



ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ
Εθνικό και Καποδιστριακό
Πανεπιστήμιο Αθηνών

Δ.Π.Μ.Σ. των Τμημάτων Φυσικής και Πληροφορικής & Τηλεπικοινωνιών
του ΕΚΠΑ
(ΜΔΕ) στον Ηλεκτρονικό Αυτοματισμό (Η/Α) (MSc in Control and Computing)

Διπλωματική Εργασία

*Έξυπνο σπίτι και οδηγός εγκατάστασης της πλατφόρμας ανοιχτού κώδικα
Home Assistant
Smart Home and Installation Guide for the Home Assistant open source platform*

Ιωαννίδης Κωνσταντίνος

Επιβλέπων Καθηγητής
Κούτουλας Δημήτριος

Αθήνα, Νοέμβριος 2019

ΠΕΡΙΕΧΟΜΕΝΑ

ΠΕΡΙΛΗΨΗ

Κεφάλαιο 1

1) Εισαγωγή

- 1.1) Ασφάλεια
- 1.2) Άνεση
- 1.3) Οικονομία

Κεφάλαιο 2

2) Πρωτόκολλα επικοινωνίας

Κεφάλαιο 3

3) Λογισμικό ανοιχτού κώδικα HOME ASSISTANT

- 3.1) Οδηγός Εγκατάστασης Home Assistant σε Raspberry Pi
- 3.2) Εισαγωγή
 - 3.2.1) Τι είναι το Home Assistant
 - 3.2.2) Τι χρησιμοποιήσαμε
 - 3.2.3) Τι είναι το Raspberry Pi
- 3.3) Τρόποι εγκατάστασης
 - 3.3.1) Οδηγός εγκατάστασης Home Assistant με τη μέθοδο του Hass.io
 - 3.3.2) Ξενάγηση στο Μενού
- 3.4) Εγκατάσταση βασικών προσθέτων
 - 3.4.1) Εγκατάσταση Configurator add-on
 - 3.4.2) Εγκατάσταση προσθέτου Samba Share
 - 3.4.3) Εγκατάσταση Duck DNS
 - 3.4.4) Εγκατάσταση για device tracking του OWNTRACKS
- 3.5) Προσθήκη συσκευών στο Home Assistant
 - 3.5.1) Προσθήκη αισθητήρων και συσκευών Zigbee
 - 3.5.2) Προσθήκη αισθητήρων και συσκευών Z-Wave
 - 3.5.3) Προσθήκη MQTT broker

Κεφάλαιο 4

4) Παραδείγματα Αυτοματισμών

- 4.1) # Example of entry in configuration.yaml
- 4.2) # Example for rainy days
- 4.3) # Examples using the sun

4.4)#Send sun rise/sun set notifications using pushbullet

4.5)#Automations for lights and blinds based on solar elevation

4.6)#Turn on lights with a resetable off timer

Κεφάλαιο 5

5)Επίλογος

ΒΙΒΛΙΟΓΡΑΦΙΑ

ΠΕΡΙΛΗΨΗ

Ο σύγχρονος τρόπος ζωής που έχουμε υιοθετήσει στην καθημερινότητα μας και ο διαθέσιμος χρόνος που περιορίζεται αρκετά από τις συνεχώς αυξανόμενες υποχρεώσεις έκανε ανάγκη αυτό που κάποτε ίσως να ήταν πολυτέλεια. Το να έχουμε την δυνατότητα ενώ απουσιάζουμε να δώσουμε εντολή στο σπίτι μας να ανοίξει μια συσκευή ή να την κλείσει ώστε με την άφιξη μας να έχουν ήδη δημιουργηθεί οι συνθήκες που θέλουμε (π.χ. να ανάψει ο θερμοσίφωνας ώστε να υπάρχει ζεστό νερό ή να ανοίξει η θέρμανση ή ο κλιματισμός και να είναι το σπίτι μας στην θερμοκρασία που θέλουμε) μας εξοικονομεί πολύτιμο χρόνο και φυσικά χρήμα. Η ανάγκη για οικονομία στην κατανάλωση ηλεκτρικής ενέργειας εκτός από την οικολογική προσέγγιση έχει και μεγάλο θετικό αντίκτυπο στην τσέπη μας και στα χρήματα που ξοδεύουμε για ηλεκτρική ενέργεια.

Η παρούσα πτυχιακή εργασία, με μορφή οδηγού, θέτει ως στόχο της να κατευθύνει τον αναγνώστη της όσο καλύτερα γίνεται στον χώρο του Internet of Things, των Smart Homes και στην ιδέα του home automation. Θα παρουσιαστούν τα βασικότερα και σημαντικότερα πρωτοκολλά επικοινωνίας όπως Zigbee, Z-Wave και MQTT κ.ά. Εν συνεχεία θα δοθούν κάποιες λύσεις ανοιχτού κώδικα που μπορούν να αποτελέσουν την καρδιά ενός 'έξυπνου' σπιτιού, ενώ κλείνοντας έχει δημιουργηθεί ένας οδηγός εγκατάστασης και ανάπτυξης για μια από αυτές, του Home Assistant, σε Raspberry Pi. Στη συνέχεια θα κάνουμε εγκατάσταση των πιο βασικών προσθέτων (ADD-ONS). Επίσης θα προσδώσουμε ασφαλή απομακρυσμένο έλεγχο στο Home Assistant μας μέσω DuckDns και επίσης θα γίνει Device tracking για τις smartphone συσκευές μας. Επιπλέον θα γίνει η προσθήκη συσκευών στο Home Assistant ανάλογα με το πρωτόκολλο επικοινωνίας που χρησιμοποιείται. Τέλος θα δωθούν παραδείγματα αυτοματισμών με την χρήση της γλώσσας YAML.

ABSTRACT

The modern way of life that we have adopted in our everyday activities and the available time is limited enough due to our constantly growing responsibilities has made a necessity that it would be once luxury. Having the ability to give orders to turn on or turn off an electrical appliance while we are from home in order to create the desirable conditions upon arrival (f.e. to turn on the boiler so as to have warm water or to turn on the heating or the air-conditioning in order our house to have the temperature that we want to) this above mentioned ability is time saving and economical as well. This necessity for saving of electrical energy not only does it have positive impact on the environment but it saves money as well.

In the thesis at hand, more specifically in a guide form, aims at familiarizing the reader as much as possible with the area of internet of things, smart home and the idea of home automation. The most important and essential “info communication protocols” will be presented like Zigbee, Z-Wave και MQTT etc. Thereinafter, some solution with regards to “open source” will be given which constitute the main aspects of smart home. In the last part of the thesis, an installment and development guide has been created for one of them (Home Assistant, in Raspberry Pi). We will then install basic add-ons. We will also add a secure remote control on Home Assistant via Duck Dns and Device tracking will be performed for our smartphone devices. Moreover, new devices will be added to “Home Assistant” according to the used communication protocol. At the end examples related to automation by using the YAML language will be provided.

Κεφάλαιο 1

1)Εισαγωγή

Οι έντονοι ρυθμοί του σύγχρονου τρόπου ζωής δημιουργούν διαρκώς νέες ανάγκες, οι οποίες απαιτούν την διαχείριση τους από ένα σύστημα αυτοματισμών και ελέγχου. Η φράση “έξυπνο σπίτι” χρησιμοποιείται για οποιαδήποτε οικία, ενσωματώνει – σε μικρότερο ή μεγαλύτερο βαθμό- τη δυνατότητα ρύθμισης και ελέγχου ορισμένων ηλεκτρομηχανολογικών εγκαταστάσεων.

Με τον όρο 'έξυπνα σπίτια' εννοούμε τα σπίτια ή τα κτήρια που είναι εξοπλισμένα με σύγχρονες και διασυνδεδεμένες 'έξυπνες' συσκευές και μηχανισμούς, δίνοντας τους τη δυνατότητα να επικοινωνούν με τους ιδιοκτήτες τους ακόμα και χωρίς να είναι απαραίτητη η φυσική παρουσία των τελευταίων. Αποτελούνται από ένα σύνολο αυτοματισμών με τους οποίους οργανώνονται και αυτοματοποιούνται οι λειτουργίες της κατοικίας βάσει των απαιτήσεων του κάθε χρήστη. Ο χρήστης έχει πλέον τη δυνατότητα να δημιουργεί σενάρια αυτοματισμών χρησιμοποιώντας την κάθε συσκευή για διάφορες χρήσεις και εφαρμογές.

Το σπίτι με νοημοσύνη σκέπτεται και ενεργεί βάση των καθημερινών σας αναγκών και συνηθειών, δίνοντας σας τον απόλυτο έλεγχο σε συστήματα ασφάλειας, θέρμανσης, φωτισμού, ηλεκτρικών συσκευών, περιεχομένων multimedia κ.α. Οι λειτουργίες αυτές ελέγχονται με το πάτημα ενός κουμπιού, είτε ο ιδιοκτήτης βρίσκεται εντός του κτιρίου είτε βρίσκεται σε κάποια απομακρυσμένη περιοχή.

Η Smart Home τεχνολογία, που βασίζεται στη φιλοσοφία του Internet of Things, κερδίζει με φρενήρεις ρυθμούς τη θέση της στην καθημερινότητα των ανθρώπων σε όλο τον κόσμο, ρυθμίζοντας το οικιακό τους περιβάλλον με βάση τις επιθυμίες τους. Οι πέντε πρώτες εφαρμογές της τεχνολογίας Smart Home παγκοσμίως, που έχουν προσελκύσει το μεγαλύτερο ενδιαφέρον των καταναλωτών, είναι οι «ασφάλεια και έλεγχος», «ενέργεια - φωτισμός» (55% και 53% αντίστοιχα) και «ψυχαγωγία - συνδεσιμότητα» (48%), η «υγεία» και οι «έξυπνες συσκευές» (43%).

Όπως εκτιμούν αναλυτές της παγκόσμιας αγοράς, μέχρι το 2023, οι ιδιώτες, οι εταιρείες και οι κρατικές δομές σε όλο τον κόσμο θα έχουν εγκαταστήσει περισσότερα από 40 δισ. συστήματα Internet of Things.

1.1)ΑΣΦΑΛΕΙΑ

Ένα έξυπνο σπίτι προσφέρει πλήθος λειτουργιών οι οποίες κατοχυρώνουν μεγαλύτερη ασφάλεια και προστασία της ιδιοκτησίας σας με εξελιγμένους τρόπους που προηγουμένως ήταν ανέφικτοι. Για παράδειγμα με τα συστήματα ασφαλείας οι ένοικοι μπορούν:

- Να ειδοποιηθούν ότι επιχειρείται διάρρηξη και παράλληλα να προκληθεί πανικός στους επίδοξους διαρρηκτες ενεργοποιώντας την σειρήνα και τον φωτισμό σε ολόκληρο το σπίτι.

Το σύστημα μπορεί να ειδοποιήσει τον ιδιοκτήτη στο κινητό του τηλέφωνο, το Κέντρο Λήψεων Σημάτων και εφόσον έχει γίνει η σχετική ρύθμιση ειδοποιείται αυτόματα και η αστυνομία.

- Αν αντιληφθούν ύποπτες κινήσεις και θορύβους κατά την διάρκεια της νύχτας, να πραγματοποιήσουν φωταψία σε ολόκληρη την οικία με το πάτημα ενός διακόπτη.
- Να ειδοποιηθούν από το σύστημα για πλημμύρα, πυρκαγιά, ακραία καιρικά φαινόμενα, βλάβες του ηλεκτρομηχανολογικού εξοπλισμού κ.α.
- Να έχουν οπτική αναπαράσταση της οικίας τους μέσω εγκατάστασης μίας ή περισσότερων καμερών οι οποίες θα μεταφέρουν την εικόνα του σπιτιού στον υπολογιστή ή στο κινητό. Όπως γίνεται αντιληπτό ένα έξυπνο σπίτι έχει ανεξάντλητες δυνατότητες σε θέματα ασφαλείας οι οποίες προσαρμόζονται στις εκάστοτε ανάγκες των ενοίκων.

1.2) ΑΝΕΣΗ

Ένα έξυπνο σπίτι προσφέρει μοναδικές συνθήκες άνεσης με το πλήθος λειτουργιών που διαθέτει. Μέσα από το σύστημα εγκατάστασης δίνεται η δυνατότητα να ενεργοποιηθούν ή απενεργοποιηθούν πολλές λειτουργίες της οικίας με την χρήση κινητού τηλεφώνου όπως να τίθεται σε λειτουργία ο θερμοσίφωνας πριν φτάσετε στο σπίτι, να ανάβουν τα εξωτερικά φώτα του σπιτιού όταν βρίσκεστε κοντά, να ρυθμίζετε την θέρμανση, να κλείνετε την παροχή ρεύματος σε κάποια συσκευή που αφήσατε ανοιχτή κ.α. Με αυτόν τον τρόπο αποκτάται ο απόλυτος έλεγχος της οικίας σας ρυθμίζοντας πολλές από τις λειτουργίες του σπιτιού από το κινητό σας τηλέφωνο.

Επιπλέον, για μεγαλύτερη άνεση και ευκολία μπορούν να προγραμματιστούν πιθανά σενάρια τα οποία εφαρμόζονται με το πάτημα ενός πλήκτρου στο κινητό ή με την λειτουργία έναν διακόπτη. Ενδεικτικά κάποια από τα σενάρια μπορεί να είναι:

«Φεύγω»

Όταν φεύγετε από το σπίτι να απενεργοποιούνται οι ηλεκτρολογικές συσκευές, η θέρμανση, η ύδρευση, το φυσικό αέριο, να ενημερώνεστε αν όλες οι πόρτες και τα παράθυρα είναι κλειδωμένα, να ενεργοποιείται ο συναγερμός κ.α.

«Διακοπές»

Όταν απουσιάζετε για διακοπές να ενεργοποιούνται / απενεργοποιούνται σε τυχαίες και λογικές ώρες ηλεκτρικές συσκευές και φώτα προκαλώντας την αίσθηση σε πιθανούς διαρρήκτες ότι το σπίτι κατοικείται, να ενημερώνεται ο ιδιοκτήτης για πιθανούς κινδύνους και καταστροφές εποπτεύοντας με κάμερες τον εσωτερικό χώρο.

«Έρχομαι»

Η επιστροφή στο σπίτι να συνεπάγεται κατόπιν δική σας θελήσεως την ενεργοποίηση ηλεκτρικών συσκευών όπως θέρμανσης, κλιματιστικού, θερμοσίφωνα κ.α. Συνεπώς ο κατάλληλος προγραμματισμός του συστήματος με βάση τις καθημερινές ανάγκες βελτιστοποιεί την ποιότητα της ζωής σας.

1.3) ΟΙΚΟΝΟΜΙΑ

Με το έξυπνο σπίτι και την ορθολογική διαχείριση των ηλεκτρομηχανολογικών εγκαταστάσεων επιτυγχάνεται μια σειρά από οικονομίες.

Συγκεκριμένα:

- Η κατάλληλη χρήση των διαθέσιμων αυτοματισμών για την θέρμανση μπορεί να εξασφαλίσει σημαντική μείωση στην κατανάλωση ενέργειας. Με αυτόν τον τρόπο η άσκοπη σπατάλη ενέργειας, όταν ο ένοικος για παράδειγμα απουσιάζει από την οικία του ή τα παράθυρα είναι ανοιχτά, μειώνεται βαθμιαία.
- Ελαχιστοποιείται το κόστος λειτουργίας της ηλεκτρομηχανολογικής εγκατάστασης απενεργοποιώντας ηλεκτρικές συσκευές και φωτά που δεν χρησιμοποιούνται από τους ένοικους.
- Αυξάνεται ο χρόνος ζωής των μηχανημάτων και ταυτόχρονα μειώνονται τα έξοδα συντήρησης τους. Η συνολική εξοικονόμηση ενέργειας για τους ιδιοκτήτες των έξυπνων σπιτιών εκτιμάται ότι ανέρχεται στο 35% κατά μέσο όρο. Επιπλέον, ένα ενσωματωμένο σύστημα αυτοματισμού αυξάνει την αξία ενοικίασης και πώλησης της οικίας προσφέροντας άνετη διαβίωση και μεγαλύτερη εξοικονόμηση ενέργειας και εσόδων από ένα συμβατικό σπίτι.

Κεφάλαιο 2

2) Πρωτόκολλα επικοινωνίας

Σε έναν ασφαλή συνδεδεμένο κόσμο, τα έξυπνα προϊόντα πρέπει να μπορούν να ανταλλάσσουν πληροφορίες με πολλαπλές συσκευές. Για να γίνει αυτό, πρέπει να μιλούν την ίδια γλώσσα, δηλαδή τα πρωτόκολλα επικοινωνίας. Τα πρωτόκολλα καθορίζουν πώς αποστέλλονται τα σήματα από τη μία συσκευή στην άλλη προκειμένου να ενεργοποιηθεί μια ενέργεια, όπως η ενεργοποίηση και απενεργοποίηση των φωτιστικών ή το κατέβασμα ή το ανέβασμα των ρολών σε ένα σπίτι.

