



ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ

Σχολή Τεχνολογίας

Τμήμα Ψηφιακών Συστημάτων

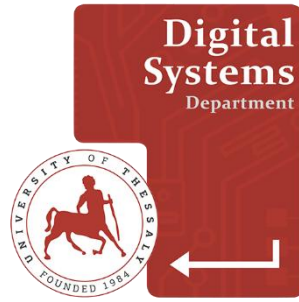
**Σχεδιασμός και Υλοποίηση μιας
Έξυπνης Αίθουσας Διδασκαλίας
με το Home Assistant**

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Κωνσταντίνος Καραμήτσος (ΑΜ: 3120118)

Επιβλέπων: Ονούφριος Χαραλάμπους, Αναπληρωτής Καθηγητής

ΛΑΡΙΣΑ, Οκτώβριος 2025



UNIVERSITY OF THESSALY

School of Technology

Digital Systems Department

Design and Setup of a Smart Classroom with Home Assistant

BACHELOR THESIS

Konstantinos Karamitsos (RN: 3120118)

Supervisor: *Onoufrios Haralampous, Associate Professor*

LARISSA, October 2025

Υπεύθυνη Δήλωση περί Ακαδημαϊκής Δεοντολογίας και Πνευματικών Δικαιωμάτων

Με πλήρη επίγνωση των συνεπειών του νόμου περί πνευματικών δικαιωμάτων, δηλώνω ρητά ότι η παρούσα πτυχιακή εργασία, καθώς και τα ηλεκτρονικά αρχεία και πηγαίοι κώδικες που αναπτύχθηκαν ή τροποποιήθηκαν στα πλαίσια αυτής της εργασίας, αποτελεί αποκλειστικά προϊόν προσωπικής μου εργασίας, δεν προσβάλλει κάθε μορφής δικαιώματα διανοητικής ιδιοκτησίας, προσωπικότητας και προσωπικών δεδομένων τρίτων, δεν περιέχει έργα/εισφορές τρίτων για τα οποία απαιτείται άδεια των δημιουργών/δικαιούχων και δεν είναι προϊόν μερικής ή ολικής αντιγραφής, οι πηγές δε που χρησιμοποιήθηκαν περιορίζονται στις βιβλιογραφικές αναφορές και μόνον, και πληρούν τους κανόνες της επιστημονικής παράθεσης. 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.

Αναλαμβάνω πλήρως, ατομικά και προσωπικά, όλες τις νομικές και διοικητικές συνέπειες που δύναται να προκύψουν στην περίπτωση κατά την οποία αποδειχθεί, διαχρονικά, ότι η εργασία αυτή ή τμήμα της δεν μου ανήκει διότι είναι προϊόν λογοκλοπής.

Ο/Η Δηλών/ούσα

(Υπογραφή)

Καραμήτσος Κωνσταντίνος

Ημερομηνία

08/10/2025

Εγκρίνεται από την Επιτροπή Εξέτασης:

Επιβλέπων/πουσα	Ονούφριος Χαραλάμπους Αναπληρωτής Καθηγητής, Τμήμα Ψηφιακών Συστημάτων, Πανεπιστήμιο Θεσσαλίας
Μέλος	Νικόλαος Σαμαράς Καθηγητής, Τμήμα Ψηφιακών Συστημάτων, Πανεπιστήμιο Θεσσαλίας
Μέλος	Κωνσταντίνος Κόκκινος Επίκουρος Καθηγητής, Τμήμα Ψηφιακών Συστημάτων, Πανεπιστήμιο Θεσσαλίας

Ημερομηνία έγκρισης: 10-10-2025

Περίληψη

Η τεχνολογία, στον γοργά εξελισσόμενο κόσμο, έχει γίνει πλέον ένα αναπόσπαστο κομμάτι της καθημερινότητάς μας· μεταβάλλοντας με τους ίδιους ακριβώς ρυθμούς τους χώρους και τους τρόπους με τους οποίους ζούμε, εργαζόμαστε και εκπαιδευόμαστε. Για να φτάσει όμως να μας επηρεάζει σε τέτοιο βαθμό, η τεχνολογία, δεν αρκεί να είναι απλά εντυπωσιακή και πρωτοπόρα, αλλά πρέπει να είναι και χρήσιμη! Η χρηστικότητα της προκύπτει μέσα από τη δυνατότητα να συγκεντρώνει δεδομένα, να προβαίνει σε υπολογισμούς (με βάση αυτά τα δεδομένα) και, τέλος, να ενεργοποιεί τους κατάλληλους αυτοματισμούς. Ένας τρόπος να υλοποιηθεί αυτή η αλληλουχία, είναι με τη χρήση λογισμικών όπως το Home Assistant.

Το HomeAssistant OS (HAOS), είναι ένα open-source λογισμικό, σχεδιασμένο να διαχειρίζεται και να εκτελεί εντολές με βάση τις τιμές και τα αποτελέσματα διάφορων αισθητήρων και ελεγκτών, υλοποιώντας έτσι τους λεγόμενους αυτοματισμούς. Λειτουργώντας απρόσκοπτα και χωρίς το οικοσύστημα και τα πρωτόκολλα της κάθε συσκευής και κατασκευαστή αυτής, να αποτελούν εμπόδιο στη σωστή λειτουργία του (αν και υπάρχουν κάποιες εξαιρέσεις).

Σε αυτή την πτυχιακή εργασία – και χωρίς το “Home” στο όνομα του να μας εμποδίζει– θα εξερευνήσουμε το HAOS και τις δυνατότητες που προσφέρει. Στόχος μας είναι να χρησιμοποιήσουμε έξυπνες ρουτίνες και αυτοματισμούς, ώστε να μετατρέψουμε έναν ήδη μοντέρνο χώρο διδασκαλίας, του Τμήματος Ψηφιακών Συστημάτων του Πανεπιστημίου Θεσσαλίας, σε έναν πιο έξυπνο, λειτουργικό και ενεργειακά αποδοτικό χώρο· δίνοντας έτσι έμφαση στη μείωση του περιβαλλοντικού του αποτυπώματος, στη βελτίωση της εμπειρίας των φοιτητών και του προσωπικού και στη δημιουργία ενός προτύπου για το πώς μια τέτοια δομή αυτοματισμών μπορεί να εφαρμοστεί και σε άλλα εκπαιδευτικά ιδρύματα.

Abstract

Technology, in the fast-paced world, has become an integral part of our daily lives, transforming at the same rapid pace, the spaces and the ways in which we live, work and learn. However, for technology to influence us to such an extent, it is not enough for it to be impressive and pioneering. It must also be useful. Its usefulness arise from its ability to collect data, perform calculations (based on the data) and ultimately activate the appropriate automations. One way to implement this sequence is through software platforms and operating systems such as Home Assistant.

