rhasspy,χρησιμοποιήσαμε το node-red σαν περιβάλλον διασύνδεσης των συσκευών μας και σαν API όπου θα επικοινωνεί με το Rhasspy. Για την εγκατάστασή του κάναμε ξανά χρήση του docker ώστε να τρέχει σε container.Σκοπός αυτής της επιλογής είναι να τρέχουν όλα τα προγράμματα μέσω του docker ώστε να έχουμε καλύτερη διαχείριση και προσβασιμότητα στα αρχεία και τις κονσόλες των προγραμμάτων. Η εντολή που χρησιμοποιούμαι για να κατεβάσουμε το container είναι:

```
docker run -it -p 1880:1880 -v
node\_red_data: /data --name mynodered nodered/node-red
```



εικόνα 5.10: Εγκατάσταση node-red

Η θύρα όπου λειτουργεί το Node-Red είναι η 1880 στην τοπική IP του Raspberry Pi.

Το λογισμικό που χρησιμοποιήσαμε ώστε να τρέχει τον mqtt broker που θα συνδέει τις IoT συσκευές με το node-red είναι το eclipse-mosquitto. Πρόκειται για έναν opensource broker ανταλλαγής μηνυμάτων,όπου υλοποιεί το MQTT πρωτόκολλο και είναι διαθέσιμος για όλες τις συσκευές,από χαμηλής ισχύος single board υπολογιστές μέχρι μεγάλους servers. Η εγκατάσταση του έγινε επίσης με την χρήση container,μέσω των εντολών:

docker pull eclipse-mosquitto

ώστε να κατεβάσουμε το image και

docker run -it –name mosquitto -p 1883:1883 eclipse-mosquitto

στην οποία τρέχει στην θύρα 1883 της τοπικής διεύθυνσης του μηχανήματος με το όνομα eclipse

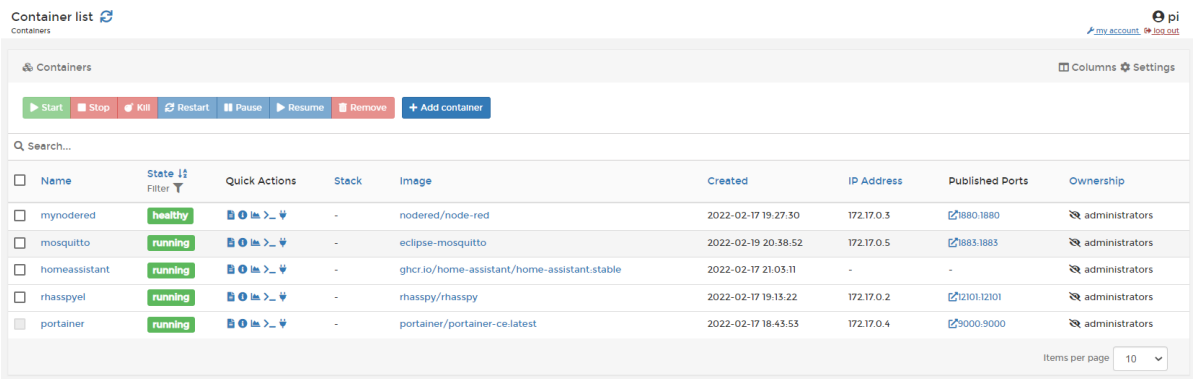
5.1.4 Λειτουργία του Portainer

Έχοντας να διαχειριστούμε πολλαπλά containers μέσω του docker,μας δίνετε η λύση μέσω του portainer,ενός γραφικού περιβάλλοντος διαχείρισης του προγράμματος docker. Με αυτό τον τρόπο μπορούμε να έχουμε μια καλύτερη εικόνα των containers.Μπορούμε να δούμε πότε δημιουργήθηκαν,σε ποία IP address ανήκουν και ποία port χρησιμοποιούν.Παράλληλα μπορούμε να σταματήσουμε ή να επανεκκινήσουμε το κάθε πρόγραμμα χωρίς να γράψουμε κώδικα,αλλά μεσω του UI. Ενώ μας δίνεται και η δυνατότητα πρόσβασης στην κονσόλα του εκάστοτε προγράμματος με πολύ εύκολο τρόπο.

Για να το τρέξουμε κατεβάσαμε πρώτα το αρχείο image μέσω του docker με την εντολή: **sudo docker pull portainer/portainer-ce:latest**

Και έπειτα τρέξαμε το container με την εντολή:

sudo docker run -d -p 9000:9000 –name=portainer –restart=always -v /var/run/docker. sock:/var/run/docker.sock -v portainer_data:/data portainer/portainer-ce:latest



εικόνα 5.11: Portainer UI

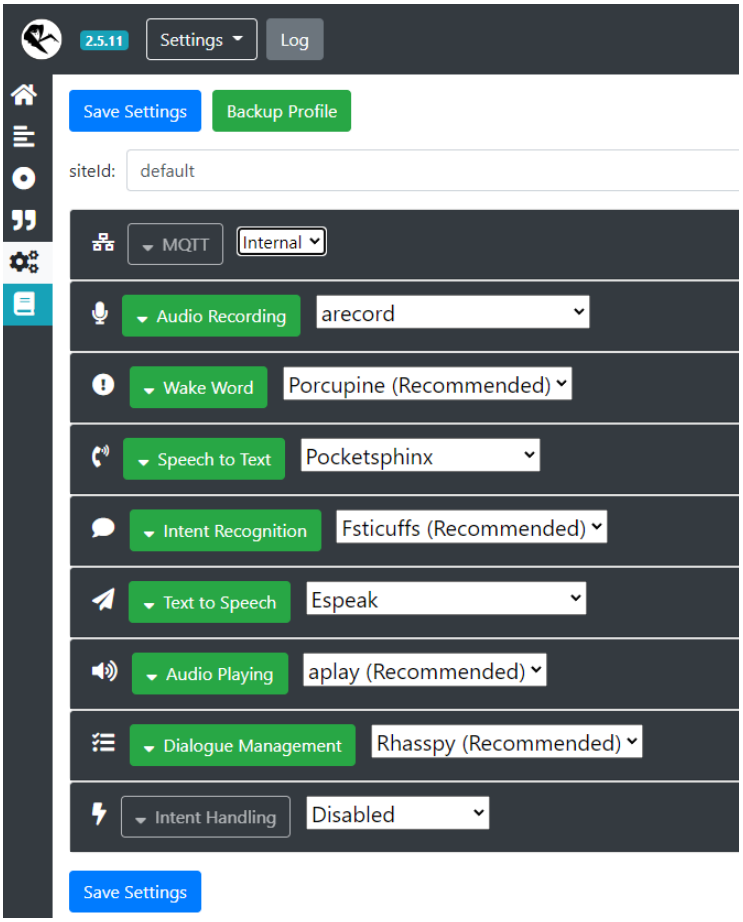
Κεφάλαιο 6

Χρήση και λειτουργία Λογισμικού

Σε αυτό το κεφάλαιο επεξηγούμε τα εργαλεία και τις υπηρεσίες που χρησιμοποιεί το Rhasspy,όπως και τα επιπρόσθετα προγράμματα που είναι απαραίτητα για να λειτουργήσει ο φωνητικός βοηθός εντός του IoT δικτύου.

6.1 Υπηρεσίες του Rhasspy

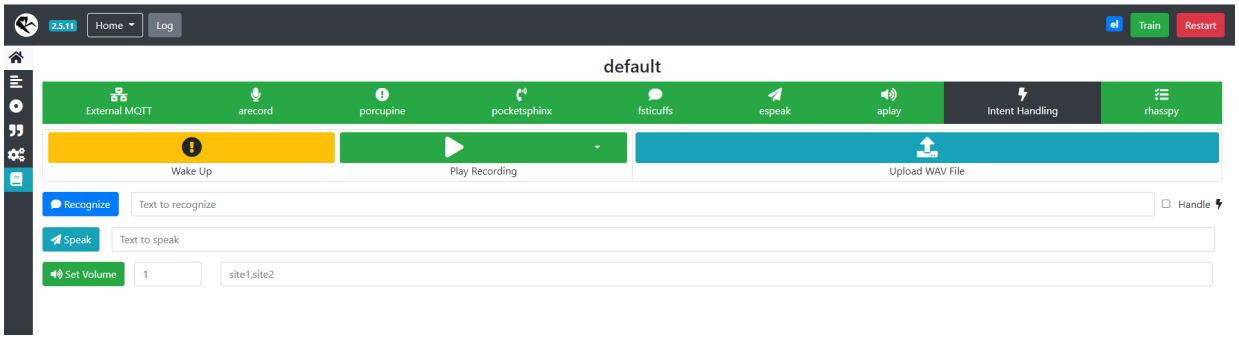
Το Rhasspy απαρτίζεται απο διάφορες,ανεξάρτητες υπηρεσίες οι οποίες επικοινωνούν μεταξύ τους μέσω MQTT,χρησιμοποιώντας ένα υπερσύνολο του πρωτόκολλου Hermes.Οι υπηρεσίες αυτές είναι οι παρακάτω(εικόνα6.1).



εικόνα 6.1: Υπηρεσίες του Rhasspy

6.1.1 Web Server

Αποτελείται από το γραφικό περιβάλλον του Rhasspy στο οποίο ο χρήστης διαχειρίζεται το προφίλ του και τα εργαλεία που χρησιμοποιεί. Ο server βρίσκεται στην θύρα 12101 της αντίστοιχης IP του μηχανήματος στο οποίο τρέχει.



εικόνα 6.2: Αρχική οθόνη Rhasspy

Στην πάνω στήλη της αρχικής σελίδας (εικόνα 6.2) μπορούμε να διακρίνουμε τα εργαλεία που χρησιμοποιεί το Rhasspy.Τα πράσινα είναι όσα λειτουργούν και σε σκούρο χρώμα όσα έχουμε απενεργοποιήμενα. Απο κάτω στο κουμπί wake up μπορούμε να "ξυπνήσουμε" τον φωνητικό βοηθό απλά πατώντας το.Στο Play recording το Rhasspy αναπαράγει την τελευταία ηχογραφημένη φωνητική εντολή,ενώ είναι δυνατή και η λήψη του αρχείου ήχου,υπό μορφή WAV. Στο κουμπί Upload WAV File,μας δίνετε η δυνατότητα ανεβάσματος ενός αρχείου ήχου προς αναγνώριση από το Rhasspy. Συνεχίζοντας προς τα κάτω στο κουτί Recognize μπορούμε να πληκτρολογήσουμε μια πρόταση με σκοπό το Rhasspy να την αναγνωρίσει,στην ουσία αντι για φωνητικές εντολές μπορούμε να πληκτρολογούμε απ' ευθείας εκεί. Στο κουμπί speak μέσω του εργαλείου text to speach που έχουμε ορίσει να χρησιμοποιεί,μπορούμε να γράψουμε στον φωνητικό βοηθό να πει κάτι. Στο set volume ρυθμίζουμε την ένταση των ηχείων.

Αριστερά υπό μορφή στήλης έχουμε

- Sentences
  προτάσεις οι οποίες χρησιμοποιούν μια απλή JSGF γραμματική και χωρίζονται σε ενότητες ή intents.Το αρχείο είναι μορφής .ini
- Slots
  Πίνακες μεταβλητών που μπορούν να χρησιμοποιηθούν στις προτάσεις του Rhasspy
- Words
  Σε αυτή την καρτέλα μπορούμε να αλλάξουμε την εκφώνηση των λέξεων,μέσω της φωνητικής μεταγραφής με βάση το διεθνές φωνητικό αλφάβητο.
- Settings
  Τα κύρια εργαλεία του Rhasspy βρίσκονται σε αυτή την στήλη καθώς και οι ρυθμίσεις τους.

Τα sentences και settings αναλύονται περαιτέρω στη συνέχεια.

6.1.2 Dialogue Manager

Η διεπαφή προγραμματισμού εφαρμογών ή αλλιώς API που είναι υπεύθυνη για την διαχείριση και την επικοινωνία των υπηρεσιών ονομάζεται Hermes. Μέσω της wake word ή όταν ο χρήστης ξεκινήσει χειροκίνητα μια συνεδρία στέλνετε το JSON αρχείο

hermes/dialogueManager/startSession. Ενώ για τις υπόλοιπες υπηρεσίες υπάρχουν αντίστοιχα άλλα JSON όπως το hermes/dialogueManager/sessionEnded όταν κλείνει μια συνεδρία και το hermes/dialogueManager/intentNotRecognized όταν αποτυγχάνει η αναγνώριση ενός intent

6.1.3 Audio Input

Η υπηρεσία που δέχεται τον ήχο από την συσκευή του μικροφώνου και μέσω του πρωτοκόλλου hermes τα στέλνει στον MQTT Broker με το topic hermes/audioServer/<siteId>/audioFrame. Τα αρχεία του ήχου έχουν μορφή WAV και θα πρέπει να είναι 16-bit 16Khz mono.