IR technology- Μεταφορά πληροφορίας μέσω Υπέρυθρων

Ένα από τα απλούστερα και πιο αξιόπιστα πρωτόκολλα, που προσφέρουν συνήθως επικοινωνία μονής κατεύθυνσης. Είναι η πρώτη επιλογή για τηλεχειριστήρια, όπως αυτά για την τηλεόρασή σας.

Ethernet

Γρήγορη και αξιόπιστη ενσύρματη επικοινωνία, με εμβέλεια έως 100m και χαμηλή ευαισθησία στις ηλεκτρομαγνητικές παρεμβολές.

WiFi

Γρήγορη και αξιόπιστη ασύρματη επικοινωνία, με εύρος περίπου 25 μέτρων.

Bluetooth

Ένα ασύρματο πρωτόκολλο μικρής εμβέλειας (περίπου 10 μέτρων) που χρησιμοποιείται συχνά σε τηλέφωνα, ακουστικά και ηχεία. Το προσαρμοστικό σύστημα αναπήδησης συχνότητας εντοπίζει υπάρχοντα σήματα, όπως το WiFi, και διαπραγματεύεται έναν χάρτη καναλιών για τις συσκευές Bluetooth, προκειμένου να ελαχιστοποιηθούν οι παρεμβολές.

Thread

Ένα ασύρματο πρωτόκολλο που αναπτύχθηκε από μια ομάδα εταιρειών όπως η Nest, η Samsung, η QUALCOMM και η OSRAM. Σχεδιασμένο για να επιτρέπει στις συσκευές του πρωτοκόλλου του να επικοινωνούν ακόμη και όταν το δίκτυο WiFi είναι ασθενές.

Zigbee

Το Zigbee είναι ένα ασύρματο πρωτόκολλο που λειτουργεί σε δίκτυο κόμβων. Δηλαδή: χρησιμοποιεί μια συσκευή για να μεταδίδει ένα σήμα σε άλλες συσκευές, ενισχύοντας και επεκτείνοντας το δίκτυο. Το Zigbee μπορεί να κατασκευαστεί σε ρυθμιστές, κλειδαριές πόρτας, θερμοστάτες και πολλά άλλα. Χρησιμοποιείται για παράδειγμα σε WeMo και Philips Hue.

Z-wave

Παρόμοια με το Zigbee, το Z-Wave είναι ένα πρωτόκολλο δικτύου πλέγματος (mesh) ανοιχτού κώδικα. Από τεχνική άποψη, η κύρια διαφορά μεταξύ των δύο είναι η απόδοση δεδομένων - το κύμα Z είναι περίπου 6 φορές πιο αργό από το Zigbee. Ωστόσο, απαιτεί λιγότερη ενέργεια για να καλύψει το ίδιο εύρος με το Zigbee.

KNX

Το KNX είναι ένα ανοιχτό πρωτόκολλο που βρίσκεται στην αγορά εδώ και δεκαετίες και είναι επίσης ένα από τα πιο δημοφιλή πρωτόκολλα για τον αυτοματισμό των κτιρίων. Λειτουργεί σε περισσότερες από μία φυσικές στρώσεις π.χ. καλωδιώσεις συνεστραμμένου ζεύγους, δικτύωση ηλεκτρικής γραμμής, υπέρυθρες, Ethernet και RF.

UPB

Το Universal Powerline Bus είναι μια ασύρματη τεχνολογία, με στόχο της την αντικατάσταση του X10, προσφέροντας μεγαλύτερη αξιοπιστία. Χρησιμοποιεί το υπάρχον ηλεκτρικό δίκτυο του σπιτιού, κάτι που σημαίνει μικρότερο κόστος, προκειμένου να αποστείλει σήματα που ελέγχουν τις συσκευές εντός και εκτός οικίας. Αν και το UPB είναι αξιόπιστο σύστημα, είναι δύσκολο να συνδυαστεί με τις νεότερες ασύρματες τεχνολογίες όπως το Wi-Fi, τα smartphones κ.ο.κ. Από την άλλη, προσφέρει χαμηλό bandwidth, με συνέπεια στις επιδόσεις, ενώ δεν διαθέτει κρυπτογράφηση, οπότε δεν είναι τόσο ασφαλές όπως το wireless. Τέλος, είναι αρκετά περίπλοκο στην εγκατάστασή του.

MQTT

MQTT ή Queue Telemetry Transport είναι ένα ελαφρύ πρωτόκολλο ανταλλαγής μηνυμάτων για την μεταφορά δεδομένων σε απομακρυσμένες τοποθεσίες όπου απαιτείται "αποτύπωμα μικρού κώδικα" ή το εύρος ζώνης του δικτύου είναι περιορισμένο. 'Τρέχει' πάνω από το

TCP/IP πρωτόκολλο, με προεπιλεγμένη θύρα τη 1883. Ακόμη η χαμηλή κατανάλωση ενέργειας είναι ένας παράγοντας που συχνά επιλέγεται και για εφαρμογές κινητών και ασύρματες συσκευές. Αυτά τα πλεονεκτήματα επιτρέπουν την εφαρμογή αυτού του πρωτοκόλλου στα συστήματα M2M (Machine to Machine) και IIoT (Industrial Internet of Things).

Κεφάλαιο 3

3)Λογισμικό ανοιχτού κώδικα *HOME ASSISTANT*

Οι συσκευές που είναι διαθέσιμες για την αυτοματοποίηση ενός σπιτιού (αισθητήρες, κάμερες, κλειδαριές, διακόπτες, φώτα κ.ά.) είναι πολλές και συνεχώς αυξάνονται. Η κάθε μία συσκευή ενδεχομένως να χρησιμοποιεί διαφορετικά applications και servers ανάλογως την εταιρία. Πολλοί είναι όμως αυτοί που επιθυμούν ασφάλεια και δεν θέλουν να εξαρτώνται από servers και cloud services τρίτων καθώς και να μην έχουν πολλές εφαρμογές στα κινητά τους τηλέφωνα για τον έλεγχο αυτών. Επίσης είναι ελκυστική η ιδέα ενός ενιαίου Hub για όλες τις συσκευές και αισθητήρες με χαμηλό κόστος. Η λύση γι' αυτούς τους χρήστες δεν είναι άλλη από τα **λογισμικά ανοιχτού κώδικα** τα οποία χρησιμοποιούνται για την αυτοματοποίηση της οικιακής χρήσης και τα οποία μπορούν να τρέξουν από έναν ηλεκτρονικό υπολογιστή μέχρι και σε ένα Raspberry Pi. Τέτοιου είδους λογισμικά είναι π.χ. το Home Assistant, το Calaos, το Domoticz, το OpenMotics, το OpenHab κ.ά. .

3.1) Οδηγός Εγκατάστασης Home Assistant σε Raspberry Pi



3.2)Εισαγωγή

Όπως είδαμε πιο πάνω υπάρχουν πολλές εναλλακτικές για την εγκατάσταση και λειτουργία ενός «έξυπνου» σπιτιού. Διάφορες εταιρείες έχουν τη δική τους λύση, είτε αυτή είναι απλή και φθηνή είτε ακριβή. Άλλη με καλύτερους αισθητήρες και άλλη με περισσότερες λειτουργίες. Πως μπορούμε όμως να εκμεταλλευτούμε όλες τις δυνατότητες των αισθητήρων και συσκευών? Πως θα επιτύχουμε το καλύτερο δυνατό αποτέλεσμα και πως θα το φέρουμε στα μετρά μας? Ποιο το κόστος γι' όλα αυτά?

Σ' αυτά τα ερωτήματα έρχεται να δώσει απάντηση και λύση η ηλεκτρονική πλατφόρμα ή λειτουργικό ανοιχτού κώδικα Home Assistant.

3.2.1)Τι είναι το Home Assistant

Το Home Assistant είναι μια πλατφόρμα ανοιχτού κώδικα (open source) διαθέσιμο για όλους και με μηδενικό κόστος. Προορίζεται για οικιακή χρήση και έχει ως στόχο τον τοπικό έλεγχο των αισθητήρων, των έξυπνων συσκευών και αυτοματισμών μιας οικίας δίνοντας προτεραιότητα στην ιδιωτικότητα και το ιδιωτικό απόρρητο. Υποστηρίζεται διαρκώς από μια μεγάλη ηλεκτρονική κοινότητα και λάτρεις των DIY λύσεων ανά τον κόσμο. Παράλληλα συνεργάζεται αρμονικά με την γνωστή πλατφόρμα GitHub. Δεν απαιτεί ιδιαίτερες γνώσεις και δεξιότητες, πάρα μόνο χρόνο, διάθεση και φαντασία. Έχει κατάλληλες παραμετροποιήσεις για να «τρέχει» σε ένα Raspberry Pi, σε έναν τοπικό διακομιστή (server) ή γενικά σε οποιοδήποτε μηχάνημα έχει τη δυνατότητα να τρέξει Python 3. Μπορεί επίσης να ενσωματωθεί σε οποιοδήποτε σύστημα σε ένα Docker container.

Με τον όρο ιδιωτικότητα εννοούμε ότι είμαστε ανεξάρτητοι από τις βάσεις δεδομένων και τους διακομιστές τρίτων εταιρειών και είμαστε αυτοί που έχουμε τον απόλυτο έλεγχο των δράσεων και των δεδομένων μας.

Η πλατφόρμα μας είναι συμβατή με μια τεράστια λίστα εταιρειών, υπηρεσιών και πρωτοκόλλων η οποία διαρκώς εμπλουτίζεται και επεκτείνεται. Η υποστήριξη της πλατφόρμας είναι μεγάλη και εκτείνεται από την κύρια σελίδα του λογισμικού, σε διάφορα σχετικά φόρουμ καθώς και στο γνωστό σε όλους YouTube, όπου ο χρήστης μπορεί να αναζητήσει και να βρει ετοιμους οδηγούς για το πως θα χτίσει το σύστημα του. Στη συνέχεια θα δούμε αναλυτικά την εγκατάσταση του λειτουργικού Home Assistant σε ένα Raspberry Pi και την προσθήκη κάποιων σημαντικών πρόσθετων add-on που είναι βασικά για να ξεκινήσει κάποιος.

3.2.2)Τι χρησιμοποιήσαμε

- 1 * Raspberry Pi 3B+.
- 1 * φορτιστή/τροφοδοτικό με προδιαγραφές 5 Volts στα 2 Ampere.
- *Μπορεί να χρησιμοποιηθεί και οποιοσδήποτε φορτιστής smartphone , τύπου micro USB.
- 1 * κάρτα μνήμης τύπου micro SD χωρητικότητας 32GB .
- 1* card reader για την σύνδεση της SD card σε έναν ηλεκτρονικό υπολογιστή (εφόσον δεν διαθέτει ο υπολογιστής μας) .
- 1 * θήκη για να ασφαλίσουμε το raspberry μας.
- 1 * καλώδιο τύπου Ethernet (δίνεται και η δυνατότητα σύνδεσης μέσω WIFI).
- 1 * USB drive.
- (Προαιρετικό) μικρές ψήκτρες για τους επεξεργαστές του Raspberry.
- Και ένας ηλεκτρονικός υπολογιστής

3.2.3) Τι είναι το Raspberry Pi

Πρώτου προχωρήσουμε με τον οδηγό ας δούμε τι είναι το βασικό μηχάνημα μας που ακούει στο όνομα Raspberry.

Το Raspberry Pi είναι ένας πλήρης υπολογιστής με μέγεθος λίγο μεγαλύτερο από πιστωτική κάρτα. Το πρώτο μοντέλο της σειράς δημιουργήθηκε στο Ηνωμένο Βασίλειο από την ομώνυμη εταιρεία Raspberry Pi Foundation με σκοπό την προώθηση της γνώσης της επιστήμης των υπολογιστών σε σχολεία αλλά και σε αναπτυσσόμενες χώρες. Γι' αυτό το λόγο η συγκεκριμένη συσκευή έχει εξαιρετικά χαμηλό κόστος. Το πρώτο μοντέλο της σειράς, Raspberry Pi 1, κυκλοφόρησε στην αγορά το 2012 και γνώρισε μεγάλη επιτυχία πράγμα που οδήγησε τους δημιουργούς του στη συνέχεια της σειράς.

Σήμερα το πιο ισχυρό μοντέλο είναι το Raspberry Pi 3B+ (Plus). Παρά τον μικρό όγκο του, διαθέτει τετραπύρρηνο επεξεργαστή Cortex-A53 (ARMv8) 64-bit SoC στα 1.4GHz, διπύρρηνη κάρτα γραφικών, 1GB μνήμης RAM, τέσσερις θύρες USB, έξοδο HDMI, υποστηρίζει ασύρματη σύνδεση μέσω Bluetooth και Wi-Fi. Τροφοδοτείται μέσω Micro USB, και έχει 40 pins GPIO γενικής χρήσης για σύνδεση με άλλα ηλεκτρονικά και περιφερειακά. Αυτά τα χαρακτηριστικά μας υπερκαλύπτουν γι' αυτό που θα ακολουθήσει.

3.3) Τρόποι εγκατάστασης

Οι βασικοί τρόποι εγκατάστασης είναι 3.

1. Hassbian

Είναι ένας εύκολος τρόπος εγκατάστασης για την εγκατάσταση του Home Assistant (HA) σε Raspberry Pi (RPi). Είναι ένα αρχείο disk image (.img) που έχει ενσωματωμένο το HA. Πρόκειται για ένα λειτουργικό που μπορεί να εγκατασταθεί και να τρέξει όχι μόνο σε μια micro SD αλλά και σε ένα USB drive. Με αυτόν τον τρόπο αν είχαμε ένα RPi που να έκανε boot από USB, θα μπορούσαμε να επωφεληθούμε απ' αυτόν τον τρόπο εγκατάστασης. Οδηγίες εγκατάστασης μπορούν να βρεθούν στο σύνδεσμο:

<https://www.home-assistant.io/docs/installation/hassbian/installation/>.

2. Εγκατάσταση σε Docker

Το Docker είναι πρόγραμμα που τρέχει σε ηλεκτρονικούς υπολογιστές με γνωστά λειτουργικά όπως Linux, Windows, MacOS και μπορεί να τρέξει εικονικά διάφορα λειτουργικά και εφαρμογές. Πραγματοποιεί Operating System-level virtualization στο οποίο ο πυρήνας (kernel) επιτρέπει την ύπαρξη πολλών απομονωμένων εργασιών/πόρων. Αυτά τα απομονωμένα κελιά ονομάζονται containers.

Η εγκατάσταση του HA μπορεί να γίνει σε ένα container ανεξάρτητα του κυρίου λειτουργικού του μηχανήματος εγκατάστασης με προσαρμοσμένες εντολές που μπορούν να βρεθούν στον ακόλουθο σύνδεσμο:

<https://www.home-assistant.io/docs/installation/docker/> .

Το θετικό σε αυτόν τον τρόπο εγκατάστασης είναι ότι μπορεί να εγκατασταθεί και σε έναν ηλεκτρονικό υπολογιστή όπου ο σκληρός δίσκος του και η μνήμη του θα μπορούν να τρέξουν πιο γρήγορα το home assistant σαν εφαρμογή, με μειωμένες πιθανότητες για αστοχία υλικού

(όπως π.χ. μια υπερθέρμανση στο RPI) και ωστόσο θετικό είναι επίσης και η ασφάλεια που σου παρέχει σε περίπτωση λάθους, όπου ο χρήστης έχει τη δυνατότητα να διαγράψει το αντίστοιχο container χωρίς να χρειαστεί να πειράξει το βασικό λειτουργικό του. Παρόλα αυτά είναι μια δύσκολη μέθοδος και απαιτεί κάποιες εξειδικευμένες γνώσεις, όχι μόνο για την εγκατάσταση αυτή καθ' αυτήν, καθώς και στον τρόπο διαχείρισης του.

3.Εγκατάσταση σε Ιδεατή Μηχανή (Virtual Machine)

Είναι μια δύσκολη διαδικασία για να την περιγράψουμε και απαιτεί και εξοικείωση με το αντικείμενο. Πληροφορίες και οδηγίες μπορούν να βρεθούν εδώ:

<https://www.home-assistant.io/blog/2017/11/29/hassio-virtual-machine/>.