Home Assistant OS (or HAOS) is an open-source OS designed to manage and execute commands based on the values and results of various sensors and smart relays, thereby implementing what are we know as automations. It operates seamlessly, without the eco-systems and protocols of each device and manufacturer posing a barrier to its proper functioning (although there are some exceptions).

In this thesis -and without the word “Home” in its name limit us- we will explore HAOS and the capabilities it offers. Our goal is to utilize smart routines and automations to transform an already modern classroom, of the Department of Digital Systems at the University of Thessaly, into a smarter, more functional and energy efficient space. In doing so, we aim to reduce its environmental footprint, improve the experience of students and staff and create a model for how such an automation structure can be applied to other educational institutions.

Ευχαριστίες

Αρχικά, θα ήθελα να εκφράσω την ειλικρινή ευγνωμοσύνη μου προς τον επιβλέποντα καθηγητή μου, τον κ. Ονούφριο Χαραλάμπους, για την αδιάληπτη καθοδήγηση, την επιστημονική υποστήριξη και την εμπιστοσύνη που μου παρείχε καθ' όλη τη διάρκεια της εκπόνησης, ενώ θα ήθελα να ευχαριστήσω θερμά και τον κ. Τηλέμαχο Στυλιανό, για την έγκριση και παραχώρηση πρόσβασης στον απαραίτητο εξοπλισμό. Ιδιαίτερες ευχαριστίες απευθύνω στην φοιτητική ομάδα τεχνικής υποστήριξης του Τμήματος και ειδικότερα στους (αλφαβητικά) Άγγελος Αγγελόπουλος και Ευάγγελος Δερβεντζής, για την πρόθυμη συνεργασία, την τεχνογνωσία και την πρακτική βοήθεια που μου πρόσφεραν σε κρίσιμα στάδια της ανάπτυξης της υλοποίησης του Home Assistant. Ακόμα, θα ήθελα να ευχαριστήσω το ίδιο το Τμήμα και μαζί και τον πρόεδρό του, τον κ. Βασίλειο Γερογιάννη, για τις μοντέρνες υποδομές και τη διαθεσιμότητα των πόρων, βασικός πυλώνας για την πρακτική εφαρμογή της εργασίας. Τέλος θα ήθελα να ευχαριστήσω την οικογένειά μου και πιο πολύ τη μητέρα μου, που με κάθε τρόπο, άμεσο ή έμμεσο, έμπρακτο ή ηθικό, μου επέτρεψε να είμαι μέλος αυτής της, τόσο όμορφης, πανεπιστημιακής κοινότητας, εδώ στη Λάρισα, παρά το γεγονός ότι είμαστε κάτοικοι Πειραιά.

Κωνσταντίνος Καραμήτσος

08/10/2025

Περιεχόμενα

ΠΕΡΙΛΗΨΗ	VII
ABSTRACT	IX
ΕΥΧΑΡΙΣΤΙΕΣ.....	XI
ΠΕΡΙΕΧΟΜΕΝΑ.....	XIII
1 ΕΙΣΑΓΩΓΗ	15
1.1 ΈΞΥΠΝΑ ΚΤΗΡΙΑ	15
1.1.1 Εξοικονόμηση ενέργειας και φυσικών πόρων.....	15
1.1.2 Έλεγχος περιβαλλοντικών συνθηκών	16
1.1.3 Συστήματα ασφαλείας σε κτήρια	16
1.2 ΛΥΣΕΙΣ BMS	16
1.3 HOME ASSISTANT	18
1.4 ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ ΚΑΙ ΟΦΕΛΗ	18
1.5 ΜΕΙΟΝΕΚΤΗΜΑΤΑ	20
1.6 ΟΙ ΣΤΟΧΟΙ ΤΗΣ ΕΡΓΑΣΙΑΣ	20
1.7 ΑΝΑΜΕΝΟΜΕΝΑ ΑΠΟΤΕΛΕΣΜΑΤΑ	21
2 ΣΧΕΔΙΑΣΜΟΣ ΈΞΥΠΝΗΣ ΑΙΘΟΥΣΑΣ ΔΙΔΑΣΚΑΛΙΑΣ	23
2.1 ΧΩΡΟΣ ΥΛΟΠΟΙΗΣΗΣ.....	23
2.2 ΕΓΚΑΤΑΣΤΑΣΗ HOME ASSISTANT.....	25
2.3 NETWORK SETUP	26
2.4 ΡΥΘΜΙΣΗ HOME ASSISTANT ΚΑΙ ΕΡΓΑΛΕΙΑ ΑΝΑΠΤΥΞΗΣ (ADD-ONS)	27
2.4.1 Studio Code Server.....	27
2.4.2 Advanced SSH & Web Terminal.....	28
2.5 ΑΙΣΘΗΤΗΡΕΣ ΚΑΙ ΣΥΣΚΕΥΕΣ ΕΞΥΠΝΟΥ ΕΛΕΓΧΟΥ.....	29
2.5.1 Shelly Motion 2	30
2.5.2 Μετεωρολογικός σταθμός GreenDigital	31
2.5.3 Shelly Plus 1PM & 2PM	32
3 ΕΓΚΑΤΑΣΤΑΣΗ ΕΞΟΠΛΙΣΜΟΥ ΚΑΙ ΡΥΘΜΙΣΗ	34