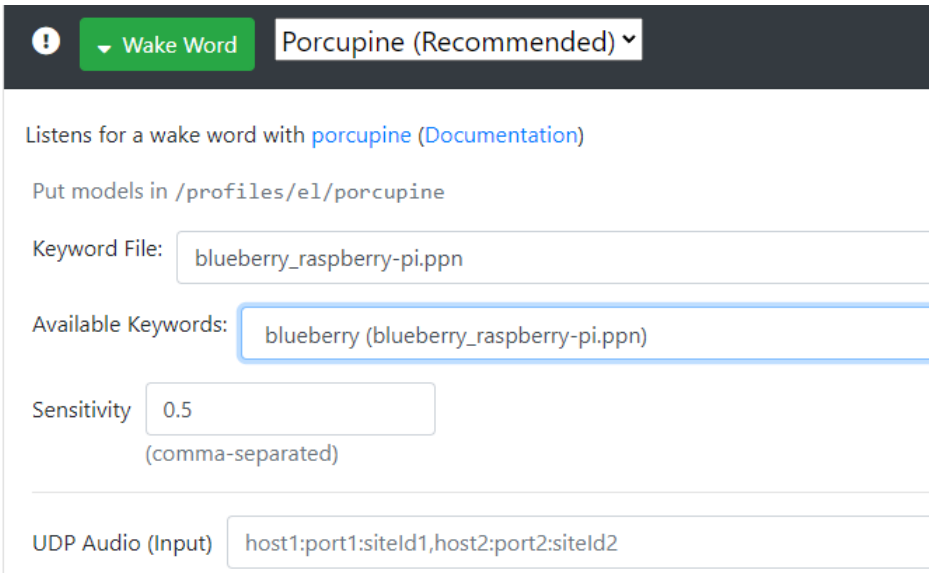
6.1.4 Wake Word Detection

Η πρώτη αλληλεπίδραση του χρήστη με τον φωνητικό βοηθό ξεκινάει μέσω της λέξης αφύπνισης.Όταν τα αρχεία τύπου WAV που λαμβάνονται μέσω του Audio Input περιέχουν την λέξη αφύπνισης τότε στέλνετε στον MQTT Broker το topic hermes/hotword/<wakewordId>/detected, όπου στη συνέχεια ο broker απαντάει με το topic hermes/dialogueManager/startSession για να ξεκινήσει η συνεδρίαση. Το Rhasspy μπορεί να χρησιμοποιήσει διαφορετικά συστήματα για την αναγνώριση wake word μέσω της φυσικής γλώσσας.Συγκεκριμένα είναι τα:

- Raven
- Porcupine
- Snowboy
- Mycroft Precise
- Pocketsphinx

Ενώ είναι δυνατή και η επιλογή external command,δηλαδή τρίτου custom προγράμματος το οποίο χρησιμεύει για τον εντοπισμό της wake word. Απο τις παραπάνω επιλογές δοκιμάσαμε την pocketshinx με την ελληνική γλώσσα,αν και γινόταν η αναγνώριση,υπήρχε συνεχή ενεργοποιήση του φωνητικού βοηθού ενώ άκουγε άκυρες λέξεις. Για αυτό το λόγω απορρίφθηκε και χρησιμοποιήσαμε την καλύτερη δυνατή λύση, το Porcupine καθώς έχει εξαιρετική απόδοση,μπορεί να παραμετροποιηθεί και δεν είναι απαραίτητη η online εγγραφή,δυστηχώς δεν

αναγνωρίζει τα ελληνικά. Πέρα απο τις φωνητικές εντολές μπορούμε να ενεργοποιήσουμε την αλληλεπίδραση με το Rhasspy κάνοντας POST στο topic /api/listen-for-command μέσω του HTTP API. Πρακτικά αυτό μπορεί να γίνει πατώντας το κουμπί Wake Up στον GUI του Rhasspy. Εμείς επιλέξαμε την λέξη blueberry από την λίστα των έτοιμων λέξεων.



εικόνα 6.3: επιλογή λέξης αφύπνισης

6.1.5 Speech to Text

Αυτή η υπηρεσία είναι υπεύθυνη για την μετατροπή της φυσικής γλώσσας σε κείμενο.Αν και η μεταγραφή των αρχείων WAV είναι το πρώτο βήμα η κύρια λειτουργία είναι η μετατροπή αυτών των αρχείων σε JSON events ώστε να τρέξουν στα intents του node-red. Τα διαθέσιμα συστήματα που χρησιμοποιεί το Rhasspy για αυτή την διαδικασία είναι τα:

- **Pocketsphinx** της CMU

  Λειτουργεί τελείως offline και στις υποστηριζόμενες γλώσσες είναι το μόνο εργαλείο που έχει διαθέσιμη και την ελληνική.

- **Kaldi**

  Λειτουργεί τελείως offline και ο συγκεκριμένος,ωστόσο χρειάζεται εγκατάσταση και αλλαγή του path στις προεπιλεγμένες ρυθμίσεις του Rhasspy για να λειτουργήσει εκτός Docker.

- **DeepSpeech**

  Η αυτόματη μηχανή αναγνώρισης φωνής (ASR-automatic speech recognition) της

Mozilla λειτουργεί μέσω machine learnning. Βασίζεται πάνω στο Tensorflow της Google ώστε να αξιοποιεί λιγότερους πόρους και να μειώσει το boot time κάνοντας την αναγώριση φωνής πιο εύκολη.

- **Remote HTTP Server** Χρησιμοποιούμε ένα απομακρυσμένο instance του Rhasspy για αναγνώριση φωνής.Αξιοποιώντας ένα σύστημα client/server με ικανοποιητική μνήμη και επεξεργαστική δύναμη σαν home server.

Υπάρχουν ποικίλες παράμετροι που επηρεάζουν το πως αντιλαμβάνεται τις φωνητικές εντολές το Rhasspy και ρυθμίζονται μέσω του profile χρήστη. Όπως είναι τα πόσα δευτερόλεπτα διαρκεί η ελάχιστη φωνητική εντολή,το πότε αναγνωρίζει τη σιωπή,ώστε να σταματήσει η ηχογράφηση,η ευαισθησία του περιβάλλοντα ήχου και άλλα.

6.1.6 Intent Recognition

Με την μετατροπή της ομιλίας σε κείμενο,το Intent Recognition αναγνωρίζει τις κατάλληλες λέξεις ώστε να τρέξει τα JSON events στο Node-Red Τα διαθέσιμα συστήματα που αναγνωρίζουν αυτά τα intents είναι:

- **Fsticuffs**

  Χρησιμοποιεί την rhasspy-nul βιβλιοθήκη και αναγνωρίζει μόνο προτάσεις όπου έχει εκπαιδευτεί το Rhasspy.Μπορεί να αναγνωρίσει εκατομμύρια προτάσεις σε χιλιοστά του δευτερολέπτου και διαθέτει γρήγορη ταχύτητα εκπαίδευσης.Στη στήλη sentences του profile μας υπάρχουν ήδη κάποια διαθέσιμα intents,ωστόσο προσθέσαμε νέα που μας βοηθούν στην υλοποίηση του σεναρίου μας. Για την χρήση φωνητικών εντολών μέσω του profile μας,το fsticuffs συνιστάται ως καταλληλότερη επιλογή.Επίσης διαθέτει λειτουργία για την καλύτερη αντιστοίχηση των λέξεων.

- **Fuzzywuzzy**

  Χρησιμοποιεί την rapidfuzz βιβλιοθήκη για να ταιριάξει intents ανάμεσα στο κείμενο και στις προτάσεις που του έχουμε προσθέσει στο προφίλ.

- **Snips NLU**

  Χρησιμοποιεί την Snips Python library για την αναγνώριση προτάσεων σε ποικίλες γλώσσες

- **RasaNLU**

  Πρόκειται για μια υπηρεσία που χρησιμοποιεί απομακρυσμένο server για να λειτουργήσει μέσω του Rhasspy.Μπορεί να γίνει εγκατάσταση και να τρέξει μέσω του docker,μόνο σε Linux/amd64 αρχιτεκτονική.Η θύρα που είναι διαθέσιμη είναι η 5005.

- **Mycroft Adapt**

  Πραγματοποιεί αναγνώριση intents μέσω του Mycroft Adapt,δουλεύει καλύτερα όταν έχουμε έναν μέτριο αριθμό προτάσεων,αλλά δεν υποστηρίζεται ακόμα στην έκδοση 2.5

- **Flair**

  Πραγματοποιεί αναγνώριση intents χρησιμοποιώντας το flair NLP framework,δουλεύει καλύτερα όταν έχουμε έναν μεγάλο αριθμό προτάσεων,αλλά δεν υποστηρίζεται ούτε το συγκεκριμένο ακόμα στην έκδοση 2.5

- **Remote HTTP Server**

  Χρησιμοποιεί έναν απομακρυσμένο server του Rhasspy για αναγνώριση των intent.Γίνεται POST του κειμένου σε ένα HTTP endpoint και λαμβάνει ένα JSON intent.

- **Local Command**

  Καλεί εξωτερικό πρόγραμμα το οποίο λαμβάνει ως είσοδο απλό κείμενο και εξάγει JSON intent στην έξοδο.

- **Hermes MQTT**

  Μέσω εξωτερικής υπηρεσίας χρησιμοποιούμε το Hermes Protocol για την αναγνώριση intents.

6.1.7 Intent Handling

Μέσω του Intent Handling μας δίνεται η δυνατότητα να στέλνουμε τα Intents μας σε κάποιο άλλο σύστημα όπως το Home Assistant ή ακόμα και σε άλλο server του Rhasspy. Σαν προεπιλογή είναι απενεργοποιημένη αυτή η υπηρεσία.

6.1.8 Text to Speech

Για την μετατροπή του κειμένου σε φωνή το Rhasspy έχει έναν αρκετά μεγάλο κατάλογο εργαλείων που υποστηρίζουν αρκετές φυσικές γλώσσες.

- **espeak**

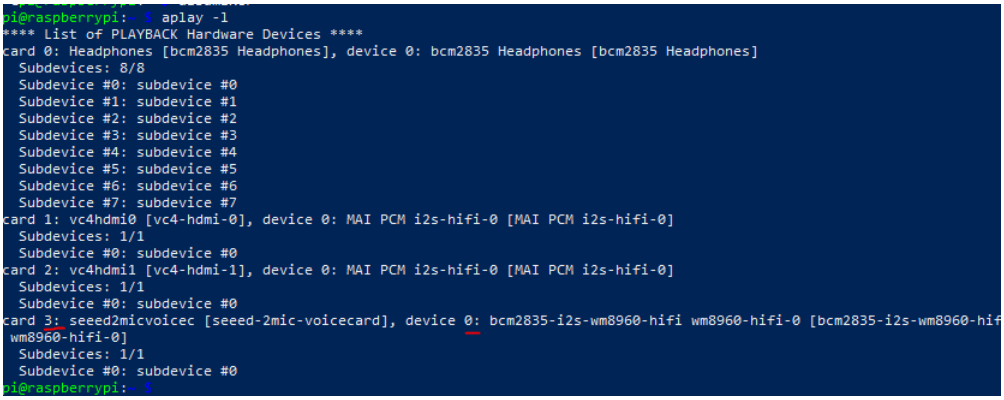
  Το espeak αποτελεί την προεπιλεγμένη επιλογή,καθώς είναι συμβατό με πάνω απο 40 γλώσσες περιλαμβανομένου και της ελληνικής και αρχαίας ελληνικής.Ο ήχος που εξάγει είναι τύπου WAV και ολόκληρο το software του ανοιχτού κώδικα έρχεται σε πολύ μικρό μέγεθος,βασισμένο πάνω στην γλώσσα C. Από τις ρυθμίσεις του προφίλ μπορούμε να ρυθμίσουμε και την ένταση του ήχου.