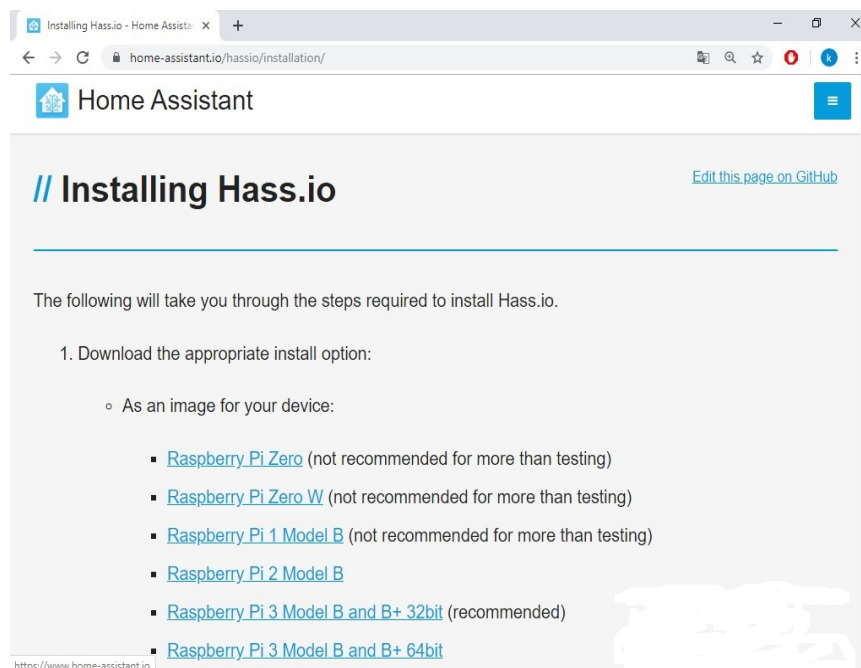
4. Hass.io

Είναι ο πιο εύκολος τρόπος να εγκαταστήσουμε το λειτουργικό μας. Είναι ένα αυτόνομο λειτουργικό. Είναι φιλικό προς το χρήστη και η διαχείριση των προσθέτων γίνεται πιο ευκολά σε σχέση με τις παραπάνω μεθόδους. Αυτός είναι ο τρόπος με τον οποίο επιλέξαμε να εγκαταστήσουμε το Home Assistant μας και θα ακολουθήσει αναλυτικός οδηγός με βήματα. Στη συνέχεια θα ακολουθήσουν βήματα για την εγκατάσταση και ρύθμιση κάποιων βασικών προσθέτων add-on. Έτσι θα έχουμε έναν πλήρη οδηγό για κάποιον που θέλει να κάνει τα πρώτα βήματα.

3.3.1)Οδηγός εγκατάστασης Home Assistant με τη μέθοδο του Hass.io

Βήμα 1: Επισκεπτόμαστε την επίσημη σελίδα του λειτουργικού μας και κατεβάζουμε την αντίστοιχα κατάλληλη έκδοση. Στην περίπτωση μας επιλέγουμε να κάνουμε εγκατάσταση σε RPi 3B+ όπου έχουμε 2 επιλογές για 32bit ή 64bit. Εμείς επιλέξαμε την 64bit έκδοση, αρά κατεβάζουμε την ακόλουθη έκδοση από τον ακόλουθο σύνδεσμο:

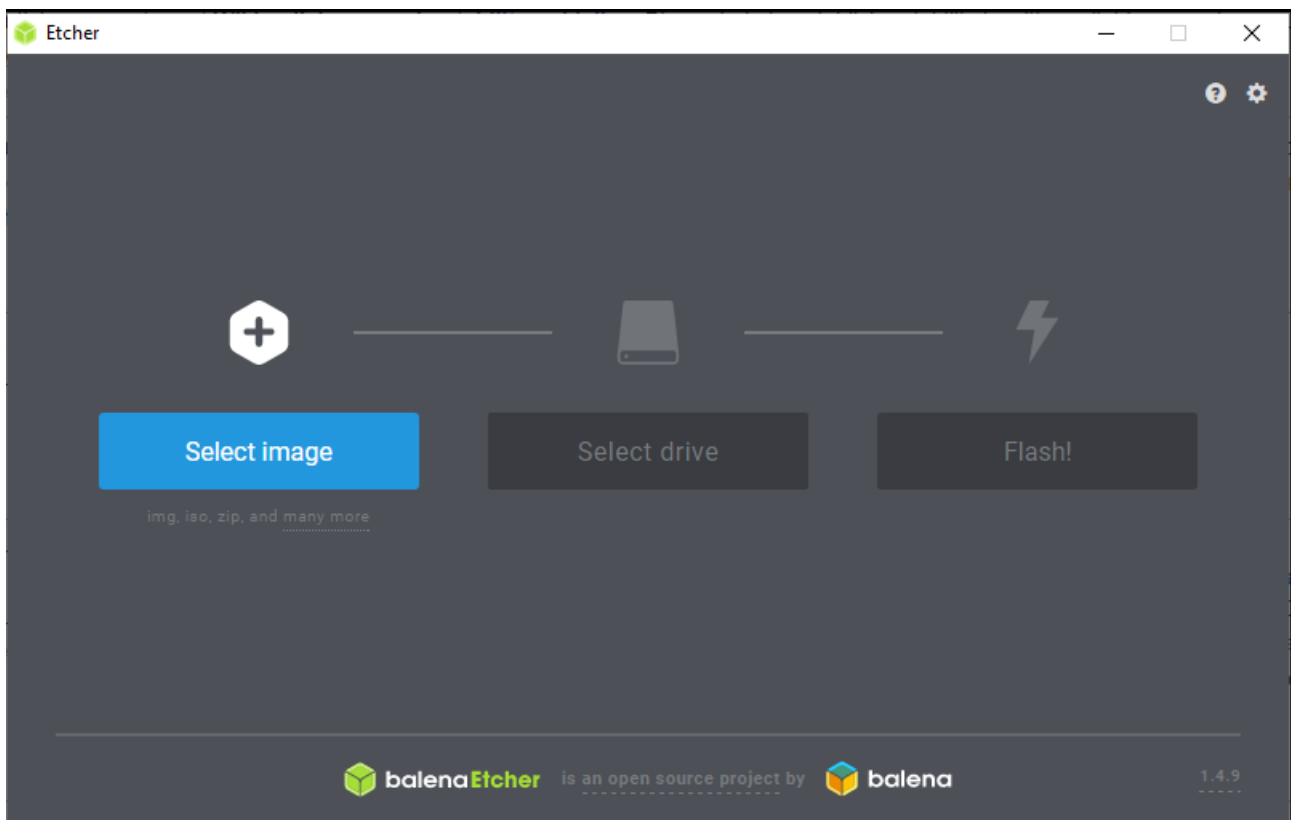
<https://www.home-assistant.io/hassio/installation/>



Βήμα 2: Έχοντας κατεβάσει τη σωστή έκδοση του βήματος 1, χρησιμοποιούμε ένα πρόγραμμα αποσυμπίεσης αρχείων για να εξάγουμε το κατάλληλο αρχείο προς εγγραφή στην κάρτα SD. Στην περίπτωση μας επιλέξαμε το εργαλείο WinRAR.
<https://www.win-rar.com/download.html?&L=0>

Βήμα 3: Σ αυτό το βήμα θα χρειαστούμε κατάλληλο πρόγραμμα για διαμόρφωση και εγγραφή σε κάρτα SD. Επιλέξαμε το εργαλείο Etcher.
<https://www.balena.io/etcher/>

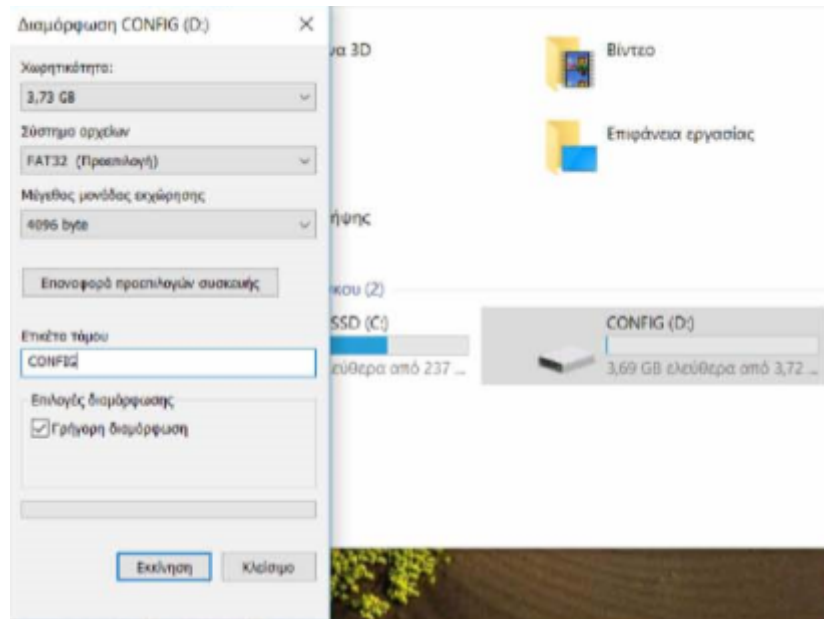
Αφού εισάγουμε την κάρτα SD στο μηχάνημα μας ή στο card reader μας ανοίγουμε το πρόγραμμα Etcher. Εκεί επιλέγουμε Select Image και αναζητούμε το αρχείο μας Hass.io που εξήγησαμε από την αποσυμπίεση στο βήμα 2. Στη συνέχεια επιλέγουμε το σωστό δίσκο προς εγκατάσταση, στην περίπτωση μας την SD, επιλέγουμε Flash και περιμένουμε μέχρι να ολοκληρωθεί η διαδικασία.



Η κάρτα SD μας είναι έτοιμη! Την αφαιρούμε από το card reader μας και είναι έτοιμη για εισαγωγή στο RPi μας.

Βήμα 4: (Αυτό το βήμα είναι προαιρετικό και απευθύνεται σε όσους σκοπεύουν να συνδέσουν το RPi τους με WIFI). Γι' αυτό το βήμα θα χρειαστούμε ένα flash drive ανεξάρτητου μεγέθους μέσα στο οποίο θα δημιουργήσουμε ένα αρχείο με τις απαραίτητες ρυθμίσεις για το WIFI. Επίσης μας δίνεται η δυνατότητα να δώσουμε στο RPi μας μία static IP address.

Αρχικά συνδέουμε το USB drive στον υπολογιστή μας και κάνουμε διαμόρφωση (format) αυτού με "Ετικέτα τόμου (Volume Label)"- **CONFIG** και "Σύστημα αρχείων(File System)"-**FAT32**.



Επιλέγουμε εκκίνηση(Start) και περιμένουμε μερικά δευτερόλεπτα μέχρι να ολοκληρωθεί η διαδικασία. Στη συνέχεια μέσα στο Flash Drive μας δημιουργούμε ένα φάκελο με όνομα network και στη συνέχεια μέσα σ αυτόν δημιουργούμε ένα αρχείο με όνομα my-network. ΠΡΟΣΟΧΗ! Αυτό το αρχείο ΔΕΝ πρέπει να έχει επέκταση/τύπο. Ανοίγοντας το αρχείο κάνουμε αντιγραφή και επικόλληση ένα από τα παρακάτω κείμενα.

<https://github.com/home-assistant/hassos/blob/dev/Documentation/network.md>

```
[connection]
id=hassos-network
uuid=YOUR_UUID_NUMBER
type=802-11-wireless

[802-11-wireless]
mode=infrastructure
ssid=YOUR_WIFI_NAME

[802-11-wireless-security]
auth-alg=open
key-mgmt=wpa-psk
psk=YOUR_WIFI_PASSWORD

[ipv4]
method=auto

[ipv6]
addr-gen-mode=stable-privacy
method=auto
```


Ενώ για Static IP αντικαθιστούμε και τα παρακάτω:

```
[ipv4]  
method=manual  
address=192.168.1.111/24,192.168.1.1  
dns=8.8.8.8;8.8.4.4;
```

Στο πεδίο SSID πληκτρολογούμε το όνομα του WIFI (USERNAME), ενώ στο πεδίο psk τον κωδικό που έχουμε ορίσει για το δίκτυο μας (PASSWORD). Για static IP αντικαθιστούμε με τα παραπάνω στην ενότητα [ipv4]. Για το UUID επισκεπτόμαστε τη σελίδα <https://www.uuidgenerator.net> και δημιουργούμε το δικό μας.

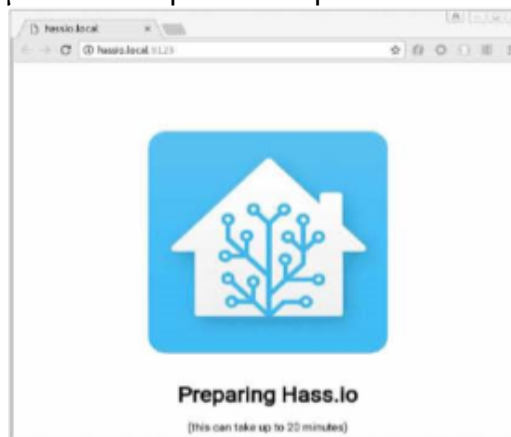


Τέλος αποθηκεύουμε το αρχείο και αφαιρούμε το USB drive από το μηχάνημα μας.

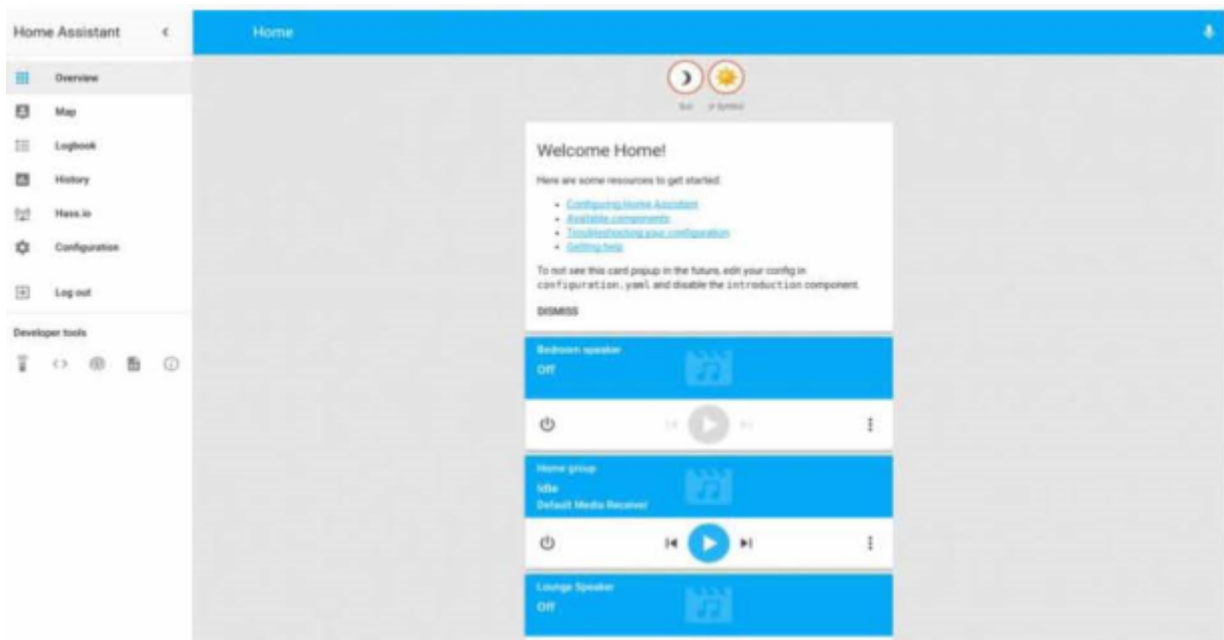
Βήμα 5: Τώρα εισαγάγουμε την κάρτα SD και το USB στο RPi μας. Εν συνεχεία το συνδέουμε στο τροφοδοτικό του και στο καλώδιο ethernet αν δεν έχουμε επιλέξει ασύρματο τρόπο σύνδεσης. Το RPi μας ξεκινάει, συνδέεται στο δίκτυο και κατεβάζει την τελευταία έκδοση καθώς και τις τελευταίες ενημερώσεις του Home Assistant. Μετά από περίπου 20 λεπτά (εξαρτάται από την ταχύτητα δικτύου μας), θα είμαστε έτοιμοι να συνδεθούμε στην πλατφόρμα μας.

Βήμα 6: Ανοίγουμε οποιονδήποτε φυλλομετρητή και πληκτρολογούμε την ακόλουθη διεύθυνση <http://hassio.local:8123> ή <http://192.168.1.XXX:8123> όπου XXX είναι η IP του subnet που δώσαμε στο βήμα 3 δημιουργώντας την static IP του RPi μας. Επίσης μπορούμε πλέον να αφαιρέσουμε και το USB drive από το Raspberry μας.