3.1	ΕΓΚΑΤΑΣΤΑΣΗ SHELLY MOTION 2	34
3.2	ΕΓΚΑΤΑΣΤΑΣΗ SHELLY PLUS 1PM & 2PM.....	34
3.2.1	Εγκατάσταση ανατολικού σκιάστρου	35
3.2.2	Εγκατάσταση δυτικών σκιάστρων	35
3.2.3	Εγκατάσταση σε φωτισμό	37
3.3	ΣΥΝΔΕΣΗ SHELLY ΣΥΣΚΕΥΩΝ ΣΤΟ WiFi ΚΑΙ ΒΑΣΙΚΗ ΡΥΘΜΙΣΗ	38
3.3.1	Shelly Motion 2.....	39
3.3.2	Shelly Plus 2PM.....	39
3.3.3	Shelly Plus 1PM.....	41
3.4	ΕΝΣΩΜΑΤΩΣΗ ΣΤΟ HOME ASSISTANT	43
3.4.1	Shelly Motion 2.....	43
3.4.2	Shelly Plus 1PM & 2PM.....	44
3.4.3	Μετεωρολογικός σταθμός GreenDigital	45
4	ΑΥΤΟΜΑΤΙΣΜΟΙ ΚΑΙ ΡΟΥΤΙΝΕΣ	51
4.1	ΣΚΙΑΣΤΡΑ	51
4.1.1	Αυτοματισμός εκτίμησης θέσης.....	51
4.1.2	Ρουτίνα/Script ευθυγράμμισης σκιάστρων	54
4.1.3	Κύριος αυτοματισμός και ανάλυση των συνθηκών του	56
4.2	ΣΥΜΠΥΚΝΩΣΗ ΣΤΑ ΤΖΑΜΙΑ	59
5	ΔΙΕΠΑΦΗ ΧΡΗΣΤΗ	63
5.1	LAB4 – ΕΙΚΟΝΙΚΗ ΚΑΤΟΨΗ ΚΑΙ ΈΛΕΓΧΟΣ ΣΥΣΚΕΥΩΝ.....	64
5.2	ENERGY DETAILS - ΠΑΡΑΚΟΛΟΥΘΗΣΗ ΕΝΕΡΓΕΙΑΣ ΚΑΙ ΚΟΣΤΟΥΣ	67
5.3	ΓΑΙΟΡΟΛΙΣ – ΔΕΔΟΜΕΝΑ ΚΑΙ ΜΕΤΡΗΣΕΙΣ ΑΠΟ ΤΟΝ ΠΑΝΕΠΙΣΤΗΜΙΑΚΟ ΧΩΡΟ.	69
5.4	GENERAL DS INFORMATION – ΓΕΝΙΚΟΤΕΡΕΣ ΠΛΗΡΟΦΟΡΙΕΣ ΣΧΕΤΙΚΑ ΜΕ ΤΟ ΤΜΗΜΑ ΨΗΦΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ	72
6	ΣΥΜΠΕΡΑΣΜΑΤΑ	75
	ΒΙΒΛΙΟΓΡΑΦΙΑ.....	77
	ΠΑΡΑΡΤΗΜΑ Α	81

1 Εισαγωγή

Παρά τις προσπάθειες του ελληνικού κράτους και της ΕΕ, για αντικατάσταση παλιών συσκευών σε νοικοκυριά και επαγγελματικές εγκαταστάσεις, μέσω επιδοτήσεων [1], προσπαθώντας να μειώσουν το ενεργειακό κόστος της καθημερινής μας ζωής, εμείς συνεχίζουμε να επιλέγουμε να αγνοούμε την ενεργειακή μας σπατάλη. Με την ποιότητα της ζωής μας να είναι φανερό πως γίνεται ολοένα και πιο ενεργειακά απαιτητική, τόσο σε προσωπικό και εθνικό επίπεδο όσο και σε παγκόσμιο, και με τις τιμές της ηλεκτρικής ενέργειας να παρουσιάζουν συνεχείς διακυμάνσεις[30], και συνήθως προς τα πάνω, ακόμα και η παραμικρή προσπάθεια εξοικονόμησής της κρίνεται αναγκαία. Τη λύση σε αυτή την ανάγκη φαίνεται να δίνει η ίδια η τεχνολογία, παρέχοντας μας τα εργαλεία που χρειαζονται για να πετύχουμε αυτή την τόσο κρίσιμη εξοικονόμηση. Τις λεγόμενες λύσεις BMS. Στο επόμενο κεφάλαιο, γίνεται μια συνοπτική παρουσίαση των εργαλείων αυτών και στη συνέχεια η πτυχιακή αυτή εστιάζεται στο κύριο αντικείμενό της, το Home Assistant.

1.1 Έξυπνα κτήρια

Τα έξυπνα κτήρια αποτελούν μια ιδιάζουσα ανάγκη σήμερα, καθώς πέρα από τη βελτίωση της ποιότητας της ζωής μας, λειτουργούν και ως κύρια εργαλεία για την καταπολέμηση της ενεργειακής σπατάλης και της κλιματικής επιβάρυνσης. Ενώ παράλληλα συμβάλλουν καθοριστικά και στην ασφάλεια των υποδομών μας, τόσο σε φυσικούς κινδύνους, όπως σεισμοί, φωτιές και υγειονομικές κρίσεις, όσο και σε απειλές που απορρέουν από την ανθρώπινη δραστηριότητα όπως εγκληματικότητα, βανδαλισμοί και διάφορες άλλες απειλές κατά της ασφάλειάς μας.

1.1.1 Εξοικονόμηση ενέργειας και φυσικών πόρων

Οι προηγμένες τεχνολογίες που ενσωματώνουν τα έξυπνα κτήρια μας επιτρέπουν, με την σωστή αξιοποίησή τους, να εξοικονομήσουμε ενέργεια και να μειώσουμε την σπατάλη των φυσικών μας πόρων. Αυτό επιτυγχάνεται με την χρήση έξυπνων ρουτινών, συσκευών, διακοπών και αισθητήρων, όπου οι αισθητήρες «διαβάζουν» τον χώρο και έπειτα «λένε» στο κτήριο και τις συσκευές του πώς να αντιδράσει (αυτοματοποιημένη

ενέργεια). Ας πάρουμε για παράδειγμα ένα σπίτι το βράδυ, όταν οι ένοικοι του έχουν πάει για ύπνο, έχοντας ξεχάσει όμως όλα τα φώτα ανοιχτά. Ένα έξυπνο σπίτι θα πρέπει να είναι ικανό να καταλάβει ότι οι ένοικοί του κοιμούνται και ότι όλα τα φώτα του είναι αναμμένα χωρίς λόγο, έτσι ώστε να πάρει την «πρωτοβουλία» και να κλείσει όλα τα φώτα που δεν χρειάζονται τη δεδομένη στιγμή.

1.1.2 Έλεγχος περιβαλλοντικών συνθηκών

Τα έξυπνα κτήρια, μπορούν επίσης να ανταποκριθούν δυναμικά και αποτελεσματικά σε περιβαλλοντικές συνθήκες, εξασφαλίζοντας την ασφάλεια και τη βιωσιμότητα του χώρου. Όσον αφορά την ασφάλεια, θα πρέπει να είναι ικανό να καταλάβει την παρουσία π.χ. φωτιάς ή ενός σεισμού ώστε να ενεργοποιήσει τα συστήματα πυρασφάλειας και εκκένωσης αντίστοιχα. Στην περίπτωση της βιωσιμότητας, και σε συνέχεια από την παραπάνω παράγραφο της ενεργειακής εξοικονόμησης, το έξυπνο κτήριο θα πρέπει, για παράδειγμα, να μπορεί να ανιχνεύσει την πτώση ή την άνοδο της θερμοκρασίας ώστε να ρυθμίσει αντίστοιχα τους μηχανισμούς θέρμανσης, στους χώρους όπου εντοπίζει να υπάρχουν άτομα, εξασφαλίζοντας έτσι τη μέγιστη άνεση χωρίς την σπατάλη ενέργειας σε άδειους χώρους.