- **OpenTTS**

  Το άλλο εργαλείο που αξίζει να ασχοληθούμε είναι το OpenTTS καθώς υποστηρίζει και αυτό την ελληνική γλώσσα.Λειτουργεί μέσω web server και για την εγκατάσταση του θα πρέπει να τρέξουμε μέσω του docker.Αφού το εγκαταστήσουμε,λειτουργεί μέσω της θύρας 5500 στην IP του rhaspberry.Μέσω του portainer μπορούμε να το τρέξουμε χωρίς να εκτελέσουμε την εντολή στο terminal. Ωστόσο κατά την λειτουργία του αντιμετωπίσαμε προβλήματα.Τόσο στην αναγνώριση των προτάσεων της ελληνικής γλώσσας,όσο και στην απευθείας σύνδεση με το Rhasspy.

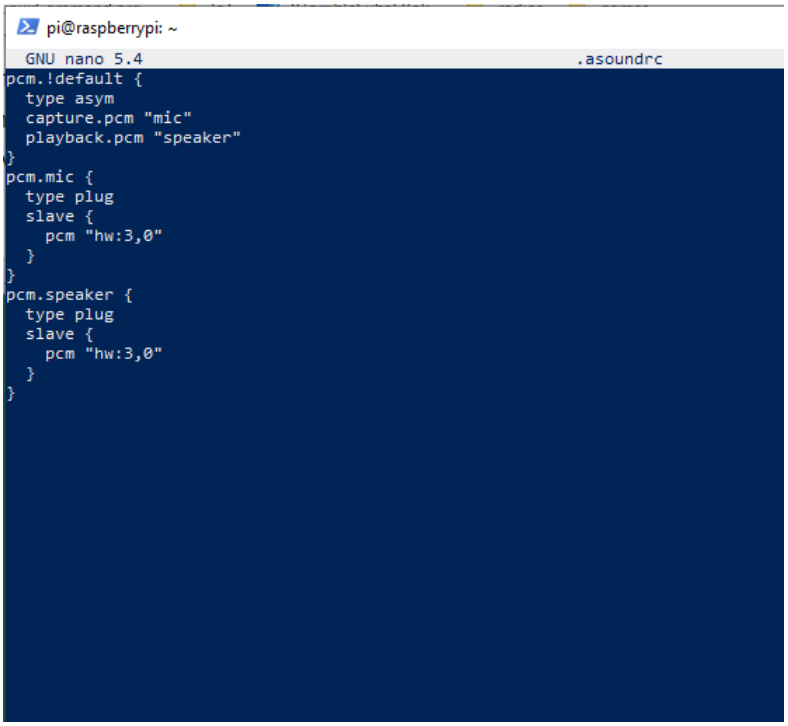
6.1.9 Audio Playing

Για την εξαγωγή του ήχου χρησιμοποιούμε το aplay,το λογισμικό του raspberry που είναι υπεύθυνο για την αναπαραγωγή του ήχου.Με την εντολή **aplay -l** βλέπουμε τις διαθέσιμες συσκευές που αναγνωρίζει η κάρτα ήχου.Για αυτή που θέλουμε να χρησιμοποιήσουμε συγκρατούμαι το card number και το device number.



εικόνα 6.4: aplay

Έπειτα παραμετροποιούμε το αρχείο .asoundrc ώστε να έχει τους σωστούς αριθμούς κάτω απο την στήλη speaker. Στην δική μας περίπτωση έχουμε αυτή την εικόνα.



εικόνα 6.5: asoundrc.

Είναι απαραίτητο να κάνουμε και τη σωστή επιλογή στο UI του Rhasspy στην στήλη Available Devices.

Sounds

Στην καρτέλα Sounds μπορούμε να δούμε τα αρχεία WAV που αναπαράγονται όταν το Rhasspy ”ξυπνάει”,όταν ολοκληρώνεται η ηχογράφηση και όταν έχουμε σφάλμα και δεν

γίνεται αναγνώριση της φωνής.

6.1.10 MQTT

Στο επίπεδο της εφαρμογής, το πρωτόκολλο MQTT λειτουργεί πάνω από το πρωτόκολλο TCP/IP του φυσικού επιπέδου του wi-fi . Κάποια από τα πλεονεκτήματά του είναι ότι αποτελεί ελαφρύ και αποδοτικό πρωτόκολλο ενώ χρειάζεται ελάχιστους πόρους για να λειτουργήσει το οποίο το καθιστά ιδανικό και για μικροελεκτές.Παρέχει αξιοπιστία στην παράδοση μηνυμάτων για αυτό το λόγο έχει 3 επίπεδα στο QoS.Έχει δυνατότητα κρυπτογράφησης μηνυμάτων χρησιμοποιώντας TLS [31] και πιστοποίηση client μέσω μοντέρνων πρωτοκόλλων και μπορεί να επεκταθεί σε χιλιάδες IoT συσκευές [32]
Χαρακτηριστικά:

- Ασύγχρονο πρωτόκολλο
- Συμπαγή μηνύματα
- Λειτουργία σε συνθήκες ασταθούς σύνδεσης της γραμμής μετάδοσης των δεδομένων
- Υποστήριξη διαφόρων επιπέδων ποιότητας υπηρεσιών (QoS)
- Εύκολη ενσωμάτωση νέων συσκευών

Στο Rhasspy έχουμε την επιλογή ανάμεσα σε δύο λειτουργίες MQTT.Την εσωτερική και την εξωτερική.Στην ουσία πρόκειται για δύο διαφορετικούς broker.

Internal MQTT

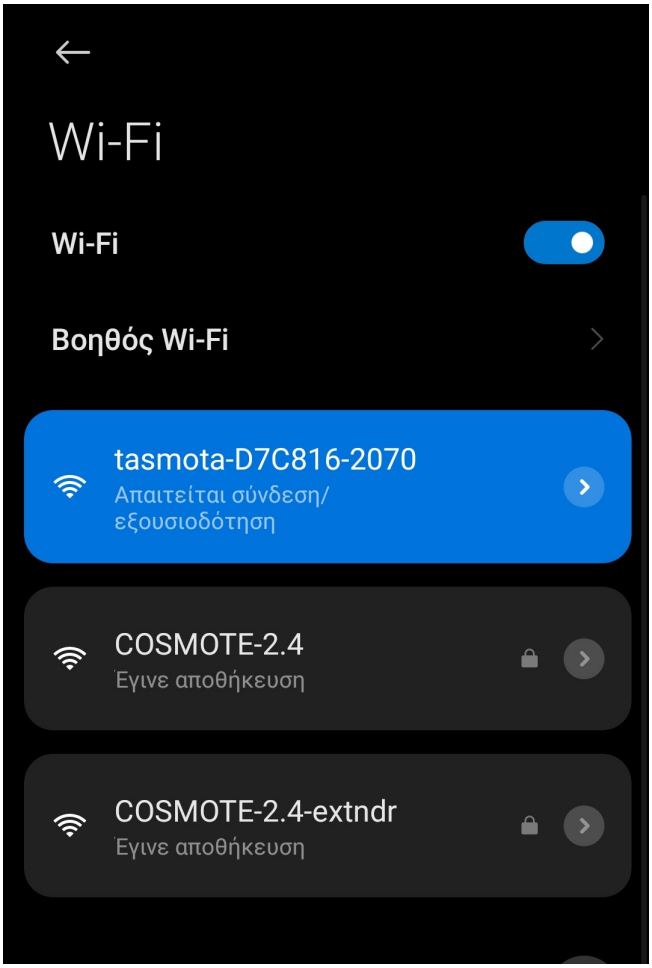
Το Rhasspy είναι ρυθμισμένο αυτόματα στο internal-εσωτερικό MQTT. Με την εκκίνηση της εφαρμογής ένας mqtt broker [33] ξεκινάει αυτόματα στην θύρα 12183.Όλες οι υπηρεσίες του Rhasspy συνδέονται και περνάνε τα μηνύματα μέσω του broker. Τα αρχεία του ήχου που δέχεται σαν είσοδο το Rhasspy είναι τύπου WAV τα οποία μεταφέρονται στον MQTT Broker.Εάν εντοπιστεί η λέξη αφύπνισης τότε ξεκινάει η διαδικασία αλληλεπίδρασης.Όλες οι υπηρεσίες του Rhasspy είναι συνδεδεμένες μέσα από αυτό τον broker.

External MQTT

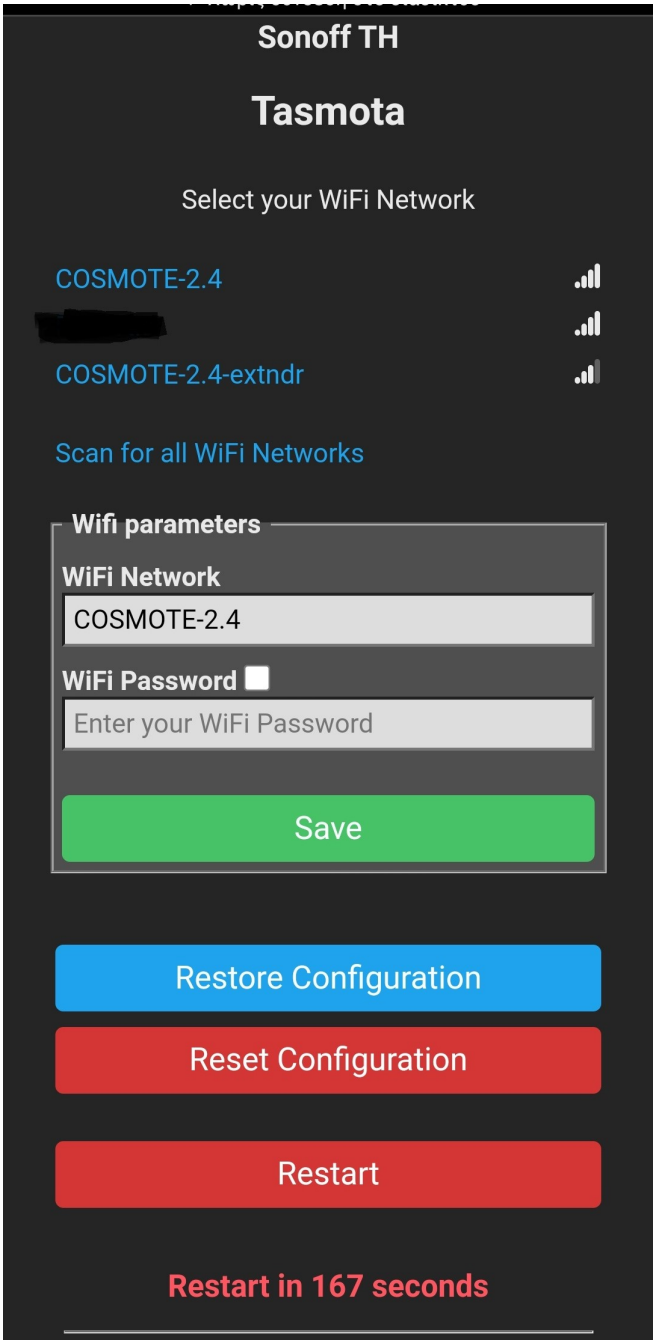
Το Rhasspy μπορεί να συνδεθεί και με έναν εξωτερικό broker,συμπληρώνοντας στο πεδίο external mqtt την Ip και την port όπου τρέχει ο server.Στην δική μας περίπτωση χρησιμοποιήσαμε το πρόγραμμα mosquitto,άρα βάζουμε την local ip και για port το 1883.