Η πρώτη εικόνα που θα δούμε θα είναι η ακόλουθη

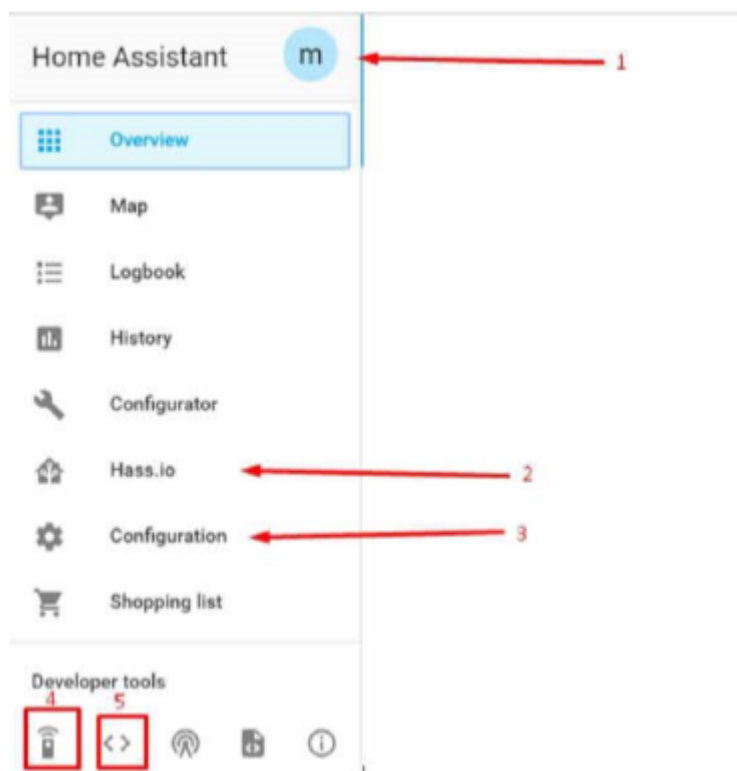


Ενώ μετά από λίγο και αν όλα πήγαν καλά θα δούμε την αρχική σελίδα του συστήματος μας

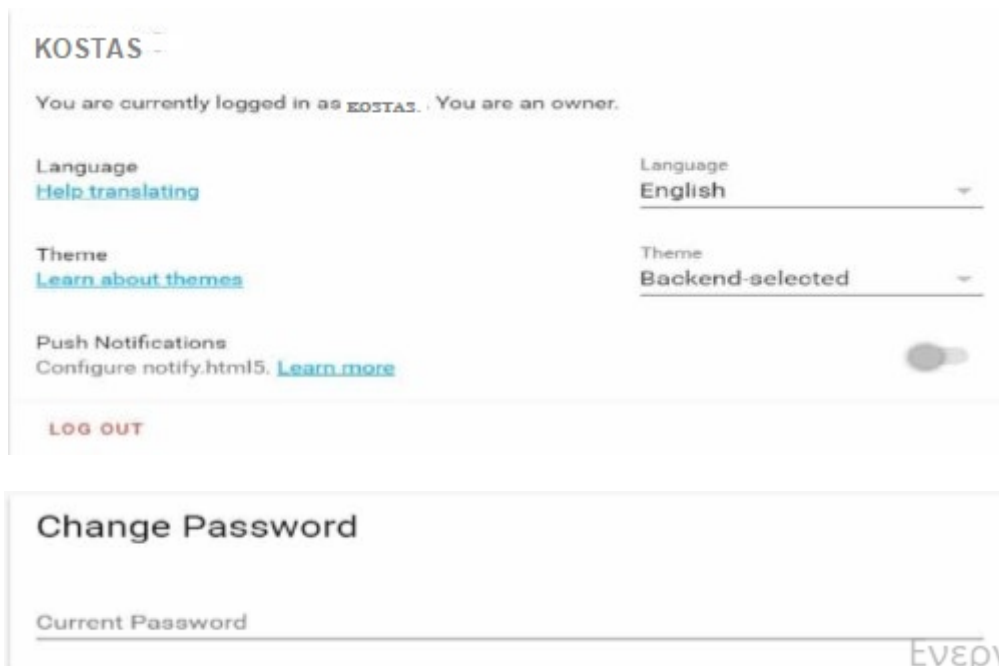


Πλέον είμαστε έτοιμοι να περιηγηθούμε στα μενού του και να ξεκινήσουμε την εγκατάσταση κάποιων βασικών προσθέτων add-on που θα μας προσδώσουν ασφάλεια και ευκολία στη διαχείριση του.

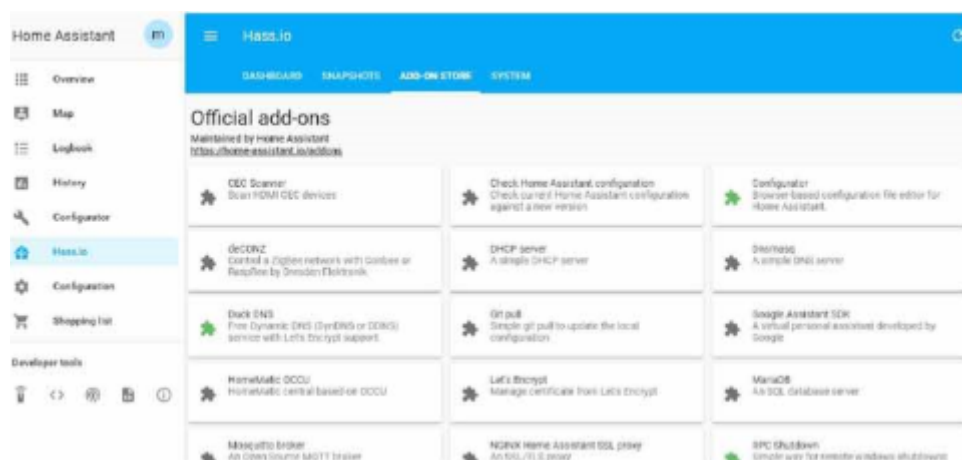
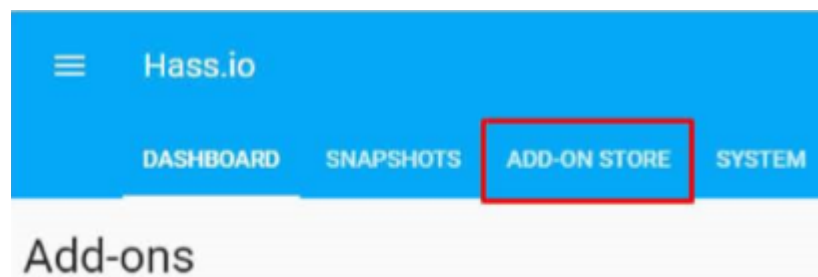
3.3.2) Ξενάγηση στο Μενού



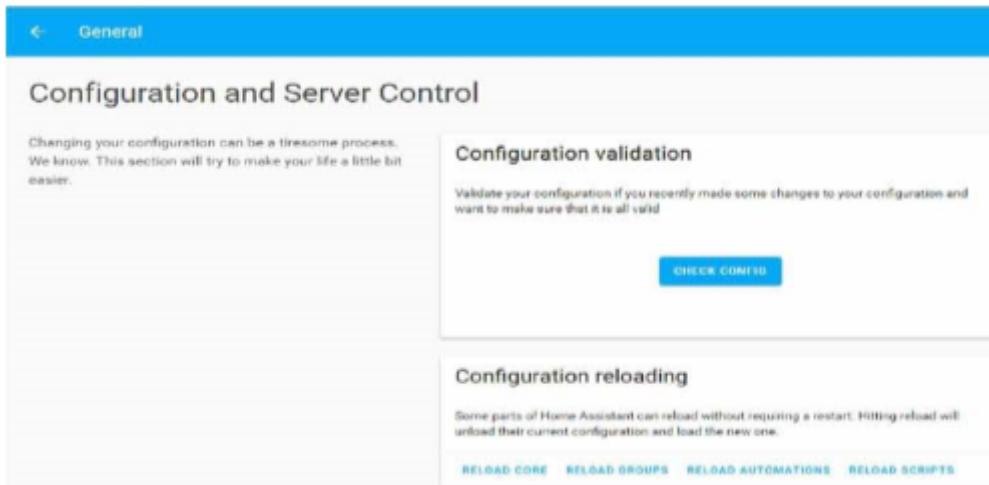
1. Πατώντας κάποιος πάνω στο κυκλικό σχήμα έχει τη δυνατότητα να αλλάξει τη γλώσσα του συστήματος, το θέμα αυτού καθώς και τον κωδικό πρόσβασης που θα δημιουργήσουμε αργότερα



2. Εδώ έχουμε μια από τις βασικές καρτέλες. Ο χρήστης έχει την επιλογή να περιηγηθεί και να διαχειριστεί τα προσθέτά του, καθώς και να προσθέσει άλλα. Εδώ ακόμα ειδοποιούμαστε για τυχόν ενημερώσεις του συστήματος του.



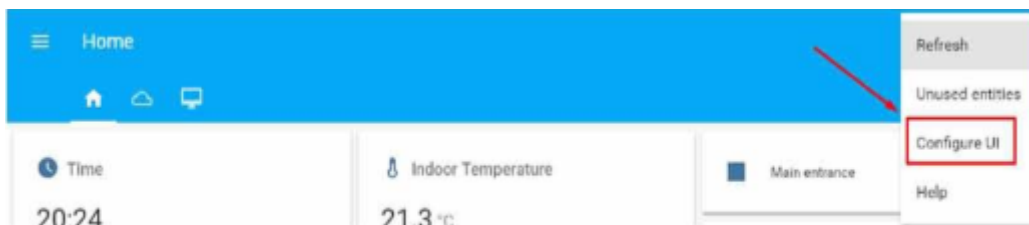
3. Και εδώ έχουμε μια πολύ σημαντική καρτέλα. Επιλέγοντας Configuration -> General μπορούμε να δούμε την κατάσταση του συστήματος μας πατώντας Check Config . Αν πάρουμε την ένδειξη Valid Config σημαίνει ότι όλα είναι σωστά και μπορούμε να λειτουργήσουμε το σύστημα μας.



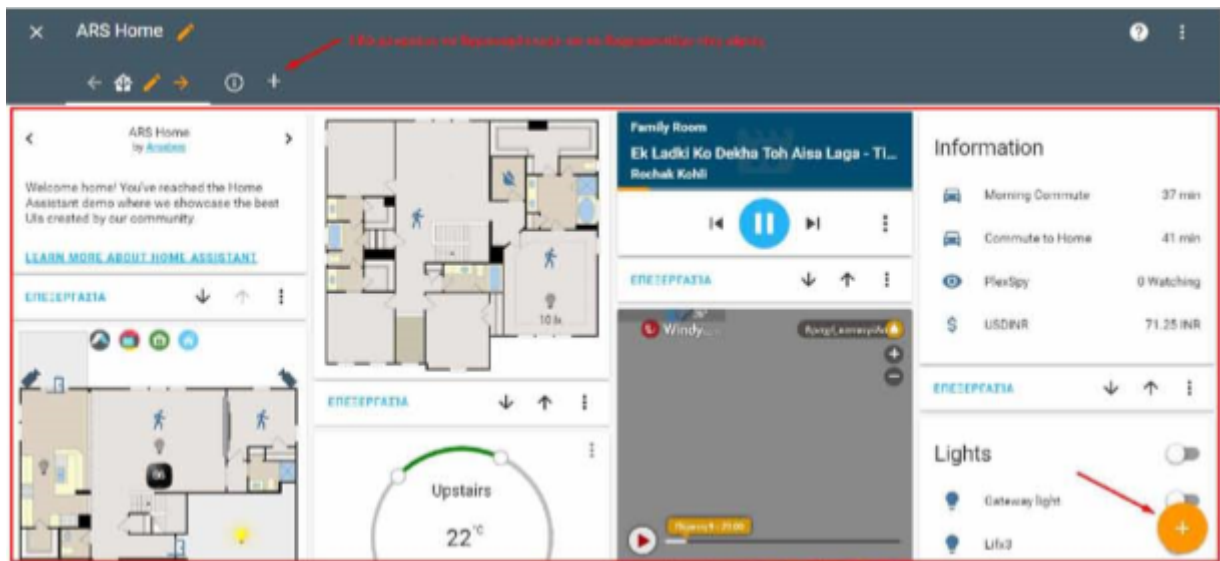
4. Εδώ μπορούμε να καλέσουμε διάφορες υπηρεσίες συστήματος με την πιο βασική να είναι `homeassistant.restart`. Αυτή θα πρέπει να καλείται μετά από σχεδόν κάθε αλλαγή στο configuration μας. Τέτοιες αλλαγές θα δούμε στη συνέχεια.

5. Σ' αυτήν την επιλογή μπορούμε να βρούμε όλα τα entities, όπως αποκαλούνται, που είναι όλοι οι αισθητήρες, υπηρεσίες και αυτοματισμοί.

Τέλος πατώντας στις τρεις κουκίδες που βρίσκονται στην πάνω δεξιά γωνιά της αρχικής μας σελίδας βρίσκουμε την επιλογή Configure UI οπότε μπορούμε να επεξεργαστούμε το User Interface της πλέον εφαρμογής μας.



Επίσης μπορούμε να προσθέσουμε/δημιουργήσουμε κάρτες και καρτέλες. Μια κάρτα μπορεί να περιέχει αισθητήρες, υπηρεσίες ή αυτοματισμούς, ενώ η καρτέλα αποτελείται από κάρτες.



Διαχείριση μιας κάρτας με έξυπνα φωτιστικά.



Δημιουργία νέας κάρτας και έτοιμοι τύποι καρτών. Με την επιλογή **MANUAL CARD** μπορούμε να δημιουργήσουμε μια δική μας κάρτα.

Αφού πραγματοποιήσαμε μια μικρή περιήγηση και επεξήγηση στα μενού τώρα μπορούμε να προχωρήσουμε στην εγκατάσταση των τριών βασικών add-ons.

3.4) Εγκατάσταση βασικών προσθέτων

3.4.1) Εγκατάσταση Configurator add-on

Με αυτό το πρόσθετο μας δίνεται η δυνατότητα να βλέπουμε όλα τα αρχεία στο σύστημα μας, καθώς και να τα επεξεργαζόμαστε. Ας δούμε όμως πρώτα τι αρχεία είναι αυτά, τι περιέχουν, σε τι γλώσσα είναι γραμμένα, αλλά και τι πρέπει να προσέχουμε.

Αρχεία .yaml και configuration.yaml

Το Home Assistant κάνει χρήση του συντακτικού YAML για τη σύνθεση του(configuration). Το YAML μπορεί να χρειάζεται λίγο χρόνο για να το συνηθίσει κάποιος αλλά είναι πραγματικά ένα ισχυρό εργαλείο που μας επιτρέπει να εκφράζουμε με απλό τρόπο τις συνθέσεις μας(configuration).