1.1.3 Συστήματα ασφαλείας σε κτήρια

Κίνδυνοι ωστόσο, είναι γεγονός πως δεν προέρχονται μόνο από το περιβάλλον, αλλά συχνά έχουν και ανθρώπινη διάσταση. Άρα εξίσου σημαντική είναι η ύπαρξη έξυπνων συστημάτων παρακολούθησης και συναγερμών που χρησιμοποιούν αισθητήρες (π.χ. κίνησης), κάμερες ασφαλείας και έξυπνες κλειδαριές. Έτσι το έξυπνο κτήριο θα είναι σίγουρα κλειδωμένο τις ώρες υψηλού κινδύνου και τις περιόδους απουσίας από αυτό, ενώ οι διάφορες κάμερες και οι αισθητήρες θα φροντίζουν για τη συνεχή επιτήρησή του και την άμεση ειδοποίησή μας με οποιοδήποτε μέσο το προγραμματίσουμε (π.χ. ειδοποίηση κινητού, email, SMS,...)· παρέχοντας έτσι ένα επίπεδο ασφαλείας που υπερβαίνει τα παραδοσιακά συστήματα.

1.2 Λύσεις BMS

Τα Building Management Systems (BMS) ή (στα ελληνικά) Συστήματα Διαχείρισης Κτηρίων δεν είναι κάτι καινούριο. Σύμφωνα και με τεχνικό άρθρο της ιστοσελίδας IoT

For All[2], υπάρχουν εδώ και τουλάχιστον 50 χρόνια και αποτελούν συστήματα σχεδιασμένα για την αυτοματοποίηση και τον έλεγχο των ηλεκτρομηχανολογικών εγκαταστάσεων ενός κτιρίου. Μέσω αυτών, βελτιστοποιείται η διαχείριση της ενέργειας, της θέρμανσης, του κλιματισμού, του φωτισμού και άλλων κρίσιμων λειτουργιών, με στόχο τη μείωση της κατανάλωσης και την αύξηση της αποδοτικότητας.

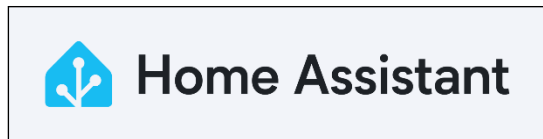
Τα ολοκληρωμένα συστήματα τύπου BMS, που ο εξοπλισμός τους είναι σχεδιασμένος να λειτουργεί ως μονάδα, αποτελούνται από έναν κεντρικό ελεγκτή (hub), ο οποίος συνδέεται με διάφορους αισθητήρες και ελεγκτές (ρελέ/relay) που συλλέγουν δεδομένα και εκτελούν εντολές με βάση αυτά. Οι αισθητήρες μπορούν να μετρούν θερμοκρασία, υγρασία, κίνηση, φωτεινότητα, κατανάλωση ενέργειας, κ.α.... Ενώ οι ελεγκτές, ελέγχουν τη λειτουργία συσκευών όπως θερμαντικά σώματα, κλιματιστικά, φώτα και ηλεκτρικά ρολά. Όλα αυτά ελέγχονται μέσω ενός πίνακα ελέγχου (dashboard ή SCADA system), που επιτρέπει στους εξουσιοδοτημένους χρήστες να παρακολουθούν την κατάσταση του κτιρίου σε πραγματικό χρόνο και να ρυθμίζουν τις παραμέτρους λειτουργίας του[3].

Σήμερα, υπάρχουν πολλές λύσεις και πλατφόρμες τύπου BMS. Από τις πιο απλές DIY λύσεις, όπως μικροελεγκτές Arduino, Raspberry Pi και ESP32, που χρησιμοποιούνται για την κατασκευή προσαρμοσμένων μηχανικών αυτοματισμών με απλούς αισθητήρες και χειροποίητες breadboard συνδέσεις[19], μέχρι και τις πιο εμπορικά ολοκληρωμένες πλατφόρμες όπως Google Home, Apple HomeKit και Amazon Alexa, που βασίζονται σε έξυπνους, αυτόνομους αισθητήρες, που προέρχονται από τους ίδιους ή διαφορετικούς κατασκευαστές και ενσωματώνονται απρόσκοπτα στα αντίστοιχα οικοσυστήματα[4].

Σε αντίθεση με τα Smart Home hub αυτών των γνωστών εταιριών, το Home Assistant δεν κάνει για χρήση «out of the box», ούτε προσφέρει την ευκολία και την ταχύτητα εγκατάστασης που προσφέρουν τα άλλα. Όπως χαρακτηριστικά λέει και το TheVerge[4], το Home Assistant «χρειάζεται χρόνο, υπομονή και skills». Όμως για αυτόν ακριβώς τον λόγο ξεχωρίζει. Για τον υψηλό βαθμό «τεχνικής και προγραμματιστικής ευελιξίας» που προσφέρει, δίνοντάς μας το δικαίωμα να του προσθέσουμε όποιο υλικολογισμικό κριθεί απαραίτητο από τον χρήστη. Όπως επίσης επιτρέποντας τη δημιουργία προσαρμοσμένων κανόνων και αυτοματισμών, με πλήρη ευελιξία και αρμονία με τις ανάγκες χρήσης.

1.3 Home Assistant

Ανάμεσα άρα στις πιο ευέλικτες λύσεις BMS και αυτή που αξιοποιείται στην παρούσα πτυχιακή, βρίσκεται το Home Assistant. Πρόκειται για μια πλατφόρμα που επιτρέπει τη διασύνδεση συσκευών από διαφορετικά οικοσυστήματα - είτε πρόκειται για επίσημους κατασκευαστές είτε για DIY λύσεις - προσφέροντας απόλυτη προσαρμογή και έλεγχο στον οικιακό και επαγγελματικό αυτοματισμό.



Εικόνα 1: Home Assistant logo[5].

Το Home Assistant (ή HA ή HAOS) είναι ένα δωρεάν λογισμικό ανοιχτού κώδικα (open source) που μπορεί να εγκατασταθεί σε σχεδόν οποιοδήποτε hardware υπολογιστή. Από έναν σύγχρονο υπολογιστή έως ένα παλιό λάπτοπ ή ένα μικρό Raspberry Pi. Ο κύριος στόχος του είναι να κάνει τη δημιουργία ενός «έξυπνου» σπιτιού ή κτηρίου πιο εύκολη και προσιτή, επιτρέποντας στον χρήστη να συνδυάζει συσκευές και αισθητήρες ανεξαρτήτως κατασκευαστή[6].