6.2 Tasmota

Για τις συσκευές που εντάξαμε στο IoT δίκτυο μας εγκαταστήσαμε το firmware Tasmota μέσω του εργαλείου Tasmotizer. Τα firmware αποτελούν λογισμικά που εγκαθίστανται μόνιμα σε μια read-only μνήμη. Μέσω του Tasmota μπορούμε να έχουμε έλεγχο και να κάνουμε χρήση MQTT, Web UI και HTTP σε συσκευές που χρησιμοποιούν το ESP chipset,ενώ είναι δυνατή και η αυτοματοποίηση χρησιμοποιώντας χρονοδιακόπτες,κανόνες και σενάρια.Για την ένταξη των συσκευών στο mqtt πρέπει να συνδεθούν εξαρχής στο τοπικό δίκτυο και μέσω της Ip τους να γίνουν ανιχνεύσιμες στον mqtt broker. Αρχικά οι συσκευές εκπέμπουν το δικό τους Wi-Fi σήμα,συνδεόμαστε στο δίκτυο τους ασύρματα(εικόνα6.6) και από εκεί πηγαίνοντας στις ρυθμίσεις του wi-fi και συμπληρώνοντας τα διαπιστευτήρια τις συνδέουμε στο οικιακό δίκτυο.(εικόνα6.7)

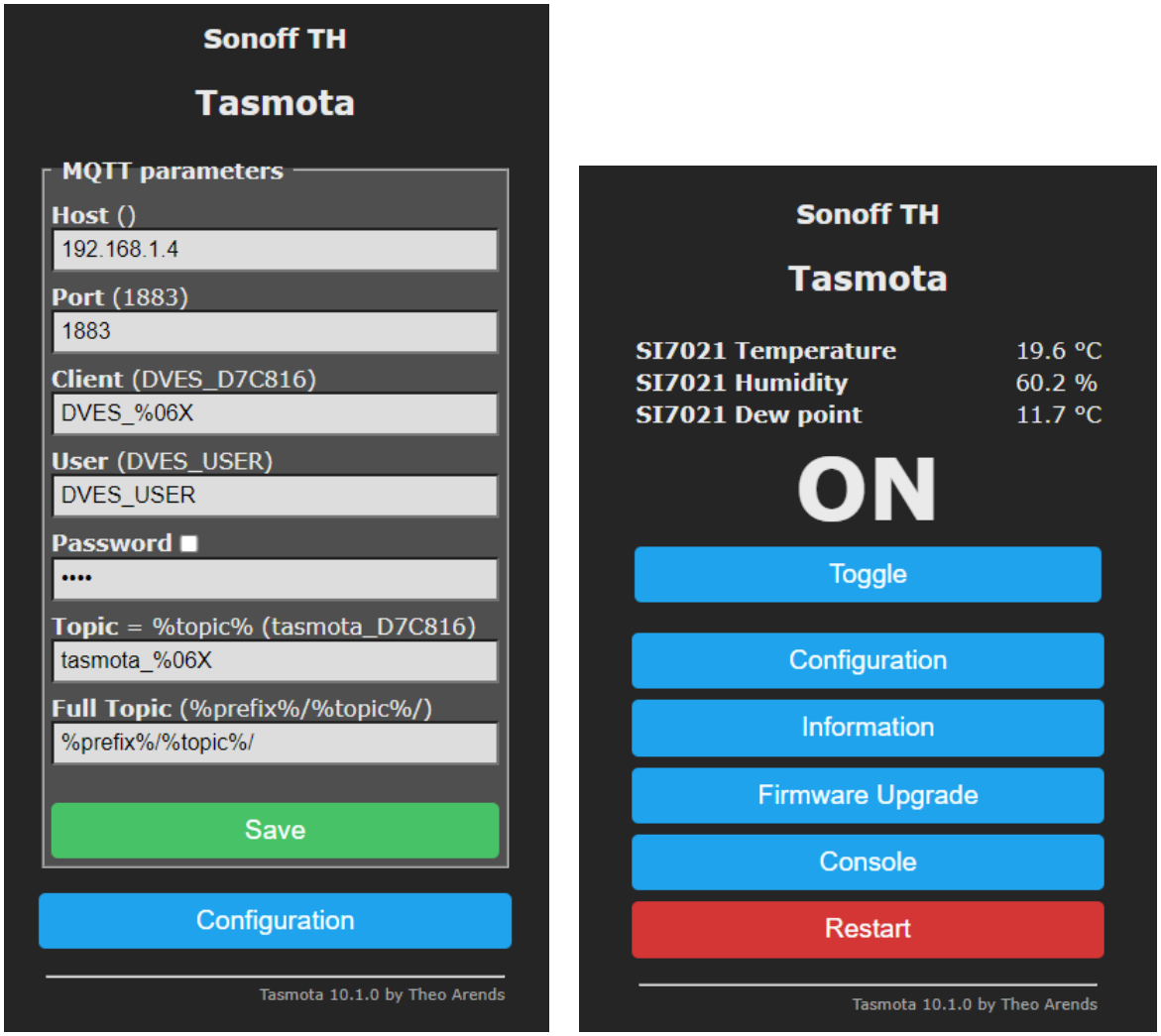


εικόνα 6.6: scanning for tasmota



εικόνα 6.7: wi-fi tasmota configuration

Έπειτα στο mqtt configuration,συμπληρώνουμε τις παραμέτρους. Στον host γράψαμε την Ip του raspberry και στο port την θύρα όπου τρέχει ο mosquitto broker,στην δική μας περίπτωση βρίσκεται στην θύρα 1883.Τα υπόλοιπα τα αφήνουμε όπως είναι και προχωράμε στην αποθήκευση και επανεκκίνηση της συσκευής.6.8(α□) Η κεντρική οθόνη της συσκευής Sonoff TH10 μαζί με τις μετρήσεις του αισθητήρα Si7021, βρίσκεται στην παρακάτω εικόνα 6.8(β□)



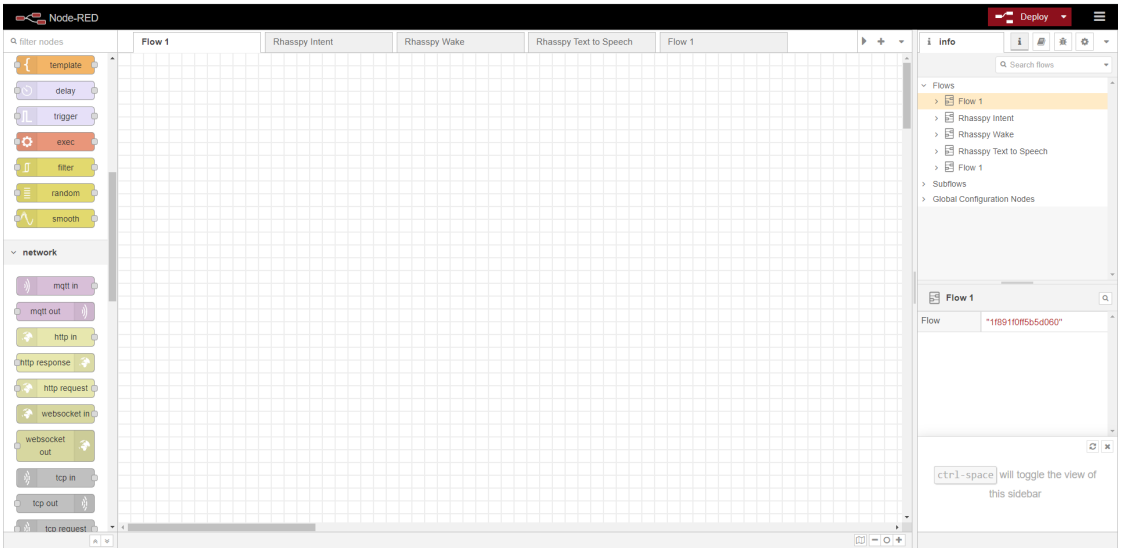
(α□) mqtt tasmota configuration (β□) tasmota UI

εικόνα 6.8: Tasmota

Παρατηρήσαμε ότι η τιμές της θερμοκρασίας και υγρασίας στέλνονται μέσω publish στον MQTT Broker ανα πέντε λεπτά,θέλοντας να μειώσουμε τον χρόνο ανατρέχουμε στην κονσόλα της συσκευής και πληκτρολογούμε την εντολή **teleperiod 60** με την οποία ορίζουμε τον ρυθμό ανανέωσης στο topic **tele/tasmota_D7C816/SENSOR** στα 60 δευτερόλεπτα. Ενώ πρέπει να ανφέρουμε ότι σε περίπτωση αποσύνδεσης ή μη εύρεσης της συσκευής στο τοπικό δίκτυο,πρέπει να γίνει reset,η επαναφορά των ρυθμίσεων απαιτεί έξι γρήγορες πιέσεις του power button.Την ίδια διαδικασία ακολουθήσαμε και για τη συσκευή Sonoff s26.

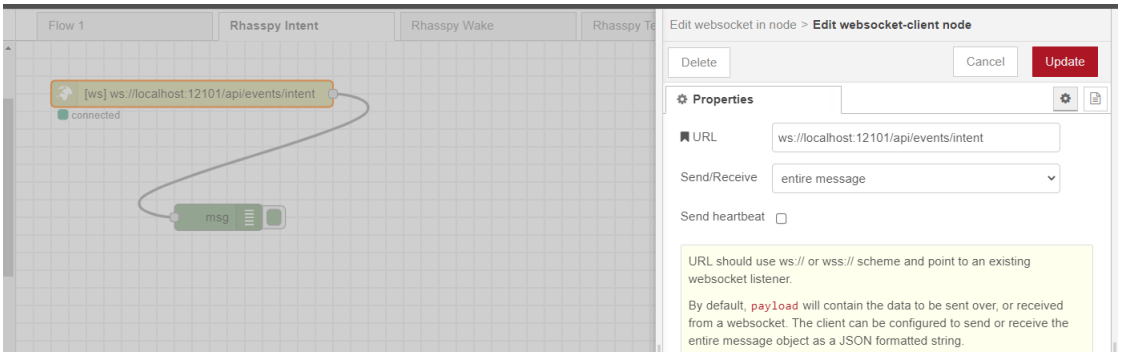
6.3 Node-Red

Η χρήση του node-red έγινε ώστε να συνδέσουμε τα δεδομένα τηλεμετρίας όπου λαμβάνουμε από το Tasmota των αισθητήρων,τα intents όπου αναγνωρίζονται μέσω του Rhasspy και των εργαλείων του,καθώς και για τη δημιουργία σεναρίων. Αριστερά του UI φαίνονται οι κόμβοι όπου ποικίλουν απο έναν απλό debug,σε functions οπού επιτρέπουν εντολές javascript,σε change node όπου μετασχοιματίζουμε το εισερχόμενο μήνυμα σε διάφορους κόμβους κ.α.
Στο κέντρο έχουμε το παράθυρο σχεδιασμού σεναρίων.
Δεξιά τις πληροφορίες και επιλογές του επιλεγμένου node.
Μπάρα ελέγχου με το μενού της εφαρμογής και το κουμπί Deploy,στην κορυφή.



εικόνα 6.9: Node-Red UI

Το πρώτο βήμα που κάναμε ήταν να συνδέσουμε ένα websocket in όπου παίρνει δεδομένα από το directory api/events/intents του Rhasspy. Βρίσκουμε τον κόμβο websocket απο την παλέτα και τον φέρνουμε στο παράθυρο σχεδιασμού(drag and drop). Κάνοντας διπλό κλικ πάνω του πρέπει να ρυθμίσουμε το socket όπου θα παίρνουμε τα δεδομένα.Στην δική μας περίπτωση χρησιμοποιούμε το εργαλείο intent του Rhasspy,όπως φαίνεται στην παρακάτω εικόνα.6.10