Για οτιδήποτε θέλουμε να χρησιμοποιήσουμε στο Home Assistant μας πρέπει να προσθέτουμε κώδικα της μορφής yaml στο αρχείο configuration.yaml για να προσδιορίζουμε τις ρυθμίσεις του. Ένα παράδειγμα είναι το ακόλουθο:
π.χ. configuration για έξυπνη λάμπα yeelight.

```
yeelight:  
  devices:  
    192.168.1.25:  
      name: Living Room
```

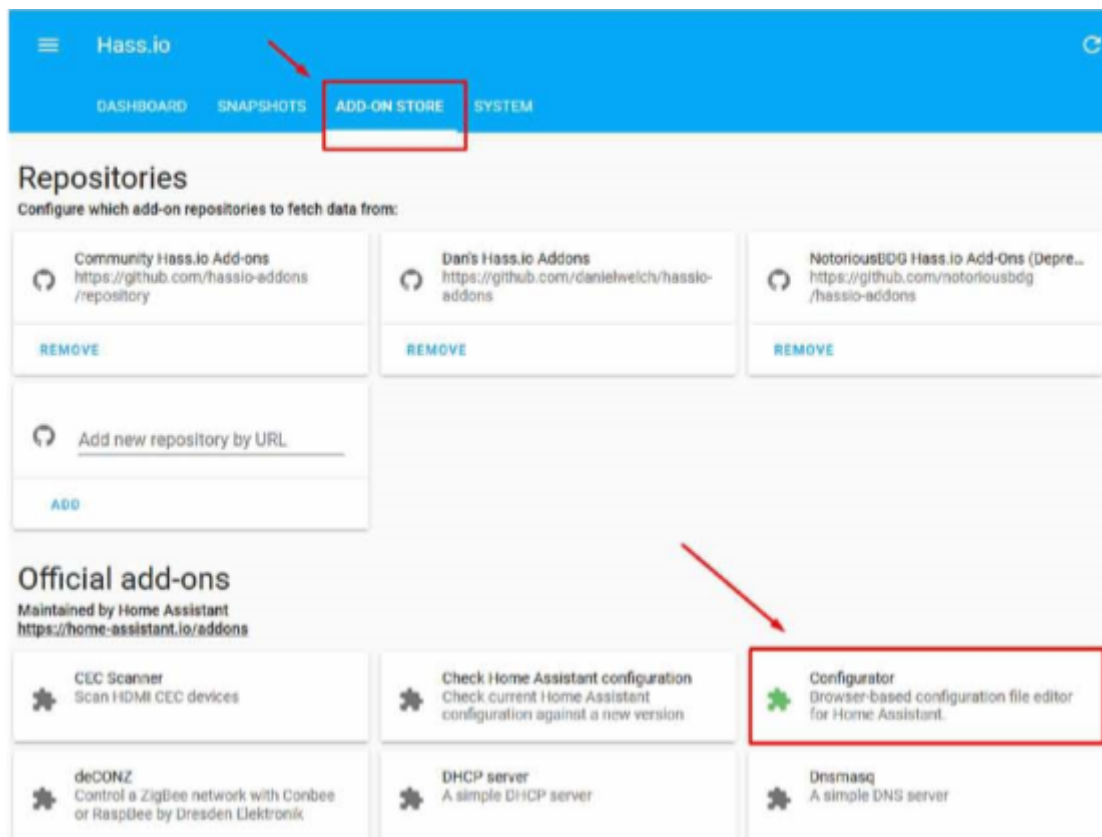
Στο συγκεκριμένο παράδειγμα γίνεται η δήλωση μιας «έξυπνης λάμπας». Πρέπει να προσέχουμε δύο πράγματα. Στο συντακτικό YAML υπάρχει διάκριση μεταξύ πεζών και κεφαλαίων χαρακτήρων, ενώ διάκριση υπάρχει μεταξύ του κενού χαρακτήρα space και του περιθωρίου Tab. Δεν πρέπει να χρησιμοποιούμε το Tab αλλιώς θα προκαλείται σύγχυση στην σύνταξη των εντολών.

Ότι ενέργεια κάνουμε στο αρχείο configuration.yaml θα πρέπει να ελέγχουμε αν έγινε σωστά και μετά για να εφαρμοστούν οι αλλαγές μας κάνουμε επανεκκίνηση του λειτουργικού μας Home Assistant. Αυτά γίνονται όπως περιγράψαμε παραπάνω στην ξενάγηση μας στις επιλογές 3 και 4 αντίστοιχα.

Περισσότερες πληροφορίες για το συντακτικό YAML:

<https://www.home-assistant.io/docs/configuration/yaml/>

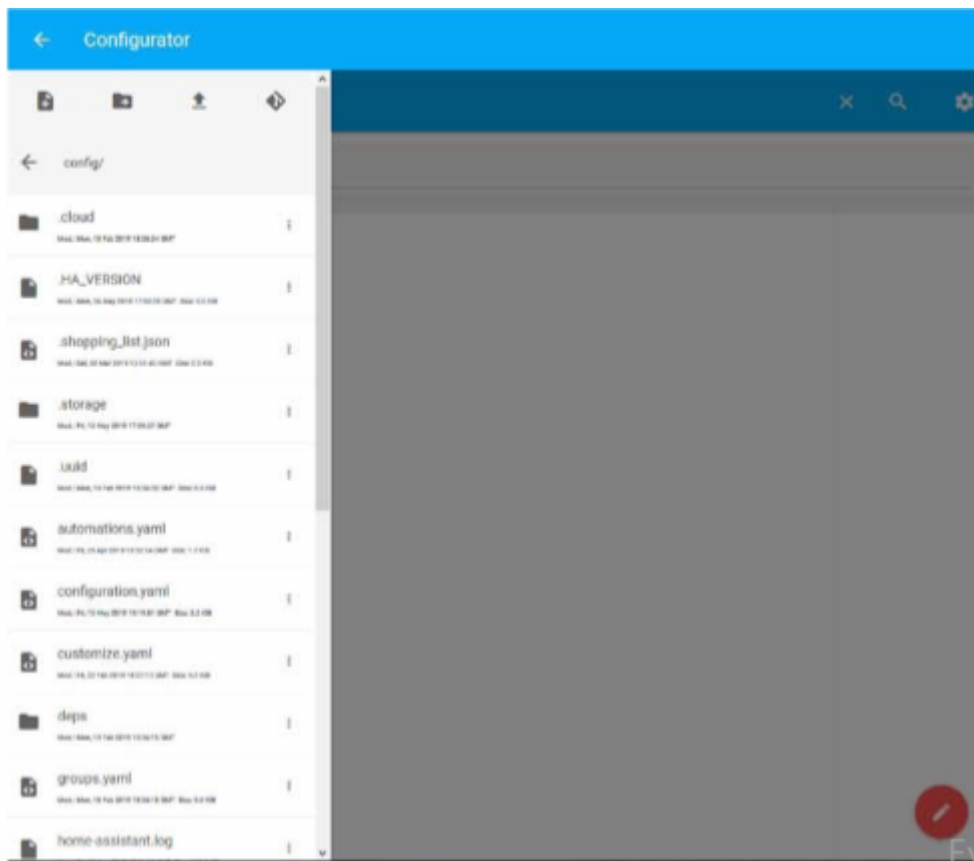
Τώρα μπορούμε να συνεχίσουμε με την εγκατάσταση του Configurator add-on.
Από τον αριστερό πίνακα στην κεντρική σελίδα επιλέγουμε Hass.io. Επειτα πατάμε στην καρτέλα ADD-ON STORE και αφού περιηγηθούμε στη σελίδα εντοπίζουμε και ανοίγουμε το πρόσθετο Configurator. Στη συνέχεια πατάμε install.



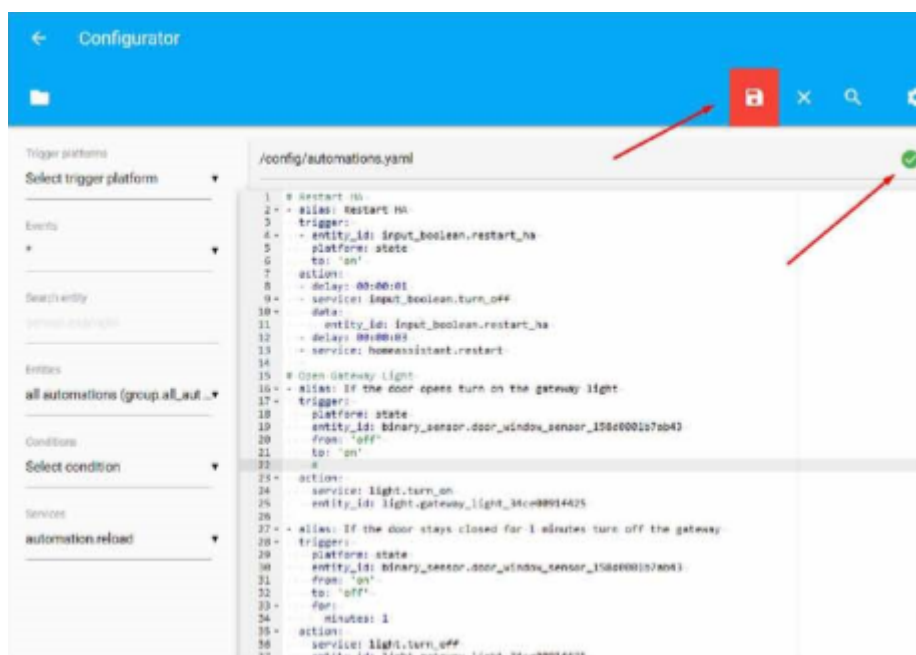
Αφού έχουμε κάνει τα παραπάνω βλέπουμε τα εξής:



Αμέσως μετά επιλέγουμε Start και είμαστε έτοιμοι. Πατώντας OPEN WEB UI έχουμε πρόσβαση σε όλους τους φακέλους και στα αρχεία αυτών.



Πλέον επιλέγοντας το αρχείο που επιθυμούμε, μπορούμε να το επεξεργαστούμε και να το αποθηκεύσουμε. Ένα άλλο σημαντικό αρχείο είναι το automations.yaml οπού σε αυτό μπορούμε να γράφουμε όλους μας τους αυτοματισμούς, όπως δηλώνει και το όνομα του.



Μετά από κάθε αλλαγή βλέπουμε με πράσινο «τικ» αν το συντακτικό μας είναι σωστό, ενώ με την κόκκινη δισκέτα κάνουμε εκ νέου αποθήκευση του αρχείου μας.

Πάμε να δούμε την πρώτη μας αλλαγή στο αρχείο configuration.yaml με την οποία θα ορίσουμε έναν κωδικό πρόσβασης για το λειτουργικό μας.

Ανοίγοντας το αρχείο configuration.yaml μέσα στον Configurator και κατεβαίνουμε μέχρι να βρούμε το πεδίο http:

```
24 -
25 - http:
26   --# Secrets are defined in the file secrets.yaml
27   --api_password: lowercasetuppercasebandonezero
28   --# Uncomment this if you are using SSL/TLS, running in Docker container, etc.
29   --# base_url: example.duckdns.org:8123
30 -
```

Σβήνουμε την δίεση '#' και δίπλα στην εντολή api_password πληκτρολογούμε τον κωδικό της επιλογής μας. Για περισσότερο υλικό :

<https://www.home-assistant.io/addons/configurator>

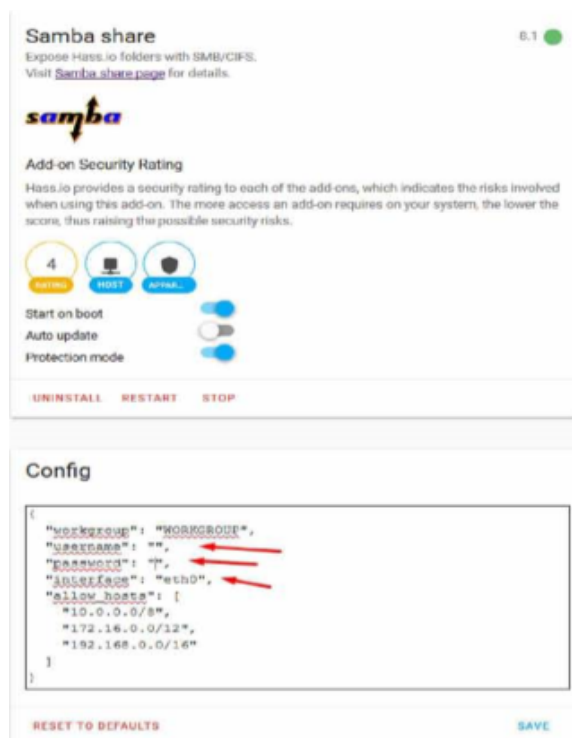
Αυτό ήταν το πιο βασικό πρόσθετο. Επόμενο είναι το Samba Share.

3.4.2) Εγκατάσταση προσθέτου Samba Share

Με αυτό το πρόσθετο δίνεται η δυνατότητα στο χρήστη να έχει πρόσβαση στους φακέλους και στα αρχεία του Hass.io χρησιμοποιώντας τις λειτουργίες δικτύου των Windows. Έτσι έχει την ευχέρεια να επεξεργάζεται τα αρχεία του από τον υπολογιστή του με λειτουργικό Windows. Ας δούμε πως θα το κάνουμε αυτό.

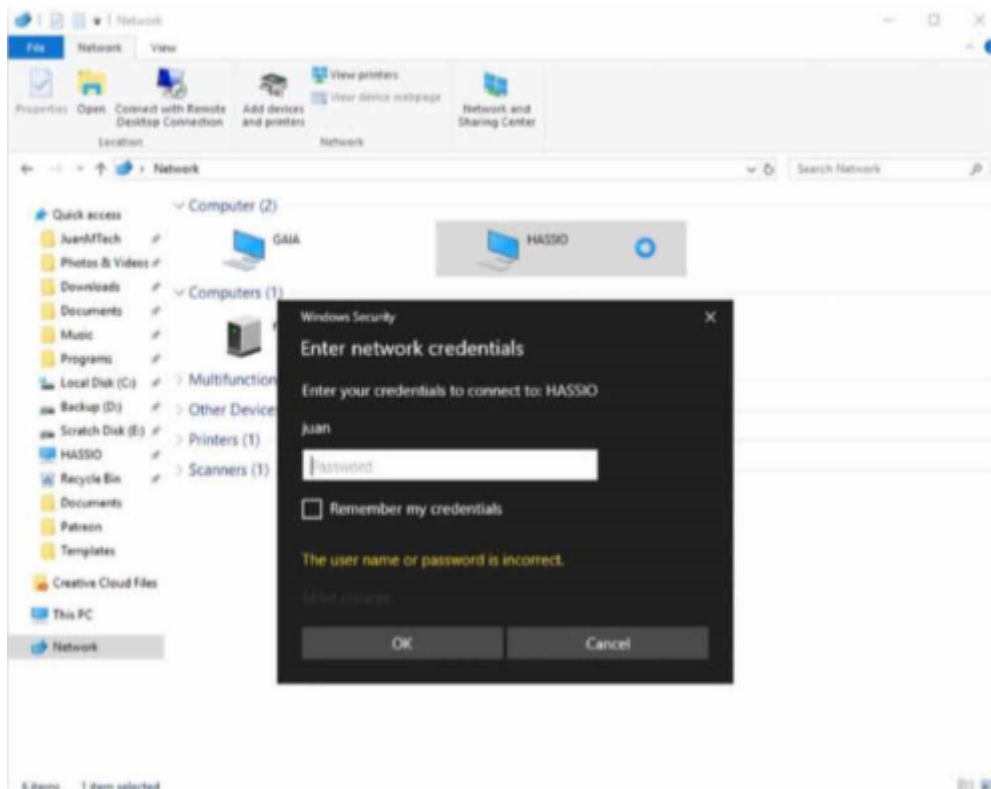
Αρχικά πρέπει να εγκαταστήσου με το πρόσθετο **Samba Share** με τον ίδιο τρόπο που εγκαταστήσαμε πιο πάνω το Configurator Add-on. Δηλαδή από τα αριστερά επιλεγούμε Hass.io -> ADD-ON STORE -> Samba Share -> Install.

Αφού τελειώσουμε με την εγκατάσταση κοιτάμε το configuration αυτού.

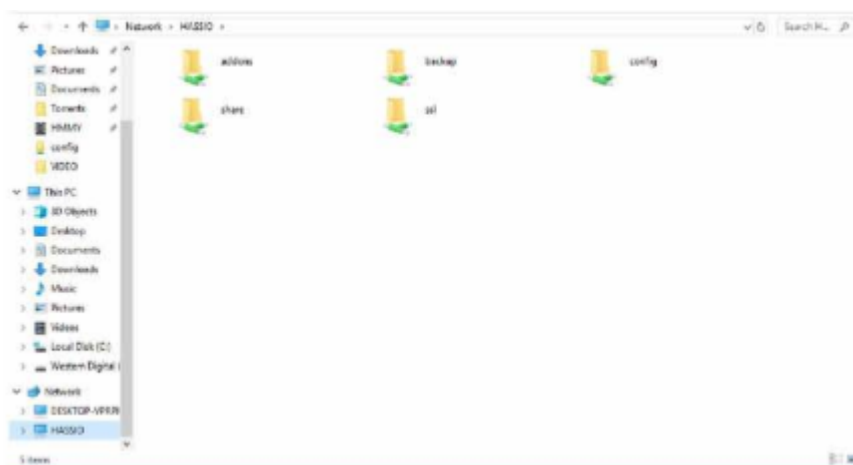


- Στα πεδία "username:" και "password:" ορίζουμε εμείς ένα όνομα και έναν κωδικό για να επιτύχουμε ασφάλεια. ΠΡΟΣΟΧΗ! Ότι πληκτρολογήσουμε πρέπει να βρίσκεται μέσα στα " ".
- Στο πεδίο "interface" έχουμε δύο επιλογές: eth0 ή wlan0 που εκφράζουν τον τρόπο σύνδεσης του RPI μας στο δίκτυο μας. Το πρώτο δηλώνει σύνδεση με καλώδιο τύπου ethernet, ενώ το δεύτερο μέσω WIFI.