Σε αντίθεση με πολλές έτοιμες εμπορικές λύσεις που εξαρτώνται από cloud υπηρεσίες και κλειστά οικοσυστήματα και λογισμικά, το HA έχει σχεδιαστεί ώστε να λειτουργεί τοπικά και αυτόνομα, δίνοντας στον χρήστη τον πλήρη έλεγχο του συστήματός του, χωρίς εξαρτήσεις από εξωτερικούς παρόχους.

1.4 Χαρακτηριστικά και Οφέλη

Σύμφωνα με την πηγή [6], αλλά και με βάση την προσωπική εμπειρία και επαφή με το συγκεκριμένο λογισμικό, το HA ξεχωρίζει καθώς:

1. *Αναπτύχθηκε από open source κοινότητα.* Η φύση του ως open source λογισμικό σημαίνει ότι θα εξελίσσεται συνεχώς, με συνεισφορές από μια ιδιαίτερα ενεργή παγκόσμια κοινότητα. Αυτό εγγυάται διαρκείς βελτιώσεις, νέες δυνατότητες και σίγουρη αντιμετώπιση προβλημάτων.
2. *Δυνατότητα τοπικής λειτουργίας και πλήρους ανεξαρτησίας.* Ένα από τα βασικά πλεονεκτήματα του Home Assistant είναι ότι δεν απαιτεί μόνιμη σύνδεση σε εξωτερικούς servers και -κατά συνέπεια- δεν χρειάζεται το διαδίκτυο για να λειτουργήσει. Έτσι εξασφαλίζεται:

- *Ιδιωτικότητα*, αφού τα δεδομένα και οι αυτοματισμοί μας παραμένουν αποκλειστικά στο τοπικό δίκτυο του σπιτιού/κτηρίου μας.
 - *Αξιοπιστία*, καθώς χωρίς την ανάγκη για επικοινωνία με εξωτερικούς servers, το σύστημα συνεχίζει να λειτουργεί ακόμα και χωρίς σύνδεση στο διαδίκτυο.
 - *Άμεση απόκριση των εντολών και άμεση λήψης της πληροφορίας από τους αισθητήρες*, αφού λόγω τοπικής συνδεσιμότητας, οι συσκευές «μιλάνε» απευθείας μεταξύ τους και, χωρίς η πληροφορία να χρειάζεται να διασχίσει το διαδίκτυο και απομακρυσμένους cloud servers, η καθυστέρηση (latency) μειώνεται σημαντικά.
3. *Ευρεία υποστήριξη συσκευών και πρωτοκόλλων*. Ανάλογα πάντα και με το hardware που του έχουμε συνδέσει, το HA είναι συμβατό με αμέτρητα προϊόντα και τεχνολογίες όπως Zigbee, Z-Wave, Matter, MQTT, Thread κ.ά., επιτρέποντας την απρόσκοπτη ενσωμάτωση συσκευών, από διαφορετικούς κατασκευαστές.
 4. *Ευκολία και ευελιξία στους αυτοματισμούς*. Οι χρήστες-admin μπορούν να δημιουργήσουν σύνθετες αυτοματοποιήσεις, είτε μέσω γραφικού περιβάλλοντος είτε με κώδικα (YAML, Python, Jinja2), ανάλογα με το επίπεδο εμπειρίας τους.
 5. *Ευκολία στον διαμοιρασμό αυτοματισμών και ρουτινών*. Οι αυτοματισμοί και τα script σενάρια, μπορούν πολύ εύκολα να διαμοιραστούν (μέσω YAML κώδικα) και να προσαρμοστούν, διευκολύνοντας την ανταλλαγή τους στην κοινότητα καθώς και την αποθήκευσή τους για χρήση σε δεύτερο χρόνο από τους χρήστες της κάθε εγκατάστασης.
 6. *Προσαρμόσιμα Dashboards ελέγχου*. Με δυνατότητα δημιουργίας εξατομικευμένων περιβαλλόντων ελέγχου και επιπλέον δυνατότητα προβολής γραφημάτων, στατιστικών και ειδοποιήσεων, για πλήρη εικόνα της λειτουργίας του συστήματος.
 7. *Εξατομικευμένη πρόσβαση και διαχείριση χρηστών*. Κάθε χρήστης μπορεί να έχει το δικό του επίπεδο πρόσβασης, κάτι που διευκολύνει την ασφαλή και εύκολη χρήση του συστήματος σε οικιακό και επαγγελματικό περιβάλλον.

1.5 Μειονεκτήματα

Παρά τα σημαντικά πλεονεκτήματά του, το Home Assistant εμφανίζει και ορισμένες αδυναμίες, που αφορούν κυρίως χρήστες χωρίς τεχνικό υπόβαθρο. Πιο συγκεκριμένα:

1. *Απαιτεί τεχνικές γνώσεις.* Η πλήρης αξιοποίηση του Home Assistant προϋποθέτει βασικές τεχνικές γνώσεις, τόσο υλικού (hardware) όσο και λογισμικού (software), καθώς, η σωστή λειτουργία του συστήματος εξαρτάται σε μεγάλο βαθμό από το διαθέσιμο hardware και την ποιότητα της διαδικτυακής υποδομής. Για παράδειγμα, χωρίς τον κατάλληλο Zigbee coordinator, το HA δεν θα μπορεί να επικοινωνεί με τις Zigbee συσκευές, ενώ η κακή υποδομή δικτύου μπορεί να ενισχύσει τον θόρυβο μεταξύ συσκευών και συνεπώς να μην επικοινωνούν σωστά ή ομαλά μεταξύ τους. Παράλληλα, αν και σε μεγάλο βαθμό η ρύθμισή του μπορεί να γίνει απλά και συμβατικά μέσα από το UI, σίγουρα κάποια στιγμή θα υπάρξει η ανάγκη ρύθμισής του μέσω κώδικα (κυρίως YAML, Python και Jinja2).
2. *Πολυπλοκότητα εγκατάστασης.* Το HA, διατίθεται σε διάφορες εκδόσεις (OS, Container, Core, Supervised), οι οποίες απευθύνονται σε διάφορα σενάρια χρήσης. Κατά συνέπεια, ο χρήστης θα πρέπει να είναι ικανός να ξεχωρίζει ποια έκδοση συμπίπτει με τις ανάγκες του και επίσης να ξέρει πώς και πού (σε τι hardware) να το εγκαταστήσει ώστε να επωφεληθεί πλήρως από αυτό. [7]
3. *Έλλειψη επίσημης υποστήριξης.* Αν και τα τελευταία χρόνια έχει δημιουργηθεί μια ομάδα, η *Nabu Casa*[18], όπου επιχειρεί να καταστήσει το HA εμπορικά πιο προσβάσιμο και πιο “plug-n-play”, η επίλυση προβλημάτων εξακολουθεί να βασίζεται στην αυτονομία του χρήστη και στη συνεισφορά της opensource κοινότητας. Αυτό, πέρα της σύγχυσης που μπορεί να δημιουργεί, ενδέχεται να καθυστερεί και τη διόρθωση σφαλμάτων και bugs, ειδικά σε περιπτώσεις σύνθετων δυσλειτουργιών.