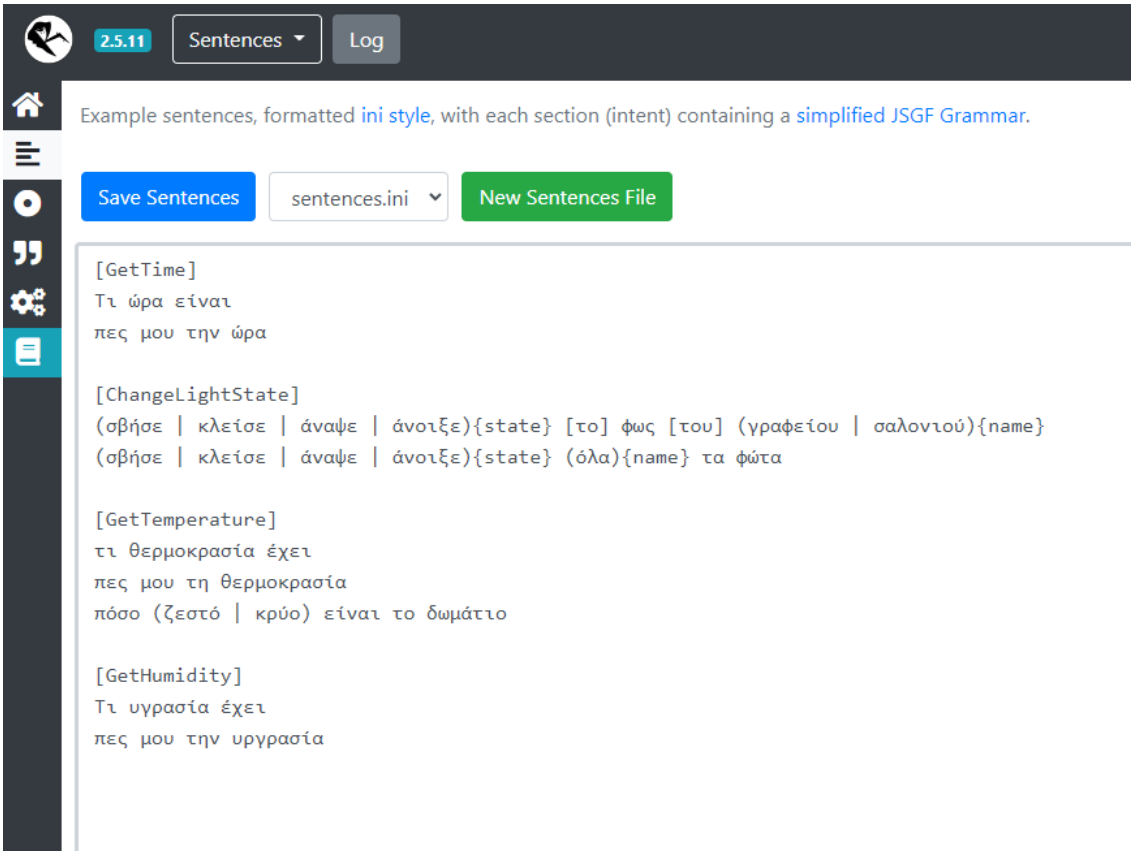
εικόνα 6.10: κόμβος websocket

Έπειτα συνδέουμε και έναν κόμβο debug.Με αυτό τον τρόπο,όποτε το Rhasspy αναγνωρίσει μια φωνητική εντολή που περιλαμβάνει ένα intent στέλνει ένα JSON event στην websocket που προσθέσαμε.
Με την ολοκλήρωση του σχεδιασμού πατάμε το κουμπί Deploy πάνω δεξιά,ώστε να φορτωθεί τα flows

6.4 Προγραμματισμός του σεναρίου μας

Η λίστα με τα intents που μπορεί να αναγνωρίσει το Rhasspy βρίσκεται στη δεύτερη καρτέλα αριστερά στο UI του Rhasspy.

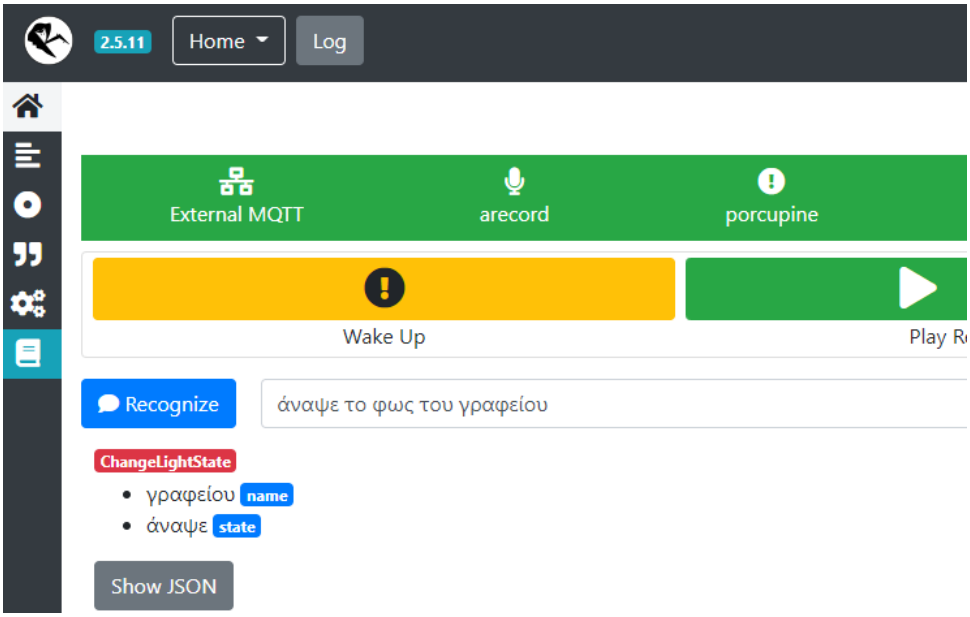
Η σύνταξη αυτών των προτάσεων είναι αρκετά απλή,στην ουσία μέσα στις τετράγωνες αγκύλες έχουμε το όνομα του intent το οποίο καλείται μέσω του εργαλείου intent recognition,όταν το speach recognition αναγνωρίσει μια από τις προτάσεις που υπάρχουν από κάτω.Μέσα στις παρενθέσεις έχουμε τις παραμέτρους οι οποίες χωρίζονται με το σύμβολο | στις οποίες ο τύπος τους βρίσκεται ανάμεσα στις αγκύλες που ακολουθούν. Χρησιμοποιήσαμε τουλάχιστον δυο προτάσεις στο κάθε σενάριο και διάφορες λέξεις για την κάθε κατάσταση καθώς πρέπει να λάβουμε υπόψιν μας και την ποικιλία του προφορικού λόγου.



εικόνα 6.11: Προτάσεις σεναρίων

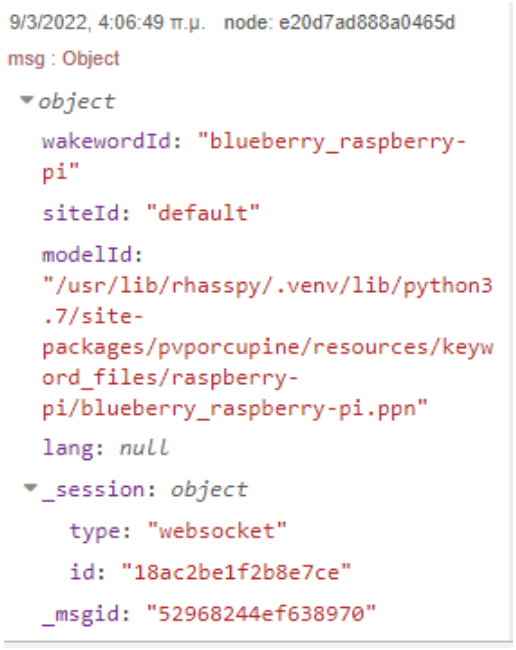
6.4.1 Ενεργοποίηση/Απενεργοποίηση διακοπτών

Εφόσον "ξυπνήσουμε" το Rhasspy και πούμε *"αναψε το φως του γραφείου"* το Speech to Text εργαλείο θα μετατρέψει αυτό που είπαμε σε κείμενο,το οποίο εάν αποτελεί μέρος μιας πρότασης που έχουμε στο προφίλ μας,θα αναγνωριστεί ως intent. Αυτό που ακούει το Rhasspy εμφανίζεται σε μορφή κειμένου δίπλα από το κουμπί recognize του κεντρικού μενού. Το intent οποίο φαίνεται στο γραφικό περιβάλλον,η λέξη "γραφείου" είναι μια παράμετρος name και η λέξη "αναψε" μια παράμετρεος state.(εικόνα6.12)

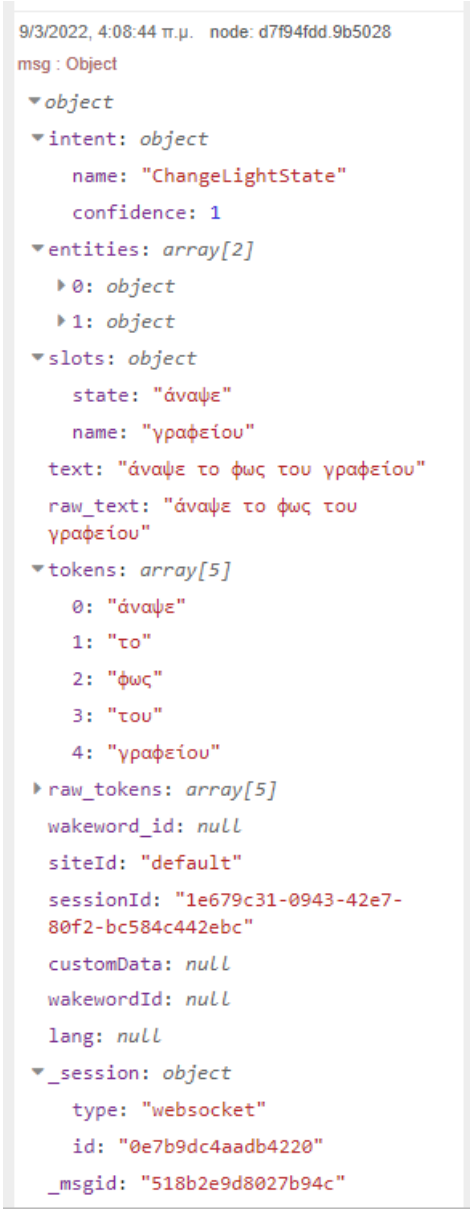


εικόνα 6.12: Αναγνώαριση Intent

Έπειτα αυτό το intent βρίσκεται σε μορφή JSON και είναι έτοιμο να σταλεί στο websocket in του node-red,όπως δείξαμε και προηγουμένως. Το debug των flows μέχρι στιγμής φαίνεται αναλυτικότερα στις παρακάτω δυο εικόνες6.13



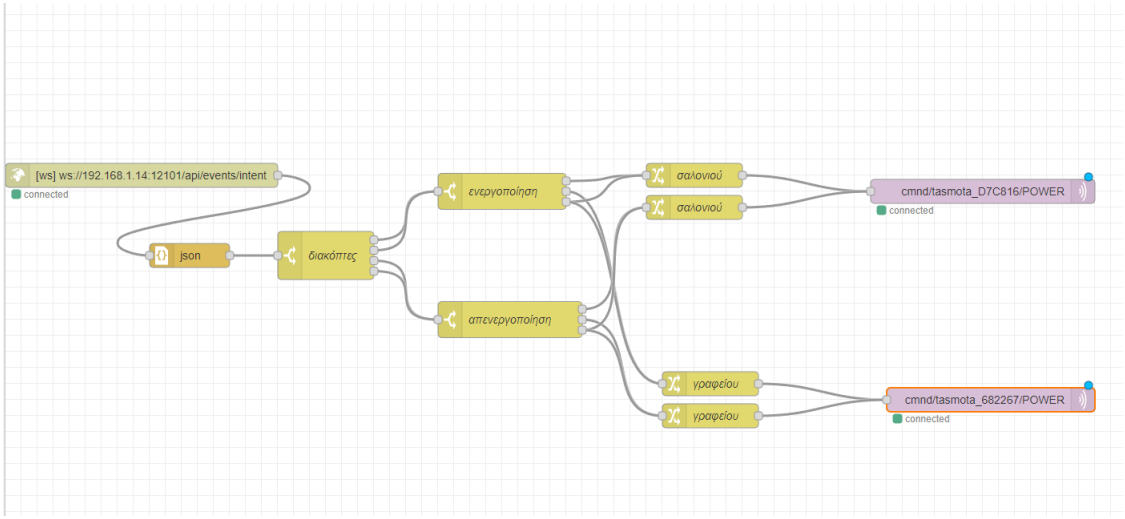
(a) wake word



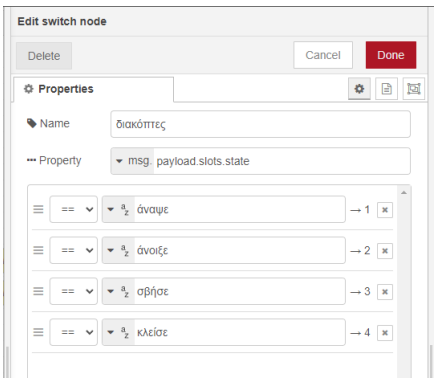
(b) change light

Figure 6.13: Έξοδος Debug JSON

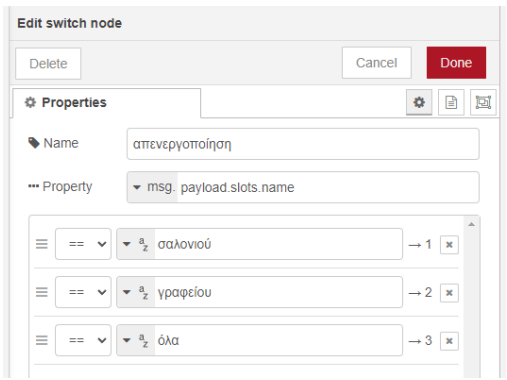
Η διαδικασία συνεχίζεται στο node-red,το επόμενο node που προσθέσαμε είναι για την μετατροπή του JSON σε Javascript Object το οποίο περιέχει το payload σαν παράμετρο. Στη δική μας περίπτωση έχουμε το payload.slots.state με τέσσερις παραμέτρους και το payload.slots.name με τρεις παραμέτρους.Το επόμενο βήμα ήταν ο διαχωρισμός αυτών των παραμέτρων μέσα από μια σειρά switch nodes,καταλήγοντας σε ένα change node το οποίο δέχεται σαν είσοδο ένα msg.payload και το μετατρέπει σε ένα ON ή OFF string. Αυτή την εντολή τελικώς δέχεται η κονσόλα tasmota της συσκευής sonoff μέσω ενός mqtt out node,το οποίο κάνει publish στο topic cmnd/tasmota_D7C816/POWER. Η οπτικοποίηση της διαδικασίας που εξηγήσαμε φαίνεται στο παρακάτω flow δίαγραμμα.6.14 και στα switch nodes6.15