Στη συνέχεια πατάμε SAVE και μετά START. Τώρα περνάμε στο windows pc μας. Ανοίγουμε τον "Εξερευνητή Αρχείων (File Explorer)", επιλέγουμε από τα αριστερά "Δίκτυο (Network)" και έπειτα HASSIO όπως θα βλέπουμε στην εικόνα.



Πλέον έχουμε πρόσβαση σε όλους τους φακέλους και αρχεία μας.



Απο εδώ και πέρα μπορούμε να επεξεργαζόμαστε τα αρχεία μας με έναν text editor όπως είναι οι Notepad++ και Atom!

<https://notepad-plus-plus.org/downloads/>

Περισσότερα για το πρόσθετο μας: <https://www.home-assistant.io/addons/samba/>

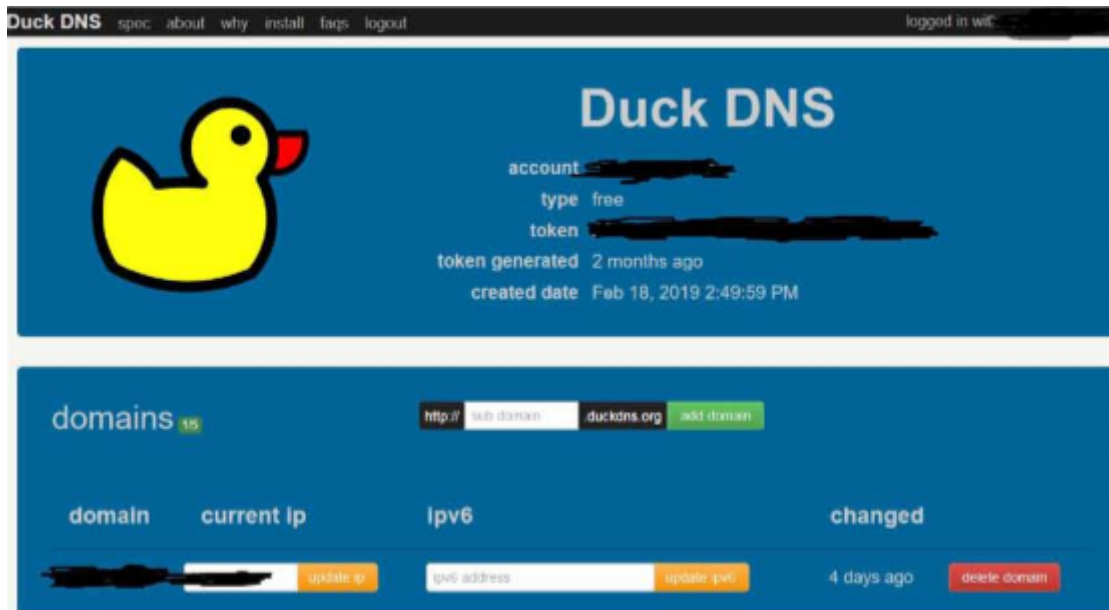
3.4.3) Εγκατάσταση Duck DNS

Τέλος θα δούμε αναλυτικά το τελευταίο και εξαιρετικής σημασίας πρόσθετο DuckDNS το οποίο μας προσφέρει ασφαλή απομακρυσμένο έλεγχο στο Home Assistant μας. Δηλαδή θα έχουμε πρόσβαση όταν είμαστε εκτός του τοπικού δικτύου του σπιτιού μας. Είναι μια δωρεάν υπηρεσία που θα αναθέσει ένα DNS (sub domains of duckdns.org) σε μια IP της επιλογής μας. Ενσωματώνει επίσης κρυπτογράφηση με την υπηρεσία Let's Encrypt. Ακολούθως θα δούμε πως θα επιτύχουμε αυτό μας το εγχείρημα.

Ξεκινάμε πάλι κάνοντας εγκατάσταση το πρόσθετο μας με τον τρόπο που δείξαμε στα δύο προηγούμενα. Hass.io -> ADD-ON STORE -> Duck DNS -> Install.

Σε αυτό το σημείο πρέπει να δημιουργήσουμε έναν δωρεάν λογαριασμό στη σελίδα του DuckDNS.

<https://www.duckdns.org/>



The screenshot shows the Duck DNS website interface. At the top, there is a navigation bar with links for 'spec', 'about', 'why', 'install', 'tags', and 'logout', and a 'logged in with' indicator. The main content area features a large yellow duck logo on the left and account information on the right, including 'account', 'type free', 'token', 'token generated 2 months ago', and 'created date Feb 18, 2019 2:49:59 PM'. Below this, there is a 'domains' section with a search bar and a '+5' indicator. A table lists domains with columns for 'domain', 'current ip', 'ipv6', and 'changed'. The table contains one entry with a redacted domain, an 'update ip' button, an 'update ipv6' button, and a '4 days ago' timestamp. A 'delete domain' button is also visible.

Όταν συνδεθούμε δημιουργείται για μας ένα μοναδικό token. Στη συνέχεια ορίζουμε εμείς ένα subdomain της επιλογής μας και πατάμε add domain. Ένα παράδειγμα είναι το smarthas.duckdns.org. Αφού αντιγράψουμε το token μας επιστρέφουμε στο configuration του DuckDNS του Home Assistant μας.

```
{
  "lets_encrypt": {
    "accept_terms": true,
    "certfile": "fullchain.pem",
    "keyfile": "privkey.pem"
  },
  "token": "sdfj-2131023-dslfjds-12321",
  "domains": ["my-domain.duckdns.org"],
  "seconds": 300
}
```

Στην επιλογή token κάνουμε επικόλληση το δικό μας token ενώ στο domain βάζουμε το subdomain που δημιουργήσαμε προγενέστερα. Και εδώ ότι γράφουμε πρέπει να περιλαμβάνεται μέσα στα " ".

Στη συνέχεια πάμε στο αρχείο configuration.yaml χρησιμοποιώντας κάποιο από τα προσθετά που εγκαταστήσαμε παραπάνω, δηλαδή Configurator ή Samba Share και κάνουμε τις εξής αλλαγές. Στο πεδίο http: οπου είχαμε ορίσει το api_password προσθέτουμε τα παρακάτω:

```
http:
  base_url: https://my-domain.duckdns.org:8123
  ssl_certificate: /ssl/fullchain.pem
  ssl_key: /ssl/privkey.pem
```

Και πάλι όπου το my-domain είναι το subdomain που δημιουργήσαμε.

Πλέον έχει μείνει ένα τελευταίο βήμα. Η ρύθμιση του Modem/Router μας για Port Forwarding. Σε αυτό το σημείο πρέπει να έχουμε ορίσει μια Static IP στο RPI μας. Αυτό γίνεται είτε με τον τρόπο που περιγράψαμε πιο πάνω, είτε μέσα από το router μας. Σ αυτό το σημείο πρέπει να πούμε ότι είναι δύσκολο να περιγράψουμε αναλυτικά τη διαδικασία για το λόγο ότι το κάθε router έχει το δικό του UI. Μπορούμε όμως να επισκεφτούμε την σελίδα <https://portforward.com> που περιέχει οδηγίες για το κάθε router. Ακολουθεί ένα παράδειγμα:



Service Name	Port Range	Local IP	Local Port	Protocol	Add / Delete
HassIo_SSL	443	192.168.X.XX	8123	TCP	⊕

***Το Home Assistant μας «ακούει» από προεπιλογή στην θύρα :8123. Όπου Local IP είναι η IP που έχουμε αναθέσει στο RPi μας.

Αφού κάναμε όλα τα παραπάνω θα μπορούμε να συνδεόμαστε στο Home Assistant μας από οπουδήποτε ανοίγοντας έναν φυλλομετρητή και πληκτρολογώντας

<https://my-subdomain-name.duckdns.org> ή τοπικά παραδείγματος χάρη στο <https://192.168.1.154:8123>.

3.4.4) Εγκατάσταση για device tracking του OWNTRACKS

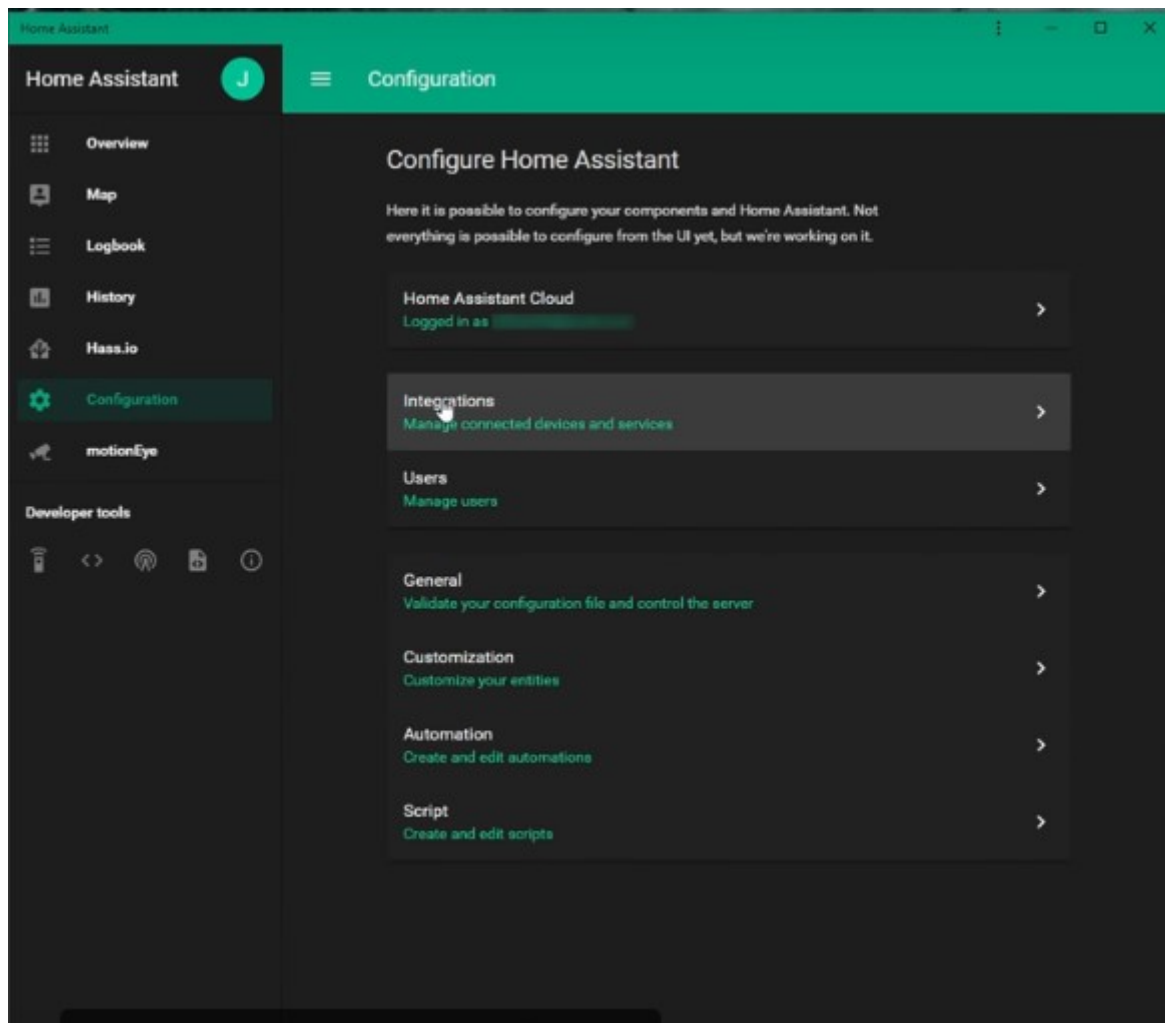
Ενα πολύ χρήσιμο ADD-ON είναι επίσης το OWNTRACKS. Η ανάγκη να αναγνωρίζουμε εάν κάποιος είναι στο σπίτι ή κατευθύνεται προς τα εκεί λύθηκε με την χρήση της εφαρμογής εντοπισμού του κινητού μας τηλεφώνου OWNTRACKS. Το συγκεκριμένο app μας βοηθάει να αναγνωρίζουμε που βρίσκεται το κινητο μας τηλέφωνο. Είναι το πιο σύνηθες INPUT για αυτοματισμούς διότι μας δίνεται η δυνατότητα να δημιουργούμε καταστάσεις όπως π.χ.

-Να μας ειδοποιεί αν το παιδί βρίσκεται στο σχολείο
-Να ανοίγει το air condition όταν φεύγουμε από την δουλειά.

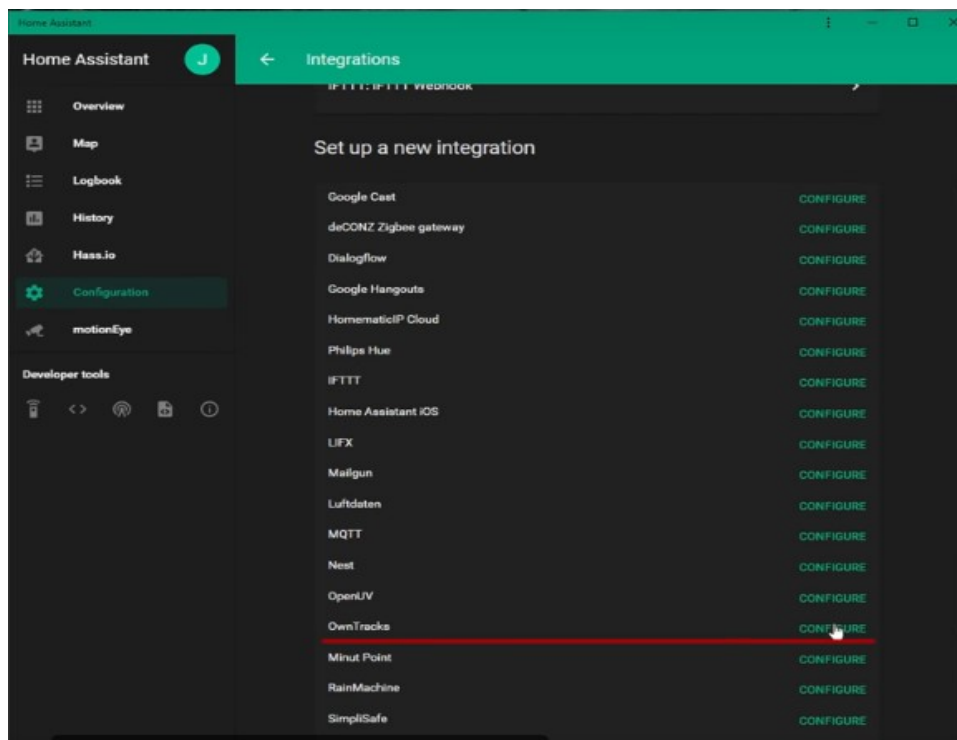
κ.ά.

Η διαδικασία είναι σχετικά απλή και περιγράφεται παρακάτω.

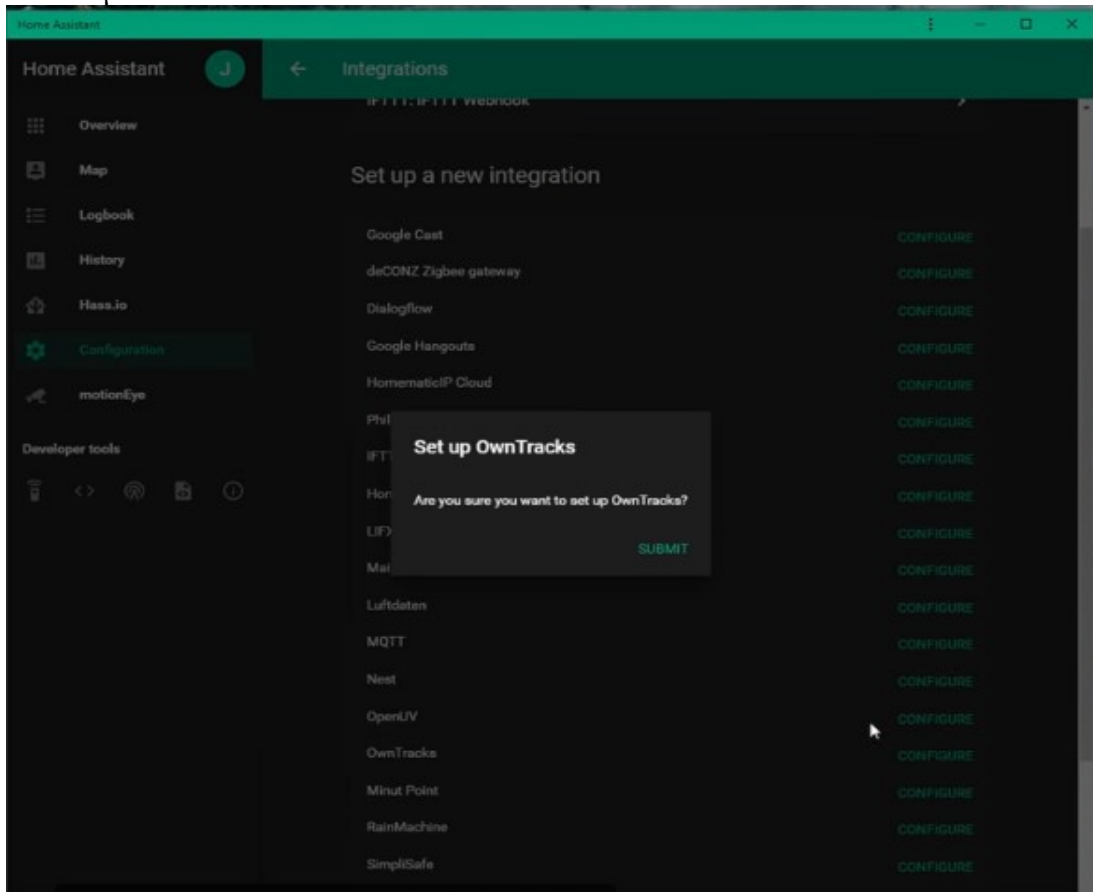
Αρχικά πηγαίνουμε στο menu του Home Assistant ,πατάμε Configuration και έπειτα Integrations.