1.6 Οι στόχοι της εργασίας

Η παρούσα εργασία έχει ως στόχο, την εφαρμογή του λογισμικού Home Assistant στον χώρο του «Εργαστηρίου 4» του Τμήματος Ψηφιακών Συστημάτων, με σκοπό τη μετατροπή του σε έναν έξυπνο χώρο διδασκαλίας και τη μελέτη των τρόπων με τους οποίους

μπορεί να επιτευχθεί εξοικονόμηση ενέργειας με τη χρήση έξυπνων αυτοματισμών, αισθητήρων και συσκευών. Καθώς και τον προσδιορισμό του βαθμού αποτελεσματικότητας τους.

Πιο συγκεκριμένα, οι επιμέρους στόχοι είναι:

1. Μετατροπή των απλών συμβατικών διακοπών για τον χειρισμό των σκιάστρων και των φωτών του εργαστηρίου σε έξυπνα, με τη χρήση έξυπνων Shelly ρελέ και ενσωμάτωσή τους στο Home Assistant.
2. Εξοικονόμηση ενέργειας μέσω αυτοματισμών. Σχεδίαση και υλοποίηση έξυπνων σεναρίων που αξιοποιούν τις έξυπνες συσκευές και τους αισθητήρες με στόχο τη βέλτιστη χρήση του φυσικού και του τεχνητού φωτισμού, καθώς και της θερμοκρασίας του χώρου (έμμεσα).
3. Συλλογή και ανάλυση δεδομένων κατανάλωσης, με καταγραφή σε πραγματικό χρόνο της κατανάλωσης τόσο των συσκευών, προκειμένου να αξιολογηθεί η αποδοτικότητα των προτεινόμενων λύσεων.
4. Γνωριμία με το Home Assistant και κατανόηση των λειτουργιών και των δυνατοτήτων που προσφέρει, καθώς και των τυχόν περιορισμών, τόσο σε επίπεδο ρυθμίσεων και αυτοματισμών όσο και σε επίπεδο διαχείρισης των συσκευών.
5. Δημιουργία ενός όμορφου και λειτουργικού dashboard για τον έλεγχο και την παρακολούθηση του συστήματος, με στόχο την απλότητα και την επεκτασιμότητα της εγκατάστασης, τόσο για τους απλούς χρήστες όσο και για τους διαχειριστές του συστήματος.

1.7 Αναμενόμενα αποτελέσματα

Με την ολοκλήρωση της παρούσας πτυχιακής εργασίας, αναμένεται να μπορεί να επιτευχθεί σημαντική εξοικονόμηση ενέργειας, μέσω της δημιουργίας και εφαρμογής έξυπνων αυτοματισμών, οι οποίοι θα βασίζονται σε πραγματικά δεδομένα κατανάλωσης και περιβαλλοντικών συνθηκών. Η μετατροπή των σκιάστρων του «Εργαστηρίου 4», όπως και του φωτισμού, από «απλά» σε «έξυπνα», εκτιμάται ότι θα συμβάλει ουσιαστικά στην εξοικονόμηση αυτή, μειώνοντας της χρήσης τεχνητού φωτισμού και θέρμανσης. Ενώ η ενσωμάτωση και αξιοποίηση έξυπνων αισθητήρων κίνησης και θερμοκρασίας θα επιτρέψει την ακριβέστερη ρύθμιση των αυτοματισμών για την καλύτερη διαχείριση της ενέργειας στον χώρο.

Παράλληλα, η δημιουργία ενός ολοκληρωμένου και χρηστικού web interface ελέγχου (dashboard), θα διευκολύνει την παρακολούθηση της κατανάλωσης και της κατάστασης (state) της κάθε συσκευής και αισθητήρα, αναδεικνύοντας έτσι και τη χρηστική αρτιότητα της εγκατάστασης και του συστήματος.

Συνολικά, τα αναμενόμενα αποτελέσματα θα πρέπει να υποδεικνύουν ότι τέτοιου είδους τεχνολογίες έχουν τη δυναμική να συμβάλουν ουσιαστικά στη βελτίωση της καθημερινής μας ζωής και στη μετάβασή μας προς ένα πιο βιώσιμο και ενεργειακά αποδοτικό περιβάλλον, χωρίς να στερούμαστε τις όποιες ανέσεις μας.

2 Σχεδιασμός Έξυπνης Αίθουσας Διδασκαλίας

Στο κεφάλαιο αυτό, παρουσιάζεται ο εξοπλισμός ο οποίος χρησιμοποιήθηκε για την πρακτική εφαρμογή της πτυχιακής, καθώς και η διαδικασία εγκατάστασής του στο «Εργαστήριο 4» του Τμήματος των Ψηφιακών Συστημάτων. Περιγράφονται αναλυτικά τόσο οι βασικές συσκευές όσο και κάθε επιπλέον εξοπλισμός που κρίθηκε απαραίτητος για την ολοκλήρωση της εγκατάστασης. Στόχος της παρουσίασης αυτής, είναι να αναδείξει την υλικοτεχνική υποδομή που απαιτήθηκε για την υλοποίηση και την καλή λειτουργία των αυτοματισμών μέσω του Home Assistant.

2.1 Χώρος Υλοποίησης

Η πρακτική εφαρμογή, υλοποιήθηκε στο «Εργαστήριο 4», το οποίο βρίσκεται στο δεύτερο κτήριο της Σχολής Τεχνολογίας του Πανεπιστημίου Θεσσαλίας.

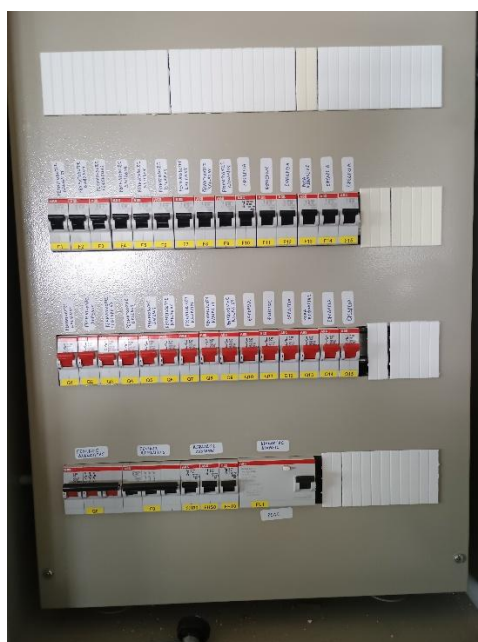


Εικόνα 2: Το Εργαστήριο 4 εσωτερικά.