εικόνα 6.14: Ροή διακοπτών στο node-red



(a) state payload

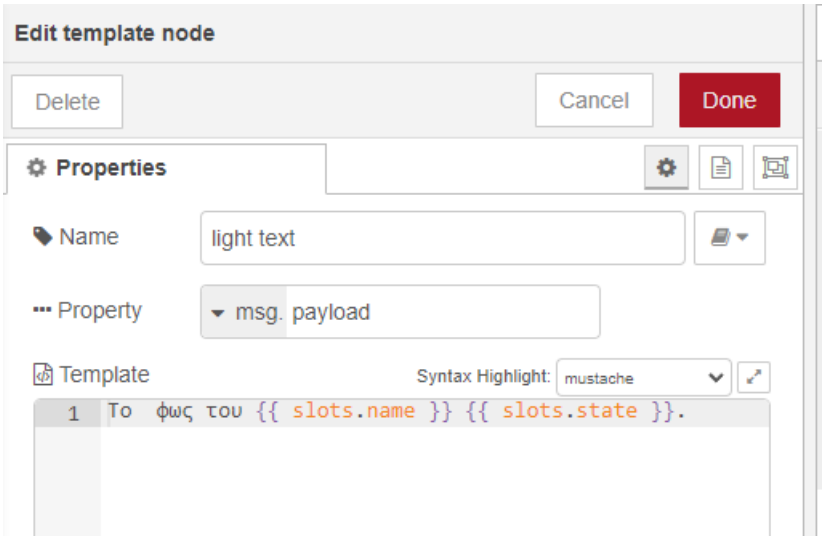


(b) name payload

Figure 6.15: Κόμβοι Switch

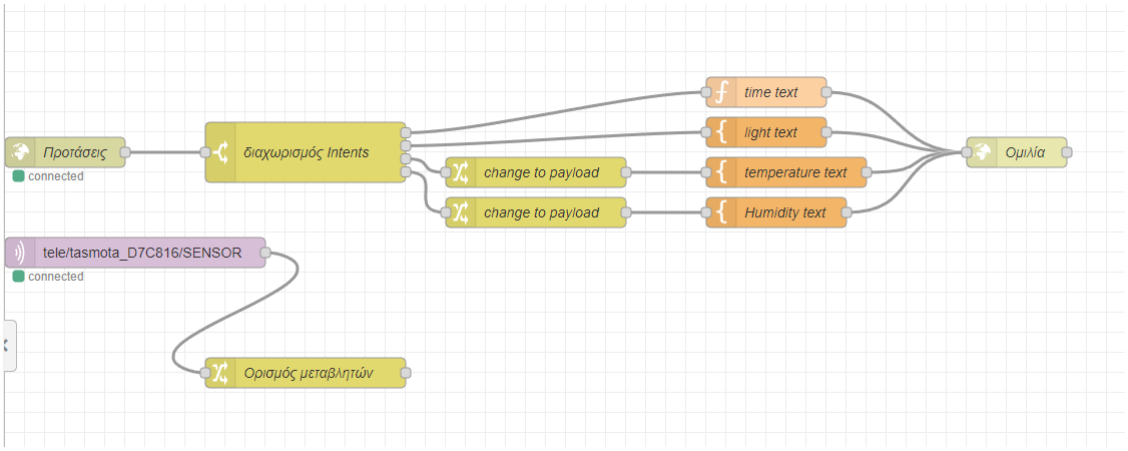
Για να έχουμε καλύτερο feedback όταν πραγματοποιείται μια εντολή δημιουργήσαμε

ένα άλλο flow το οποίο χρησιμοποιεί την text to speech api του Rhasspy για να μας δίνει φωνητική επιβεβαίωση. Σε αυτό το σενάριο θέλουμε να μας λέει ότι το φως του χώρου,άναψε ή έσβησε. Ξεκινάμε το πρώτο μας node ξανά με ένα websocket in το οποίο είναι συνδεδεμένο με το ws://localhost:12101/api/events/intent,έπειτα χωρίζουμε περιπτώσεις με ένα switch node με βάση το msg.intent.name. Το flow όταν αναγωρίσει το intent *ChangeLightState* συνεχίζει με ένα template node στο οποίο θα γράψουμε το κείμενο που θα μεταφερθεί στην text to speech api του Rhasspy μέσω ενός http request mode.Η έξοδος του tempate node πρέπει να είναι *plain text*,ενώ για format χρησιμοποιήσαμε το *mustache template* όπως φαίνεται και στην εικόνα6.16



εικόνα 6.16: Template ομιλίας

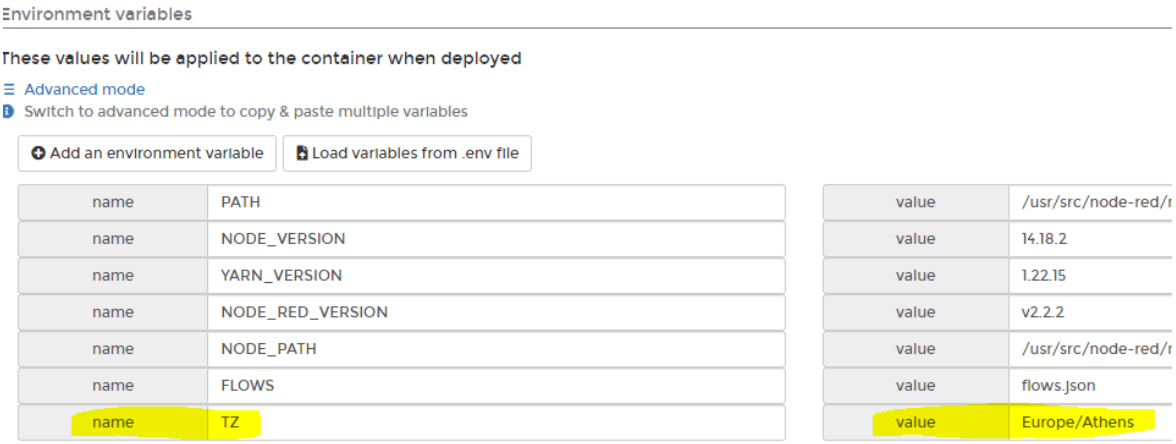
Για αυτό το παράδειγμα το slots.name παίρνει την λέξη "γραφείου" και το slots.state την λέξη "άναψε". Τελικώς το εργαλείο espeak θα πει *«το φως του γραφείου άναψε»*



εικόνα 6.17: Flow φωνητικής ανάδρασης

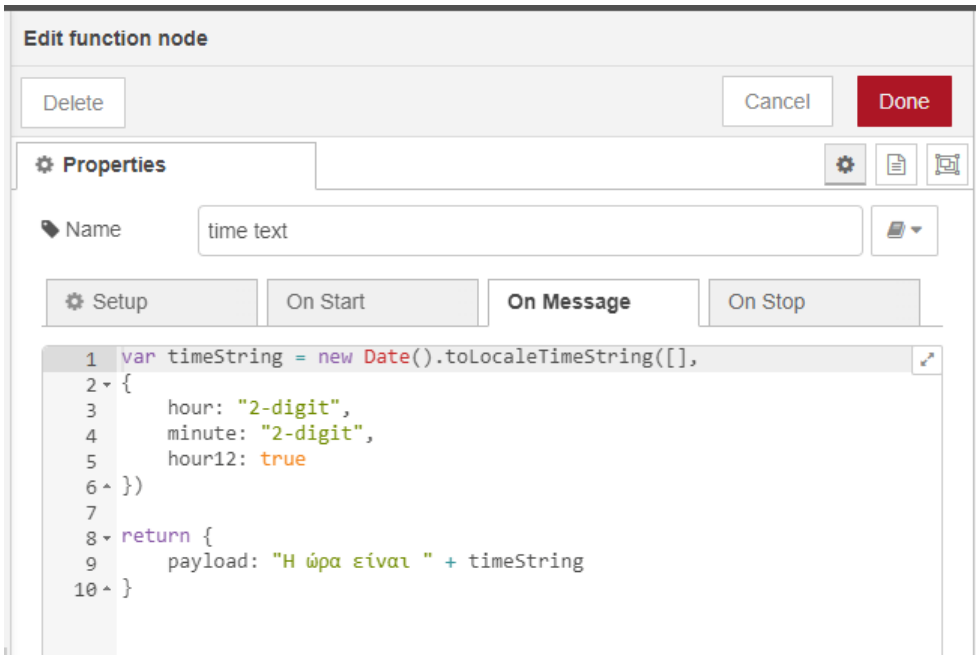
6.4.2 Λήψη ώρας

Στο intent GetTime ο φωνητικός βοηθός μας λέει την ώρα.Για να το καλέσουμε μπορούμε να πούμε *"τι ώρα είναι"* ή *"πες μου την ώρα"* όπως φαινόταν και στην παραπάνω εικόνα με τις προτάσεις σεναρίων. 6.11 Έπειτα το Rhasspy αναγνωρίζει το intent,το οποίο περνάει στον κόμβο websocket in(προτάσεις στην εικόνα 6.17) του node-red και στη συνέχεια σε ένα function node,το οποίο επιστρέφει την τοπική ώρα.Η μεταβλητή της ώρας λαμβάνεται από το ρολόι του node-red.Εξ'αρχής το ρολόι είναι ρυθμισμένο να δείχνει την Μέση Ώρα Γκρίνουιτς ή όπως είναι γνωστή πλέον την UTC (Coordinated Universal Time). Για να το αλλάξουμε στην ώρα Αθήνας ανατρέξαμε στο container του node-red μέσω της εφαρμογής portainer και έπειτα στην καρτέλα Duplicate/Edit. Έπειτα στις Environment variables στην καρτέλα Env προσθέσαμε την μεταβλητή με όνομα TZ και τιμή Europe/Athens.



εικόνα 6.18: Ρύθμιση ρολογιού Node-Red

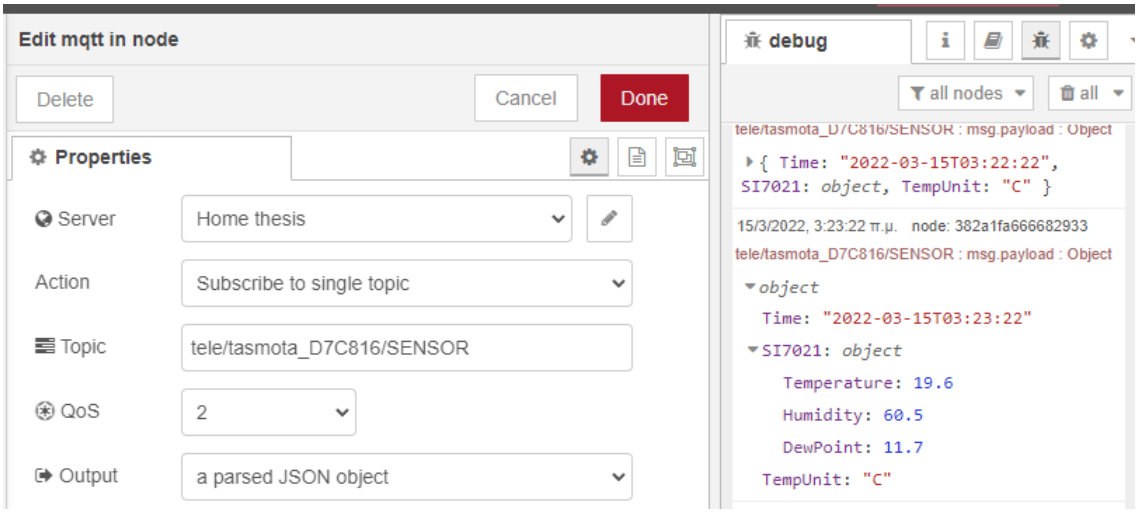
Εντός της συνάρτησης στον κόμβο function,δημιουργούμε μια νέα μεταβλητή timeString στην οποία ορίζουμε την τιμή του container και αποθηκεύει την ώρα και τα λεπτά σε μορφή string. Στις αγκύλες γράφουμε το κείμενο που επιστρέφει το payload. Τέλος για να μετατρέψουμε την τιμή που επιστρέφει το function σε φωνητική εντολή,χρησιμοποιήσαμε ένα http request node που κάνει POST το payload στην υπηρεσία text to speech του Rhasspy.