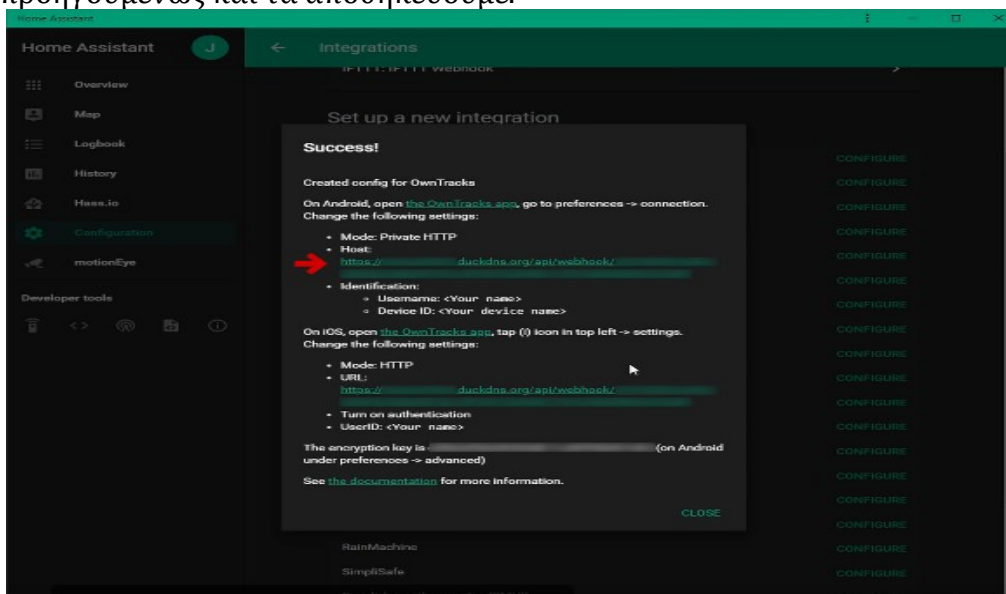
Κάνουμε Search του add-on OWNTRACKS και πατάμε Configure



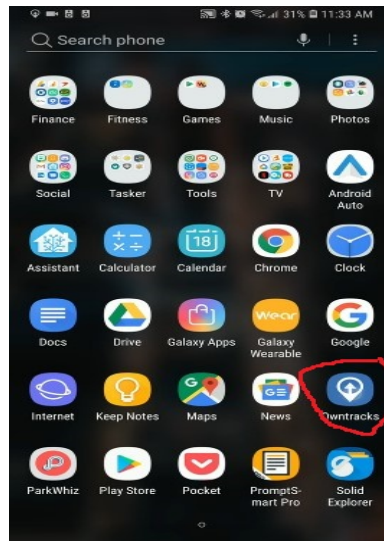
Έπειτα πατάμε submit



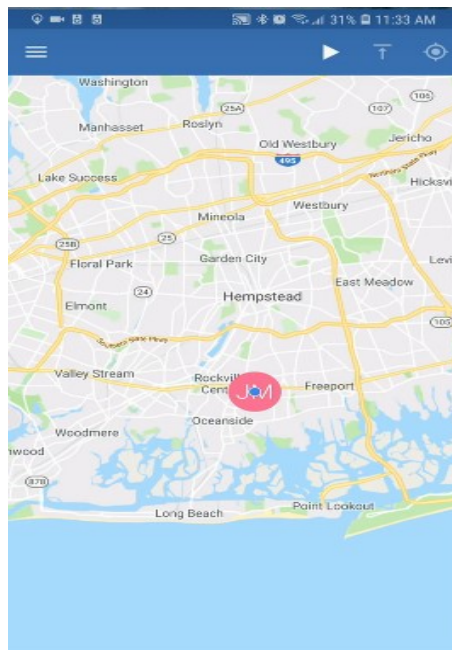
Στη συνέχεια μας δίνει το Host Name και το Encryption Key που δημιουργήσαμε στο DuckDns προηγουμένως και τα αποθηκεύουμε.



Το επόμενο βήμα είναι να εγκαταστήσουμε το application OWNTRACKS στο κινητό μας τηλέφωνο.



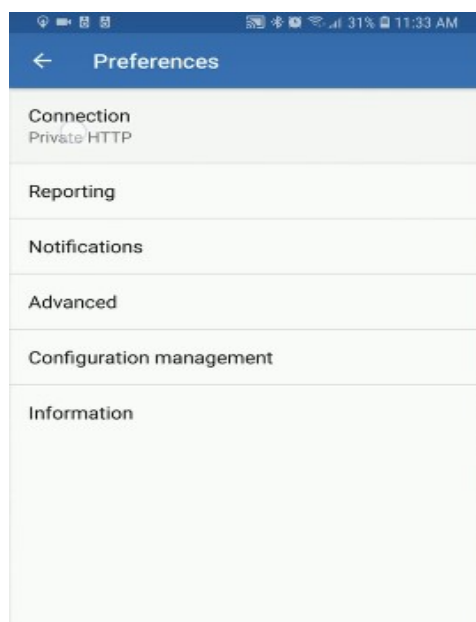
Και στη συνέχεια αφού μπούμε στην εφαρμογή κάνουμε click στο menu με τις 3 παύλες.



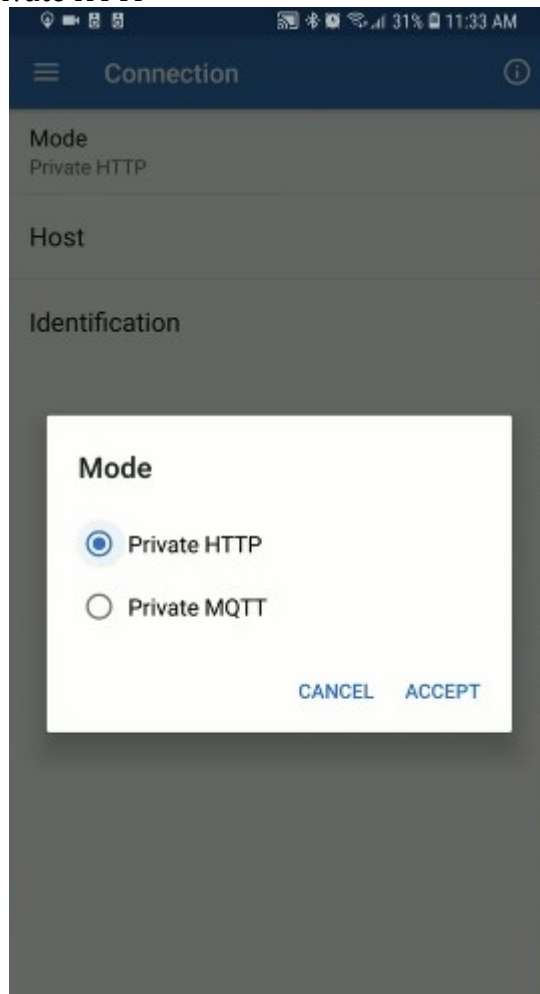
Πατάμε στο preferences



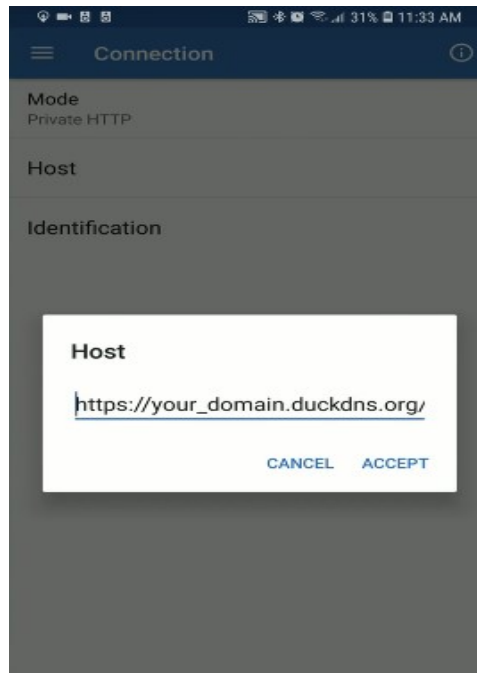
Και έπειτα πατάμε στην επιλογή Connection



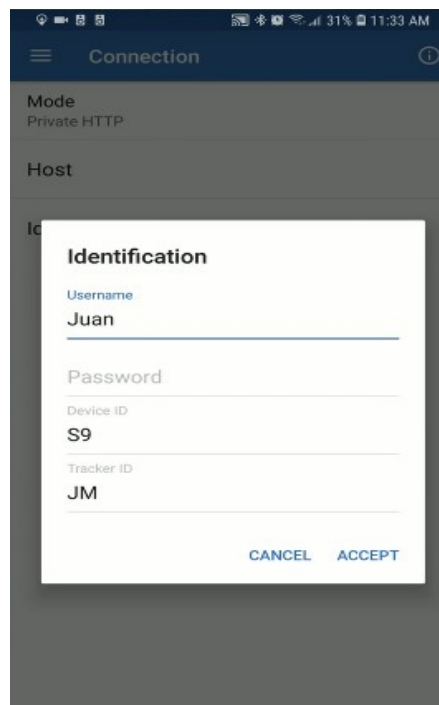
Σετάρουμε το Mode σε Private HTTP



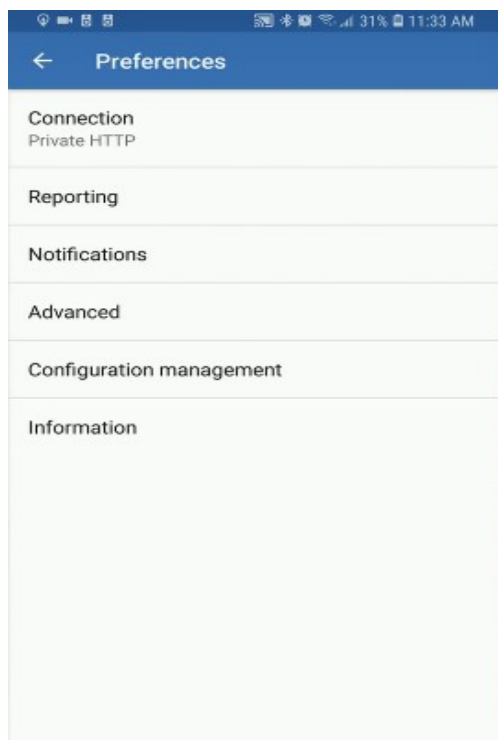
Και στο Host βάζουμε το Host Name που δημιουργήσαμε στο DuckDNS και που μας δώθηκε επίσης και απ'το AddOn Owntracks όταν το εγκαταστήσαμε στο Home Assistant προηγουμένως.



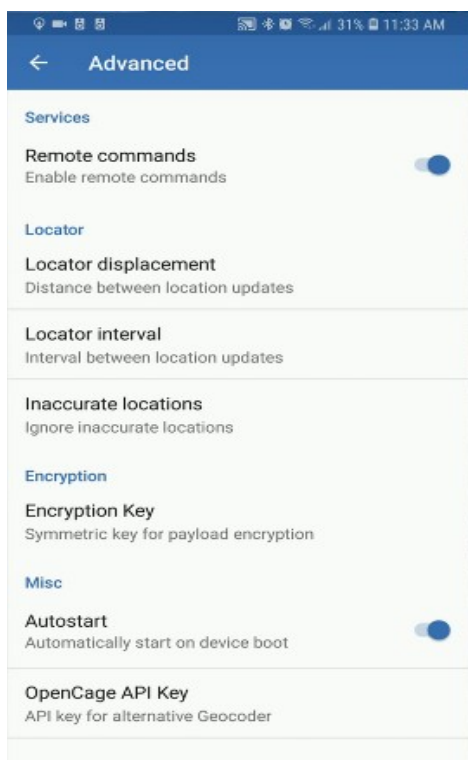
Στο identification βάζουμε ένα Username, ένα DeviceID και ένα TrackerID



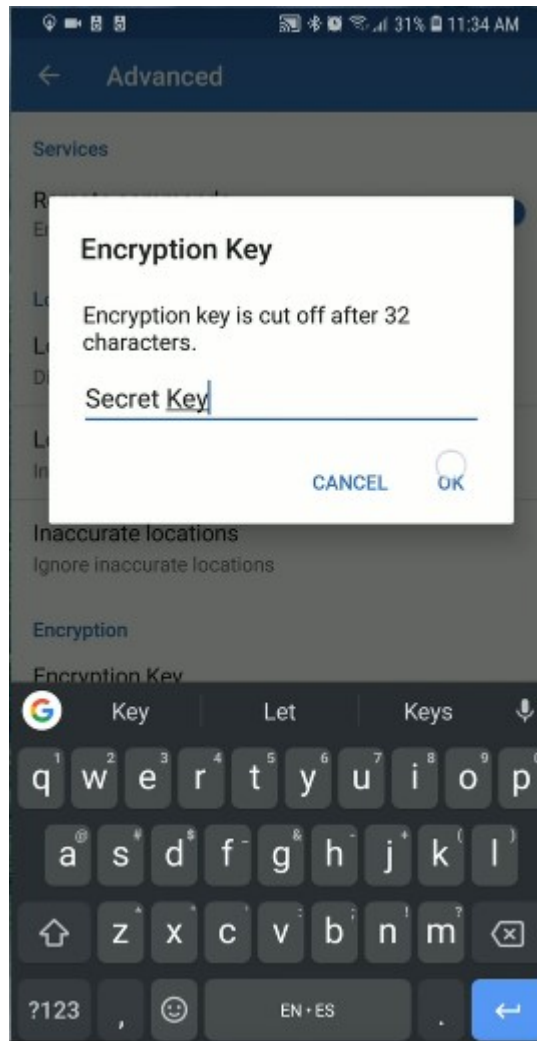
Έπειτα για να δηλώσουμε το key πηγαίνουμε πάλι στο menu και μετά στο preferences και επιλέγουμε το Advanced



Κάνουμε κλικ στο Encryption Key



Τέλος δηλώνουμε το key που δημιουργήσαμε στο DuckDns



3.5) Προσθήκη συσκευών στο Home Assistant

Πλέον είμαστε έτοιμοι να στήσουμε το έξυπνο σύστημα μας, ανεξάρτητα από εταιρικούς servers, και με την ασφάλεια να είναι από τα κύρια γνωρίσματα του συστήματος μας. Θα δείξουμε τρόπους για αναγνώριση και προσθήκη συσκευών με ZigBee και Z-Wave καθώς και τη δημιουργία ενός MQTT Broker. Οι συσκευές Wi-Fi αναγνωρίζονται αυτόματα όταν βρεθούν στο δίκτυο.

3.5.1) Προσθήκη αισθητήρων και συσκευών Zigbee

Το Raspberry Pi μας δεν έχει τη δυνατότητα να λειτουργήσει από μόνο του σαν ένας κόμβος για επικοινωνία Zigbee. Γι' αυτό το λόγο θα χρησιμοποιήσουμε μια εξωτερική συσκευή.