Η επιλογή του συγκεκριμένου χώρου, ως χώρος υλοποίησης, δεν ήταν τυχαία. Ο χώρος αυτός, αποτελεί μία σύγχρονη αίθουσα διδασκαλίας πληροφορικής, εξοπλισμένη με σταθερούς υπολογιστές, τρία μηχανοκίνητα ρολά ασφαλείας και σκίασης παραθύρων και

με μία βασική επαγγελματική ηλεκτρολογική υποδομή. Χαρακτηριστικά που τον καθιστούν τεχνικά κατάλληλο για την υλοποίηση λύσεων αυτοματισμού. Ενώ ακόμα, σαν φοιτητής -και όποτε φυσικά το ζητούσα κι από τον υπεύθυνο καθηγητή μου- είχα άμεση πρόσβαση στον χώρο. Γεγονός που μου επέτρεψε να προχωρήσω με την εγκατάσταση των έξυπνων συσκευών και την τακτική παρακολούθηση της καλής λειτουργίας του συστήματος.

Είναι επίσης σημαντικό να αναφερθεί πως ο χώρος διαθέτει ηλεκτρολογικό πίνακα με δυνατότητα φιλοξενίας επιπλέον έξυπνων διακοπών ρελέ, τύπου ράγας, με λειτουργία μέτρησης κατανάλωσης, ανοίγοντας τον δρόμο για περαιτέρω ανάπτυξη των αυτοματισμών, σε επίπεδο ηλεκτρολογικής εγκατάστασης.



Εικόνα 3: Ο ηλεκτρολογικός πίνακας της αίθουσας.

Επιπλέον, πρόκειται για μια αίθουσα με θεωρητικά αυξημένη ενεργειακή κατανάλωση λόγω του όλου εξοπλισμού (ηλεκτρονικών υπολογιστών, του επαγγελματικού φωτισμού και των μηχανοκίνητων σκιάστρων), κάνοντας έτσι ουσιαστικό το ενδιαφέρον για την εφαρμογή λύσεων εξοικονόμησης ενέργειας.

Τέλος, η ύπαρξη επεκτάσιμης ηλεκτρολογικής υποδομής και η φύση του χώρου ως μία μοντέρνα αίθουσα διδασκαλίας, μου παρείχαν την ελευθερία να πειραματιστώ με τη διάταξη της εγκατάστασης και τα διάφορα σενάρια ελέγχου, χωρίς τους περιορισμούς που θα είχα σε άλλους χώρους. Όλα τα παραπάνω συνέβαλαν ώστε, να αποτελέσει το κατάλληλο περιβάλλον για την πιλοτική εγκατάσταση και αξιολόγηση των αυτοματισμών, μέσω του Home Assistant.

2.2 Εγκατάσταση Home Assistant

Όπως αναφέρθηκε και στην εισαγωγή, το Home Assistant χρειάζεται να εγκατασταθεί σε κάποιο hardware και να λειτουργεί 24/7. Στα επίσημα εγχειρίδια εγκατάστασης του Home Assistant [7], αναφέρεται πως αυτό μπορεί να γίνει με δύο βασικούς τρόπους:

- Με την επίσημη διανομή Home Assistant OS
- Ή μέσω εγκατάστασης σε Container (π.χ. Docker).

Η έκδοση σε Container όμως, φαίνεται να υπολείπεται σε δυνατότητες, καθώς δεν υποστηρίζει τα Add-ons και δεν προσφέρει One-click updates.

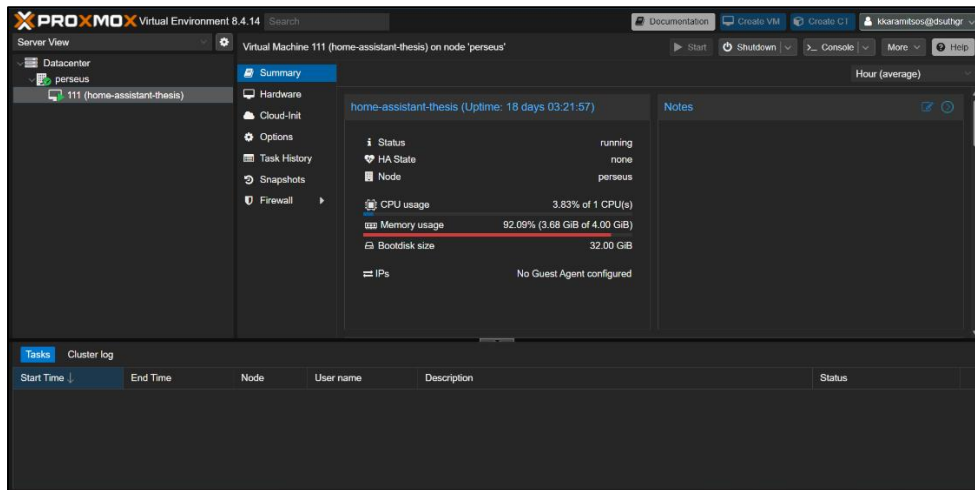
	HA OS ¹	Container ¹
Automations	✓	✓
Dashboards	✓	✓
Integrations	✓	✓
Add-ons	✓	✗
Blueprints	✓	✓
One-click updates	✓	✗
Backups	✓	✓

Εικόνα 4: Home Assistant installation types[7].

Έτσι λοιπόν, για τη συγκεκριμένη υλοποίηση, το Home Assistant εγκαταστάθηκε ως εικονική μηχανή (Virtual Machine) σε έναν Proxmox VE Hypervisor server, αξιοποιώντας μια «.qcow2» OS version του. Η επιλογή αυτή συνδυάζει τα πλεονεκτήματα της πλήρους υποστήριξης των λειτουργιών του HA, με την ευελιξία διαχείρισης που προσφέρει ένα περιβάλλον virtualization.