εικόνα 6.19: Κόμβος συνάρτησης GetTime

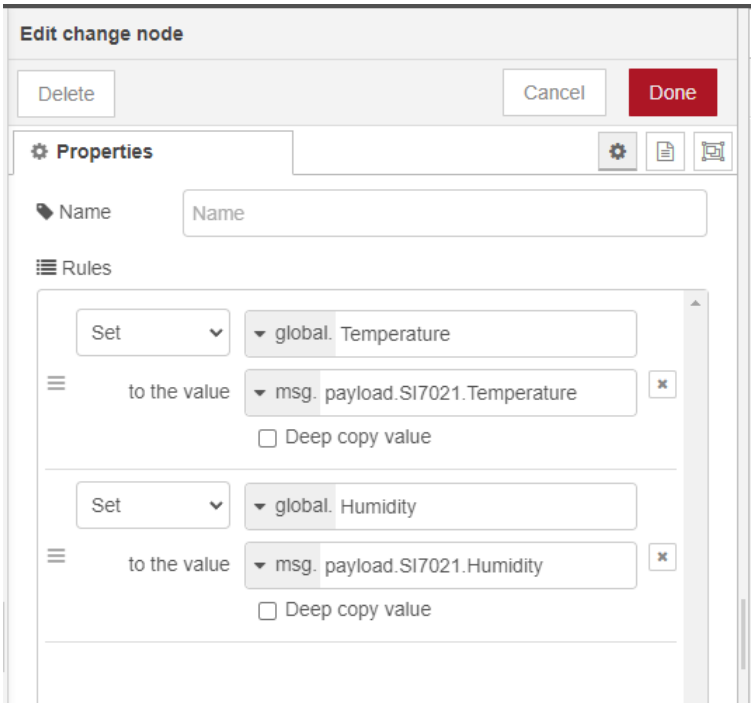
6.4.3 Λήψη Θερμοκρασίας/Υγρασίας

Όπως είδαμε προηγουμένως μέσω του αισθητήρα Si7021 είναι δυνατή η μέτρηση της θερμοκρασίας και της υγρασίας του χώρου.Θεωρήσαμε απαραίτητη την δημιουργία ενός intent οπού θα μας ενημερώνει για αυτές τις τιμές όταν ρωτάμε τον φωνητικό βοηθό.Οι προτάσεις που χρησιμοποιήσαμε για να καλέσουμε το Intent του GetTemperature και GetHumidity φαίνονται στην παραπάνω εικόνα. 6.11 Μέσω ενός mqtt in κόμβου όπου κάνουμε εγγραφή στο topic του αισθητήρα παίρνουμε τις τιμές που χρειαζόμαστε όπως φαίνεται και στην παρακάτω εικόνα. 6.20Ο ρυθμός ανανέωσης είναι 60 δευτερόλεπτα όπως είδαμε προηγουμένως μέσω της εντολής teleperiod 60 που δώσαμε στην κονσόλα της συσκευής tasmota.Η έξοδος του κόμβου πρέπει να είναι JSON object,οι τιμές της θερμοκρασίας και υγρασίας φαίνονται εντός του object SI7021 στο debug του node.



εικόνα 6.20: Κόμβος subscribe του αισθητήρα

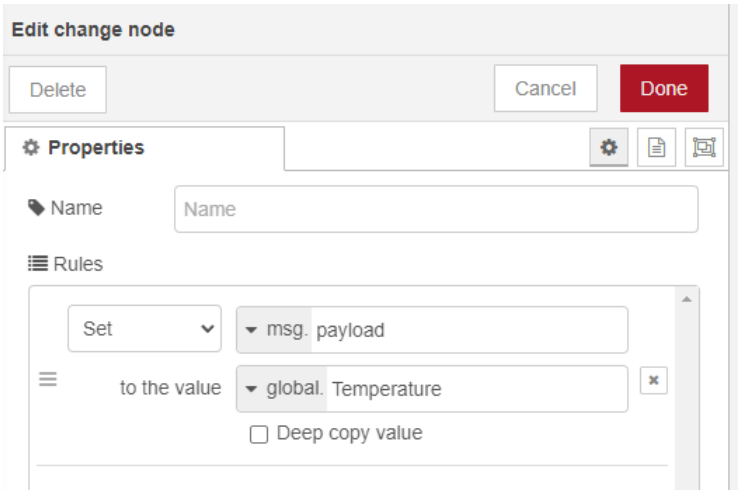
Για να αποθηκεύσουμε αυτές τις τιμές ώστε μετά να τις περάσουμε εντός του flow οπού καλούμε τα intents θα χρειαστούμε έναν change κόμβο.Με τον τρόπο που φαίνεται στην παρακάτω εικόνα6.23,ορίζουμε δυο μεταβλητές τύπου global ώστε να μπορούμε να τις χρησιμοποιήσουμε σε όλα τα flows που έχουμε εντός του γραφικού περιβάλλοντος.



εικόνα 6.21: Κόμβος ορισμού μεταβλητών

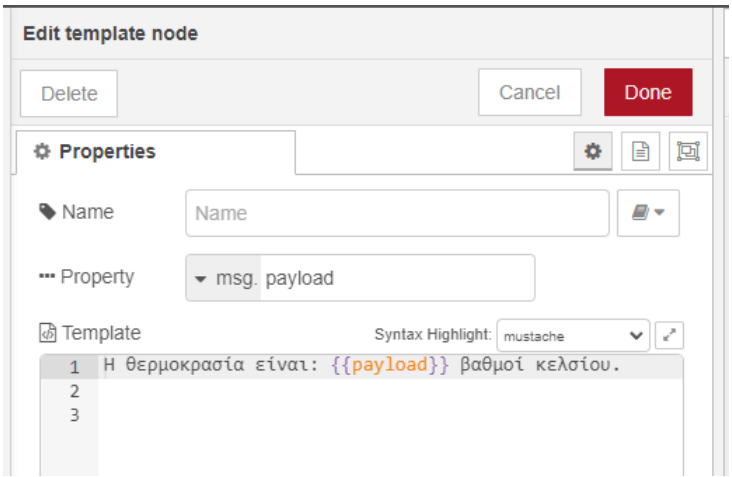
Στη συνέχεια εντός του flow 6.17οπού καταλήγει στον κόμβο που κάνει POST στην υπηρεσία espeak,δημιουργούμε δύο επιπλέον change nodes στα οποία ορίζουμε το payload

να παίρνει την τιμή των global μεταβλητών.



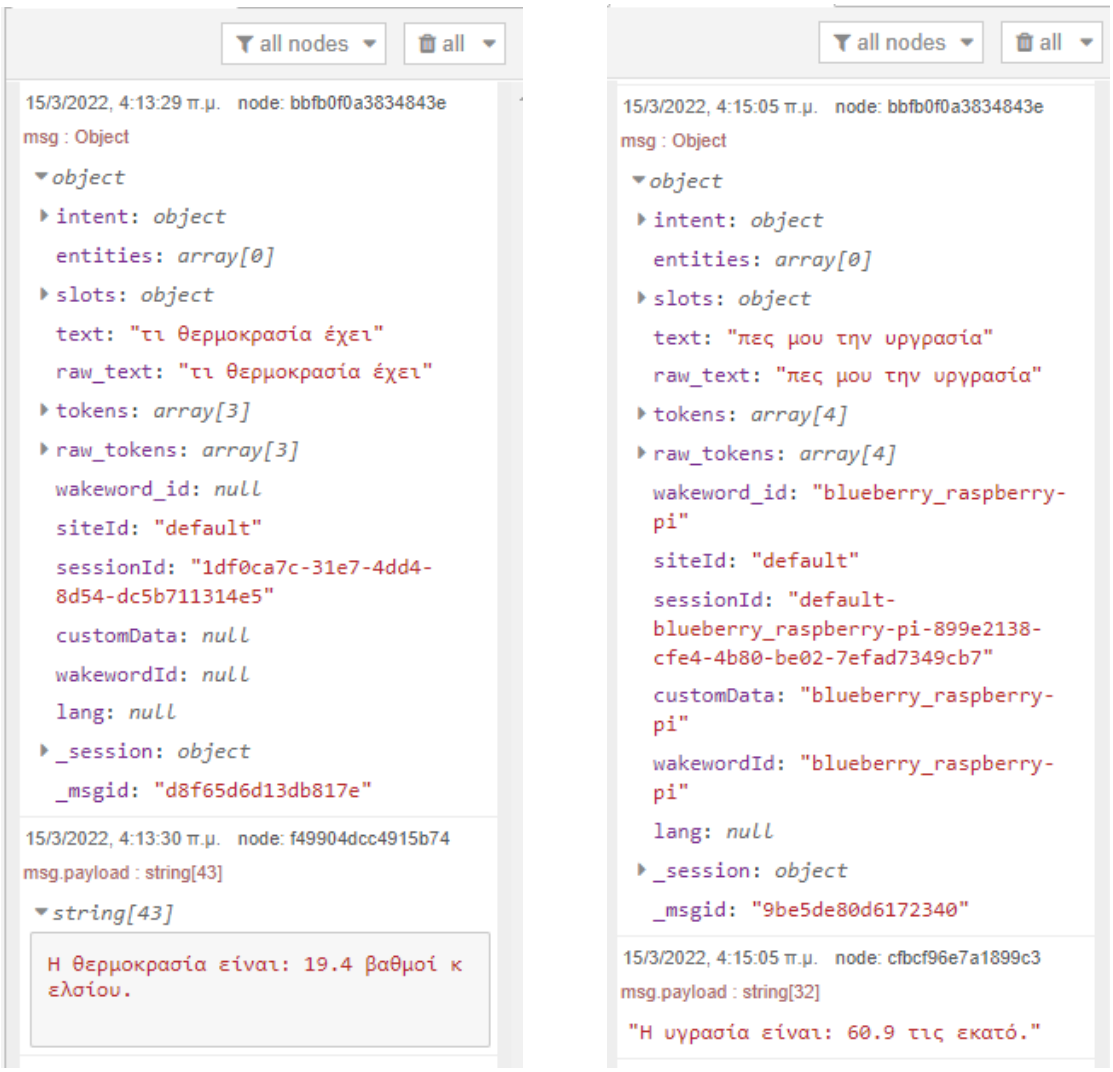
εικόνα 6.22: Κόμβος ορισμού μεταβλητών

Τέλος μέσω ενός template node γράφουμε το κείμενο που θέλουμε ως έξοδο για την μετατροπή του σε ομιλία.Η έξοδος και το format του συγκεκριμένου κόμβου είναι ίδια με το προηγούμενο template node που χρησιμοποιήσαμε για την επιβεβαίωση της κατάστασης των διακοπτών.



εικόνα 6.23: Template ομιλίας ΙΙ

Στις παρακάτω εικόνες φαίνεται το debug του node-red όταν έχουν τρέξει τα intetns GetTemperature και GetHumidity. Η εντολή που έλαβε φαίνεται στις αγκύλες του text,ενώ η φωνητική απάντηση που έδωσε φαίνεται στο κάτω μέρος με ένα payload string.