Conbee ή RaspBee by Dresden Elektronik

Είναι οι πιο εύκολες λύσεις και οι πιο φιλικές προς το χρήστη καθώς είναι έτοιμες λύσεις (out-of-the-box). Η μονή διαφορά αυτών των δύο είναι ο τρόπος σύνδεσης. Το Conbee είναι μια συσκευή USB που μπορεί να μπει εξωτερικά στο RPi είτε σε ένα υπολογιστή ενώ από την άλλη το RaspBee 'κουμπώνει' πάνω στην μονάδα GPIO του RPi. Και τα δυο καλύπτουν μια τεράστια ποικιλία από συσκευές και αυτό είναι σημαντικό γιατί όπως αναφέραμε παραπάνω η κάθε εταιρεία χρησιμοποιεί τη δική της ZigBee εκδοχή.

Από τη στιγμή που τα εγκαταστήσου με πάνω στο RPi μας πρέπει να κάνουμε εγκατάσταση το αντίστοιχο add-on τους για να συνεργαστούν με το HA μας. Το add-on λέγεται deCONZ και θα το βρούμε στο Hassio->ADD-ON STORE->deCONZ->Install. Μετά την εγκατάσταση δίνουμε την κατάλληλη τιμή στο config box (βάσει οδηγιών), πατάμε START και έπειτα OPEN WEB UI.

Αφού δημιουργήσουμε ένα Password θα βρεθούμε στην κεντρική σελίδα GUI (Graphical User Interface) από όπου μπορούμε πλέον να προσθέσουμε τις ZigBee συσκευές μας. Οι αισθητήρες αποτελούν πλέον entities και είναι έτοιμοι να μπουν σε σενάρια αυτοματισμών. Τέλος πρέπει να πούμε πως και οι δύο λύσεις προσφέρουν μελλοντική υποστήριξη πράγμα που τις καθιστά future proof λύσεις. Το Conbee/RaspBee μπορεί να βρεθεί εδώ:

<https://www.phoscon.de/en/conbee2>

<https://phoscon.de/en/raspbee>



ConBee usb stick



RasBee GPIO

3.5.2) Προσθήκη αισθητήρων και συσκευών Z-Wave

Και σε αυτήν την περίπτωση θα χρειαστούμε κάποια εξωτερική συσκευή που θα παίζει το ρόλο του κόμβου για τις Z-Wave συσκευές. Η δημοφιλέστερη υλοποίηση στην αγορά είναι αυτή της Aeotec με το Aeotec Z-Stick Series 5. ενώ κάποιες άλλες είναι οι εξής:

- ο Ever spring USB stick - Gen 5
- ο Sigma Designs UZB stick
- ο Vision USB stick – Gen 5
- ο Zooz Z-Wave Plus S2 stick ZST10
- ο ZWave.me Raspberry Board
- ο ZWave.me UZB1 stick

Στη συνέχεια μεταφερόμαστε στο αρχείο **configuration.yaml** και προσθέτουμε τα εξής:

```
# Example configuration.yaml entry
zwave:
  usb_path: /dev/ttyACM0
  device_config: !include zwave_device_config.yaml
```

Όπου το `usb_path:` είναι το path που έχει τοποθετηθεί η συσκευή. Για usb βάζουμε `/dev/ttyACM0` ενώ για GPIO `/dev/ttyAMA0`. Το `device_config:` πρέπει να μπει υπό ορισμένες προϋποθέσεις. Περισσότερες πληροφορίες και οδηγίες : <https://www.home-assistant.io/docs/z-wave/installation>

Για την πρόσθεση και αφαίρεση συσκευών Z-Wave:
<https://www.home-assistant.io/docs/z-wave/adding>

Για τον πίνακα ελέγχου Z-Wave:
<https://www.home-assistant.io/docs/z-wave/control-panel/>



Z stick Gen5 AEOTEC

3.5.3) Προσθήκη MQTT broker

Όπως είδαμε ένα από τα πιο σημαντικά πρωτοκολλά στο χώρο του IoT είναι το MQTT. Έτσι λοιπόν δεν θα μπορούσε να λείπει από ένα πλήρες σύστημα αυτοματισμών. Στο δικό μας σύστημα θα χρησιμοποιήσουμε ένα γνωστό broker με ονομασία Mosquitto Broker. Για την εγκατάσταση πάμε Hassio- >ADD-ON STORE->Mosquitto broker->Install. Επειτα πάμε στο Config Box που θα έχει αυτή τη μορφή.

```
{
  "logins": [
    {"username": "local-user", "password": "mypw"}
  ],
  "anonymous": false,
  "customize": {
    "active": false,
    "folder": "mosquitto"
  },
  "certfile": "fullchain.pem",
  "keyfile": "privkey.pem"
}
```

Αν θέλουμε να ορίσουμε ένα "username" & "password" το ορίζουμε μέσα στο "logins", πάντα μέσα στα " " αλλιώς τα σβήνουμε και αλλάζουμε το πεδίο "anonymous": true. Στη συνέχεια σώζουμε τις αλλαγές μας "SAVE" και ξεκινάμε το add-on μας "START". Ακολουθώντας από την δεξιά στήλη επιλέγουμε Configuration->Integrations->MQTT->CONFIGURE->click "Enable Discovery"->Submit. Πλέον όταν θα υπάρχει στο χώρο μας μια συσκευή που να υποστηρίζει MQTT το σύστημα μας θα την αναγνωρίζει αυτόματα.

Αυτά ήταν τα βασικά βήματα και ενέργειες που πρέπει να ακολουθήσει κάποιος χρήστης με σκοπό να δημιουργήσει το ιδιωτικό του τοπικό δίκτυο που θα του παρέχει ασφάλεια, σταθερότητα και ιδιωτικότητα. Πλέον όλα αφήνονται στη φαντασία του εκάστοτε χρήστη για το τι σενάρια αυτοματισμών θέλει να δημιουργήσει.

Κεφάλαιο 4

4) ΠΑΡΑΔΕΙΓΜΑΤΑ ΑΥΤΟΜΑΤΙΣΜΩΝ

- 4.1) # Example of entry in configuration.yaml

**# Turns on lights 1 hour before sunset if people are home
and if people get home between 16:00-23:00**

automation:

- **alias: 'Rule 1 Light on in the evening'**

trigger:

- # Prefix the first line of each trigger configuration

- # with a '-' to enter multiple

- platform: sun

- event: sunset

- offset: '-01:00:00'

- platform: state

- entity_id: group.all_devices

- to: 'home'

condition:

- # Prefix the first line of each condition configuration

- # with a '-' to enter multiple

- condition: state

- entity_id: group.all_devices

- state: 'home'

- condition: time

- after: '16:00:00'

- before: '23:00:00'

action:

- # With a single service call, we don't need a '-' before service - though you can if you want to

- service: homeassistant.turn_on

- entity_id: group.living_room

Turn off lights when everybody leaves the house

- alias: 'Rule 2 - Away Mode'

```
trigger:
  platform: state
  entity_id: group.all_devices
  to: 'not_home'
action:
  service: light.turn_off
  entity_id: group.all_lights
```

Notify when Paulus leaves the house in the evening

- alias: 'Leave Home notification'

```
trigger:
  platform: zone
  event: leave
  zone: zone.home
  entity_id: device_tracker.paulus
condition:
  condition: time
  after: '20:00'
action:
  service: notify.notify
  data:
    message: 'Paulus left the house'
```

Send a notification via Pushover with the event of a Xiaomi cube. Custom event from the Xiaomi component.

- alias: 'Xiaomi Cube Action'

```
hide_entity: false
initial_state: false
trigger:
  platform: event
  event_type: cube_action
  event_data:
    entity_id: binary_sensor.cube_158d000103a3de
action:
  service_template: notify.pushover
  data_template:
    title: "Cube event detected"
    message: "Cube has triggered this event: {{ trigger.event }}"
```

• 4.2)# Example for rainy days

automation:

- **alias: 'Rainy Day'**

trigger:

- platform: state
- entity_id: sensor.precip_intensity
- to: 'rain'

condition:

- condition: state
- entity_id: group.all_devices
- state: 'home'
- condition: time
- after: '14:00'
- before: '23:00'

action:

- service: light.turn_on
- entity_id: light.couch_lamp

- **alias: 'Rain is over'**

trigger:

- platform: state
- entity_id: sensor.precip_intensity
- to: 'None'

condition:

- condition: sun
- after: 'sunset'
- after_offset: '-01:00:00'

action:

- service: light.turn_off
- entity_id: light.couch_lamp

• 4.3)# Examples using the sun

#Turn on the living room lights 45 minutes before sunset if anyone is at home

automation:

trigger:

- platform: sun
- event: sunset

```
offset: "-00:45:00"
condition:
  condition: state
  entity_id: group.all_devices
  state: home
action:
  service: homeassistant.turn_on
  entity_id: group.living_room_lights
```

- **4.4)#Send sun rise/sun set notifications using pushbullet**

```
automation:
- alias: 'Send notification when sun rises'
  trigger:
    platform: sun
    event: sunrise
    offset: '+00:00:00'
  action:
    service: notify.pushbullet
    data:
      message: 'The sun is up.'
- alias: 'Send notification when sun sets'
  trigger:
    platform: sun
    event: sunset
    offset: '+00:00:00'
  action:
    service: notify.pushbullet
    data:
      message: 'The sun is down.'
```

- **4.5)#Automations for lights and blinds based on solar elevation**

- **alias: 'Turn a few lights on when the sun gets dim'**

trigger:

platform: numeric_state

entity_id: sun.sun

value_template: "{{ state_attr('sun.sun', 'elevation') }}"

below: 3.5

action:

service: scene.turn_on

entity_id: scene.background_lights

- **alias: 'Turn more lights on as the sun gets dimmer'**

trigger:

platform: numeric_state

entity_id: sun.sun

value_template: "{{ state_attr('sun.sun', 'elevation') }}"

below: 1.5

action:

service: scene.turn_on

entity_id: scene.more_lights

- **alias: 'Close blind at dusk'**

trigger:

platform: numeric_state

entity_id: sun.sun

value_template: "{{ state_attr('sun.sun', 'elevation') }}"

below: -2.5

action:

service: switch.turn_off

entity_id: switch.blind

- **4.6)#Turn on lights with a resetable off timer**

automation:

- **alias: Turn on kitchen light when there is movement**

trigger:

platform: state

entity_id: sensor.motion_sensor

to: 'on'

action:

service: homeassistant.turn_on

entity_id: light.kitchen_light

- alias: Turn off kitchen light 10 minutes after last movement

trigger:

platform: state

entity_id: sensor.motion_sensor

to: 'off'

for:

minutes: 10

action:

service: homeassistant.turn_off

entity_id: light.kitchen_light

#Or in the case of multiple sensors/triggers:

automation:

- alias: Turn on hallway lights when the doorbell rings, the front door opens or if there is movement

trigger:

- platform: state

entity_id: sensor.motion_sensor, binary_sensor.front_door, binary_sensor.doorbell

to: 'on'

action:

- service: homeassistant.turn_on

data:

entity_id:

- light.hallway_0

- light.hallway_1

- service: timer.start

data:

entity_id: timer.hallway

- alias: Turn off hallway lights 10 minutes after trigger

trigger:

platform: event

```
event_type: timer.finished
event_data:
  entity_id: timer.hallway
action:
  service: homeassistant.turn_off
  data:
    entity_id:
      - light.hallway_0
      - light.hallway_1
```

```
timer:
  hallway:
    duration: '00:10:00'
```

**#You can also restrict lights from turning on based on time of day and
#implement transitions for fading lights on and off.**

- alias: Motion Sensor Lights On

```
trigger:
  platform: state
  entity_id: binary_sensor.ecolink_pir_motion_sensor_sensor
  to: 'on'
condition:
  condition: time
  after: '07:30'
  before: '23:30'
action:
  service: homeassistant.turn_on
  entity_id: group.office_lights
  data:
    transition: 15
```

- alias: Motion Sensor Lights Off

```
trigger:
  - platform: state
    entity_id: binary_sensor.ecolink_pir_motion_sensor_sensor
    to: 'off'
  for:
    minutes: 15
```

action:

- service: homeassistant.turn_off

entity_id: group.office_lights

data:

transition: 160

Κεφάλαιο 5

5)Επίλογος

Σκοπός αυτής της διπλωματικής εργασίας ήταν να δώσει στον αναγνώστη μια καλή εικόνα για το τι είναι Internet of Things, Smart Home και Home Automation. Παρουσιάστηκαν πρωτόκολλα επικοινωνίας για διασύνδεση και επικοινωνία μεταξύ 'έξυπνων' συσκευών καθώς και κάποια είδη των συσκευών αυτών και των αυτοματισμών που προσφέρουν. Γνωστοποιήθηκαν συνοπτικά ολοκληρωμένες λύσεις ανοιχτού κώδικα.Κλείνοντας, συντάχθηκε ένας αναλυτικός οδηγός για την εγκατάσταση και ανάπτυξη του λειτουργικού Home Assistant σε ένα Raspberry Pi, με σκοπό να δοθεί το έναυσμα για τη δημιουργία ενός αυτόνομου, ασφαλούς και τοπικού κόμβου που θα αυτοματοποιήσει την καθημερινότητα ενός χρήστη.

Παρά το γεγονός ότι οι έξυπνες οικιακές συσκευές έχουν βελτιωθεί και πολλαπλασιαστεί όπως και οι αυτοματισμοί μεταξύ αυτών, δεν παύουν όμως να υπάρχουν περιθώρια ανάπτυξης και βελτίωσης. Όπως είδαμε δύσκολα μπορεί κάποιος να βρει ολοκληρωμένες λύσεις από μια εταιρεία πράγμα που καθιστά την διασυνδεσιμότητα μεταξύ συσκευών δύσκολη και πολλές φορές δαπανηρή. Έτσι δημιουργείται η ανάγκη για ανάπτυξη ενός ενιαίου τρόπου επικοινωνίας μεταξύ έξυπνων συσκευών. Επίσης, το κόστος των συσκευών και των υπηρεσιών μπορεί να υποστεί μείωση μέσα στα επόμενα χρόνια και έτσι η ιδέα ενός 'έξυπνου' οικιακού οικοσυστήματος να γίνει ακόμα πιο προσιτή. Οι λύσεις ανοιχτού κώδικα συνεχώς βελτιώνονται με βάση το feedback των χρηστών, που συνεχώς αυξάνονται, και γίνονται πιο φιλικές προς αυτούς. Επιπλέον, μπορεί να δοθεί μεγαλύτερη βάση στην εξοικονόμηση ενέργειας και την προστασία του περιβάλλοντος μιας και ο χρήστης θα έχει επίγνωση της ενεργειακής κατανάλωσης κάθε συσκευής. Επιπροσθέτως, με την εξέλιξη των ψηφιακών βοηθειών και της Τεχνίτης Νοημοσύνης θα δίνεται ακόμα μεγαλύτερο εύρος στους τρόπους που ένας χρήστης θα μπορεί να αυτοματοποιήσει και να προστατέψει το σπίτι του. Τέλος, αυτό που αποτελεί την μεγαλύτερη πρόκληση στο θέμα μας είναι η ασφάλεια. Πολλά πρωτοκολλά επικοινωνίας όπως και έξυπνες συσκευές είναι ευάλωτα σε κυβερνοεπιθέσεις. Μεγάλος όγκος προσωπικών δεδομένων που χρησιμοποιούν οι συσκευές και οι servers αυτών είναι δελεαστικός στόχος για κακόβουλα άτομα στο χώρο του διαδικτύου. Οι εταιρείες θα πρέπει να επενδύσουν σε αυτόν τον τομέα με σκοπό την απολυτή ασφάλεια των δεδομένων όλων των χρηστών. Παράλληλα με όλα αυτά, θα δημιουργηθούν νέες ειδικότητες και θα ανοίξουν θέσεις εργασίες σε όλο τον κόσμο. Εν κατακλείδι, αν για όλα αυτά βρεθούν λύσεις και επιτευχθεί η ανάλογη ανάπτυξη τότε το IoT θα αποτελεί αναπόσπαστο κομμάτι κάθε κατοικίας και θα προσδίδει μια πιο εύκολη και ποιοτική ζωή.

Βιβλιογραφία

- [1] Website of Home Assistant «www.home-assistant.io».
- [2] «Wikipedia» Network Protocol.
- [3] «Wikipedia» Raspberry Pi.
- [4] D. Hendricks, «IoT Evolution World».
- [5] N. Sakovich, «Sam Solutions» Internet of Things (IoT) Protocols and Connectivity Options: An Overview.
- [6] M. B. Yassein, W. Mardini και K. Ashwaq, «Smart homes automation using Z-wave protocol».
- [7] C. R. Muthu, R. Prabakaran και M. Shanmugaraj, «Study on ZigBee technology» International Conference on Electronics Computer Technology.
- [8] J. Baker, «Opensource».
- [9] R. Santos, «Random Nerd Tutorials».
- [10] F. T. Council, «Forbes» 14 Predictions For The Future Of Smart Home Technology.
- [11] «Statista» Internet of Things - Statistics & Facts.