(a) temperature debug (b) humidity debug

Figure 6.24: Debug ομιλίας

Μέρος ΙΙΙ

Συμπεράσμα και Μελλοντικές Επεκτάσεις

Κεφάλαιο 7

Συμπεράσμα και ΜελλοντικέςΕπεκτάσεις

7.1 Συμπεράσματα

Παρά τα μειονεκτήματά που έχει ο φωνητικός βοηθός μας έναντι αυτών του εμπορίου,σε ότι αφορά την έλλειψη ενσωματωμένης τεχνητής νοημοσύνης, την πιο ορθή κατανόηση της φυσικής γλώσσας σε μεγαλύτερη εμβέλεια και θορυβώδης περιβάλλον, παράγοντες που εξαρτώνται κυρίως από το software και τους αλγόριθμους μηχανικής μάθησης που βρίσκονται στους cloud servers της εκάστοτε εταιρίας,τα κύρια προτερήματα της δικής μας υλοποίησης είναι η χρήση της ελληνικής γλώσσας οπού δεν διατίθεται σε κανέναν εμπορικό φωνητικό βοηθό,έχοντας μια πιο προσιτή και εύκολη λύση για όσους μιλάνε ελληνικά απευθυνόμενοι σε μεγαλύτερο κοινό.Παράλληλα δεν τίθεται θέμα ανησυχίας της ιδιωτικότητας του χρήστη καθώς δεν γίνεται χρήση κάποιας τρίτης υπηρεσίας εκτός του τοπικού δικτύου,ενώ αξίζει να αναφερθεί ότι και σε περίπτωση προβλήματος του πάροχου τηλεφωνίας ή αποσύνδεσης από τον παγκόσμιο ιστό,εφόσον στο modem μας λειτουργεί το DSL, το σενάριο μας εκτελείται απροβλημάτιστα.

7.2 Μελλοντικές Επεκτάσεις

Πάνω στην ήδη υπάρχουσα λύση θα ήταν δυνατό και ιδιαίτερα χρήσιμο σε κάποιες περιπτώσεις, η εφαρμογή και επέκταση στους παρακάτω τομείς.

7.2.1 Αναβάθμιση συγκεκριμένων εργαλείων λογισμικού

Η μηχανή TextToSpeach,espeak που χρησιμοποιούμε μπορεί να κάνει την μετατροπή της ελληνικής γλώσσας offline,ωστόσο η ποιότητα της ανάγνωσης των ελληνικών αποκλίνει αρκετά από την ομαλότητα της φυσικής γλώσσας,γι'αυτό το λόγο είναι απαραίτητη η χρήση μιας μηχανής TtS που θα μπορεί να ενταχθεί στο Rhasspy,η οποία θα μιμείται την ελληνική ομιλία πιο κοντά στην φυσική της μορφή. Επιπρόσθετα η ένταξη ενός εργαλείου speaker recognition όπου θα μπορεί να διαχωρίζει τα πρόσωπα με βάση την φωνή τους θα αύξανε περαιτέρω την ασφάλεια και θα βοηθούσε σε περισσότερες εφαρμογές.

7.2.2 Εγκατάσταση σε νοσοκομεία και οίκους ευγηρίας

Το σενάριο οπού θα μπορούμε να έχουμε σε διάφορα δωμάτια νοσοκομείων,οίκων ευγηρίας ή άλλων κτηρίων εγκατεστημένο ένα σύστημα όπου ασθενείς,ηλικιωμένοι ή άλλα άτομα που έχουν κινητικά ή άλλα προβλήματα να δίνουν φωνητικές εντολές στα ελληνικά ώστε να έχουν μια πιο εύκολη πρόσβαση και αλληλεπίδραση με υλικό ή προς τρίτους,χωρίς μεγάλο κόστος και συνεχή συντήρηση είναι εφικτό μέσω ενός συστήματος με satellites, οι οποίοι συνδέονται σε έναν μεγαλύτερο κεντρικό server. Οι δορυφόροι θα μπορούν να είναι υπολογιστές raspberry pi και θα αναλαμβάνουν την αναγνώριση της wake word,της ηχογράφηση μέσω του μικροφώνου και την αναπαραγωγή μέσω των ηχείων. Ενώ ο server είναι υπεύθυνος για την μετατροπή φωνής σε κείμενο,την αναγνώριση των intent,την μετατροπή του κειμένου σε φωνή και τον χειρισμό των intent.

7.2.3 Επέκταση σε άλλες εφαρμογές και συσκευές

Η δυνατότητα αναγνώρισης της ελληνικής γλώσσας είναι κάτι που θέλουμε να δούμε να εφαρμόζεται και σε περισσότερους φωνητικούς βοηθούς. Η βάση του opensource μας δίνει πολλές δυνατότητες για μελλοντικές επεκτάσεις της εφαρμογής μας,κυρίως σε νέες συσκευές έξυπνων ηχείων που βασίζουν το λειτουργικό σε ανοιχτό κώδικα.Ήδη γνωρίζουμε ότι ο προγραμματιστής του Rhasspy,Michael Hansen ο οποίος εργάζεται πλέον στην Mycroft θα ήθελε στο μέλλον να ενταχθούν στο υπό ανάπτυξη έξυπνο ηχείο της εταιρίας Mark II,τα κύρια χαρακτηριστικά του Rhasspy.

Η χρήση του φωνητικού βοηθού από άτομα που πάσχουν από κινητικά προβλήματα,προβλήματα όρασης και ηλικιωμένους που ομιλούν κυρίως ελληνικά,είναι ένα ζήτημα

που θα πρέπει να εξεταστεί καθώς θα μπορεί να αποτελέσει ένα πολύ χρήσιμο εργαλείο για την καθημερινότητα τους. Η δωρεάν διάθεση του Rhasspy βοηθάει πολύ στην επέκταση και άλλων συστημάτων,ωστόσο η απαραίτητη γνώση προγραμματισμού είναι κάτι που το καθιστά αδύνατο να απευθυνθεί στον απλό χρήστη,ωστόσο μέσω της ενσωμάτωσης του σε ένα mobile app,οπού η συσκευή του κινητού θα λειτουργεί σαν broker όλων των υπηρεσιών του προγράμματος,θα ήταν πολύ πιθανό να γίνει αρκετά πιο προσιτό στην χρήση.

7.2.4 Αυτονομία ενέργειας

Η ενέργεια που καταναλώνει το Raspberry Pi 4 σε κατάσταση αδράνειας είναι περίπου 4 W,ενώ όταν απασχολείται πάνω από ένας πυρήνας του επεξεργαστή η κατανάλωση μπορεί να φτάσει και τα 5W. Αν και τα ποσά κατανάλωσης είναι αρκετά μικρά,η εξάρτηση του Raspberry Pi να βρίσκεται συνεχώς στο ρεύμα και κοντά σε πρίζα είναι κάτι που θα μπορούσαμε στο μέλλον να το αποφύγουμε,έχοντας μια πηγή ενέργειας που θα το έκανε πιο φορητό. Μια DIY λύση είναι με την χρήση AA μπαταριών και μιας Battery eliminator circuit συσκευής η οποία στην ουσία είναι ένας ρυθμιστής τάσης,οπού χρησιμεύει για να ρίξουμε την υψηλή τάση σε χαμηλότερα επίπεδα,να τα συνδέσουμε δημιουργώντας ένα κύκλωμα για την παροχή ρεύματος. Ευκολότερη δε και πιο πρακτική λύση είναι Το ίδιο ωστόσο αποτέλεσμα με πιο πρακτικό και εύκολο τρόπο μας τον δίνει η συσκευή PiJuice HAT [34] ,πρόκειται στην ουσία για ένα UPS οπού κρατάει το Raspberry Pi εκτός ρεύματος για αρκετή ώρα και εφαρμόζει επάνω στα πινάκια της πλακέτας του υπολογιστή,ενώ υπάρχουν διάφορα μεγέθη χωρητικότητας mAh που είναι διαθέσιμο.

Παράρτημα Α

Ακρωνύμια και συντομογραφίες

NLP Natural Language Processing

IoT Internet of Things

VA Virtual Assistant

SSH Secure SHell

HDMI High-Definition Multimedia Interface

IP Internet Protocol adress

UI User Interface

ALSA Advanced Linux Sound Architecture

API Application Programming Interface

MQTT Message Queuing Telemetry Transport

TCP/IP Transmission Control Protocol/Internet Protocol

HTTP Hypertext Transfer Protocol

GPIO General Purpose input/output

GUI Graphic User Interface

QoS Quality of Service

TLS Transport Layer Security

JSON JavaScript Object Notation

DIY Do It Yourself

WAV Waveform Audio File Format

DSL Digital Subscriber Line

UPS Uninterruptible Power Supply

Βιβλιογραφία

[1] “seeedstudio.com.” [Online]. Available: https://www.seeedstudio.com/blog/2018/11/23/6-important-speech-recognition-technology-you-need-to-know/

[2] “alexa.” [Online]. Available: https://beebom.com/apple-homepod-vs-amazon-echo-vs-google-home/

[3] “downdetector.” [Online]. Available: https://downdetector.com/status/aws-amazon-web-services/

[4] “googlehomesdk.” [Online]. Available: https://developers.googleblog.com/2020/04/local-home-sdk-ready-for-actions.html

[5] “data smart assistant collect.” [Online]. Available: https://www.pcmag.com/news/amazons-alexa-collects-more-of-your-data-than-any-other-smart-assistant

[6] “mycroftmarki.” [Online]. Available: https://mycroft.ai/product/mycroft-mark-1/

[7] “Smart home - statistics and facts.” [Online]. Available: https://www.statista.com/topics/2430/smart-homes/#topicHeader__wrapper

[8] “Voice assistants: How artificial intelligence assistants are changing our lives every day.” [Online]. Available: https://www.smartsheet.com/voice-assistants-artificial-intelligence

[9] “www.ibdm.com.” [Online]. Available: https://www.ibm.com/ibm/history/exhibits/specialprod1/specialprod1_7.html

[10] A. T. A. S. A. N. Fadhil, “Voice recognition system using machine learning techniques.” [Online]. Available: https://www.sciencedirect.com/science/article/pii/S221478532102931X

[11] X. L. G.-H. Tu, “The insecurity of home digital voice assistants – amazon alexa as a case study.” [Online]. Available: https://arxiv.org/pdf/1712.03327.pdf

[12] “Amazon local controls.” [Online]. Available: https://www.amazon.com/gp/help/customer/display.html?nodeId=GCC6XV9DX58VW5YW

[13] “Google local home sdk.” [Online]. Available: https://developers.google.com/assistant/smarthome/concepts/local

[14] “www.reviews.org.” [Online]. Available: https://www.reviews.org/home-security/smart-assistant-privacy-what-data-is-collected-and-how-to-protect-yourself/

[15] “www.amazon.com.” [Online]. Available: https://www.amazon.com/gp/help/customer/display.html?nodeId=GVP69FUJ48X9DK8V

[16] “https://blog.google/.” [Online]. Available: https://blog.google/products/assistant/doing-more-protect-your-privacy-assistant/

[17] “www.apple.com.” [Online]. Available: https://www.apple.com/in/newsroom/2019/08/improving-siris-privacy-protections/

[18] “Mycroft voice assistant.” [Online]. Available: https://mycroft.ai/

[19] “Jasper voice assistant.” [Online]. Available: https://jasperproject.github.io/

[20] “Rhasspy voice assistant.” [Online]. Available: https://rhasspy.readthedocs.io/en/latest/

[21] “What is a container.” [Online]. Available: https://www.docker.com/resources/what-container

[22] “raspberry specs.” [Online]. Available: https://https://www.raspberrypi.org/products/raspberry-pi-4-model-b/specifications/

[23] “Os.” [Online]. Available: https://downloads.raspberrypi.org/raspios_armhf/release_notes.txt

[24] “Hat speaker.” [Online]. Available: https://respeaker.io/2_mic_array/

[25] “sonoff-th10.” [Online]. Available: https://sonoff.tech/product/wifi-diy-smart-switches/th10-th16

[26] “sonoff s26.” [Online]. Available: https://sonoff.tech/product/wifi-smart-plugs/s26

[27] “Esp8266.” [Online]. Available: https://en.wikipedia.org/wiki/ESP8266

[28] “rasspberrypi.” [Online]. Available: https://www.raspberrypi.com/software/operating-systems/

[29] “Linux man page.” [Online]. Available: https://linux.die.net/man/1/arecord

[30] “From scratch on a raspberry pi.” [Online]. Available: https://rhasspy.readthedocs.io/en/latest/tutorials/#from-scratch-on-a-raspberry-pi

[31] “Transport layer security.” [Online]. Available: https://en.wikipedia.org/wiki/Transport_Layer_Security

[32] “Why mqtt.” [Online]. Available: mqtt.org

[33] “Eclipse mosquitto.” [Online]. Available: mosquitto.org

[34] “Pijuice hat.” [Online]. Available: https://uk.pi-supply.com/products/pijuice-standard