Ο Proxmox (Host) server βρίσκεται εντός του γειτονικού «Εργαστηρίου 3», όπου και φιλοξενείται μέρος της κεντρικής δικτυακής υποδομής του κτηρίου. Αποτελεί έναν υπολογιστή του τμήματος, ειδικά σχεδιασμένος από την ομάδα τεχνικής υποστήριξης, για σενάρια virtualization και είναι μόνιμα συνδεδεμένος στο τοπικό δίκτυο μέσω ethernet, εξασφαλίζοντας σταθερή επικοινωνία με το δίκτυο. Η εικονική μηχανή που έχει δημιουρ-

γηθεί σχετικά, έχει διατεθειμένο 1 πυρήνα επεξεργαστή (αν και τα σχετικά manual συνιστούν τουλάχιστον 2), 32GB αποθηκευτικό χώρο και είναι ρυθμισμένη να χρησιμοποιεί τουλάχιστον 2GB με έως 4GB μνήμης RAM [20]. Το γεγονός αυτό αποδεικνύει πως η πλατφόρμα μπορεί να λειτουργήσει αξιόπιστα ακόμη και σε περιβάλλοντα με αρκετά περιορισμένες δυνατότητες hardware.



Εικόνα 5: Screenshot από το UI του Proxmox.

Για την πρόσβαση του κάθε διαχειριστή και χρήστη στο UI του Home Assistant (αλλά και στον Proxmox server, για την εγκατάσταση του HA), απαιτείται μία client συσκευή («πελάτης»). Η συσκευή αυτή μπορεί να είναι είτε ένας σταθερός υπολογιστής, είτε ένας φορητός (laptop), είτε ακόμα και ένα έξυπνο κινητό τηλέφωνο (smartphone). Στο πλαίσιο της παρούσας εργασίας, χρησιμοποιήθηκαν και οι τρεις τύποι συσκευών για την πρόσβαση και διαχείριση του HA.

2.3 Network setup

Η ρύθμιση του δικτύου αποτελεί θεμελιώδες στάδιο για την επιτυχημένη εγκατάσταση του Home Assistant και των έξυπνων συσκευών, καθώς και για την απρόσκοπτη εκτέλεση των εκάστοτε αυτοματισμών. Για τον λόγο αυτόν, κατόπιν αιτήματος στο Κέντρο Δικτύου του πανεπιστημίου και σε συνεννόηση πάντα με την ομάδα τεχνικής υποστήριξης, δημιουργήθηκε ένα αποκλειστικό SSID Wi-Fi δίκτυο όπως και ένα συγκεκριμένο εύρος local IP διευθύνσεων. Οι διευθύνσεις αυτές χρησιμοποιήθηκαν για την στατική διευθυνσιοδότηση των έξυπνων συσκευών, του Hypervisor Server, καθώς και του ίδιου του HA. Για την απομακρυσμένη πρόσβαση στην παραπάνω εγκατάσταση, χρησιμοποιείται σύνδεση μέσω VPN, η οποία επιτρέπει ασφαλή επικοινωνία χωρίς την ανάγκη

για «άνοιγμα» δημόσιων πυλών (ports) στο διαδίκτυο. Πλέον, είμαστε έτοιμοι να προχωρήσουμε στην εγκατάσταση και σύνδεση των αισθητήρων και συσκευών έξυπνου ελέγχου.

2.4 Ρύθμιση Home Assistant και εργαλεία ανάπτυξης (Add-ons)

Μετά την ολοκλήρωση της βασικής εγκατάστασης του Home Assistant, τόσο σε επίπεδο hardware (ενότητα 2.2) όσο και σε επίπεδο λογισμικού - όπου ολοκληρώθηκε μέσω του γραφικού περιβάλλοντος του Home Assistant και σύμφωνα με τις οδηγίες που αυτό περιέγραφε - η μόνη κρίσιμη ρύθμιση που έπρεπε να ακολουθήσει ήταν ο ορισμός στατικής IP. Αυτό πραγματοποιήθηκε πολύ εύκολα μέσα από Settings(Ρυθμίσεις)->System(Σύστημα)-> Network(Δίκτυο) και στο πεδίο «Configure network interfaces», όπου αφού δηλώθηκαν τα απαραίτητα IPv4 στοιχεία, έγινε αποθήκευση και επανεκκίνηση.

Ακόμα, για την ενίσχυση των δυνατοτήτων ανάπτυξης (κώδικα, αυτοματισμών, template sensors...) και παραμετροποίησης του συστήματος σημαντικό ήταν να γίνει εγκατάσταση των add-ons *Studio Code Server* και του *Advanced SSH & Web Terminal*. Παρακάτω, περιγράφονται τι είναι το κάθε ένα καθώς και οι δυνατότητες και οι λειτουργίες τους.

2.4.1 Studio Code Server

Το συγκεκριμένο πρόσθετο (add-on) παρέχει ένα ενσωματωμένο περιβάλλον κώδικα, εντός του Home Assistant, τύπου *Visual Studio Code* επιτρέποντας την άμεση τροποποίηση αρχείων ρυθμίσεων (όπως αρχεία secrets) και την ανάπτυξη αυτοματισμών και ρουτινών και έτσι κάνοντάς την πιο άμεση και εύκολη, σε σχέση με τον συμβατικό τρόπο μέσω του UI. Σύμφωνα και με την επίσημη τεκμηρίωση [38] της (open-source) κοινότητας του HA, το συγκεκριμένο add-on βασίζεται στην πλατφόρμα code-server η οποία επιτρέπει την πρόσβαση σε ένα περιβάλλον *Visual Studio Code* απευθείας από κάποιον φυλλομετρητή (browser) για όλο το configuration περιεχόμενο του HA. Μάλιστα, ο *Studio Code Server* ενσωματώνεται πλήρως στο περιβάλλον του HA παρέχοντας προεγκατεστημένες και προκαθορισμένες επεκτάσεις όπως:

- Home Assistant extension για υποστήριξη αυτοματισμών και αισθητήρων τύπου template.
- MDI icons extension για χρήση εικονιδίων στο UI.

- YAML extension για σύνταξη και αυτόματη συμπλήρωση αρχείων ρυθμίσεων.
- Η εγκατάσταση του add-on γίνεται πολύ εύκολα με τα βήματα να περιλαμβάνουν:
1. Αναζήτηση του πρόσθετου από την σελίδα αναζήτησης add-ons εσωτερικά του Home Assistant (<http://x.x.x.x:8123/hassio/store>).
 2. Επιλογή του add-on και κλικ στην επιλογή «Install» για την εγκατάσταση.
 3. *Προαιρετικά:* Πριν την εκκίνηση του πρόσθετου, καλό είναι να ενεργοποιηθούν οι εξής επιλογές:
 - a. Start on boot: ώστε το πρόσθετο να εκκινείται αυτόματα με την εκκίνηση του συστήματος.
 - b. Watchdog: για αυτόματη επανεκκίνηση του πρόσθετου σε περίπτωση σφάλματος ή απρόσμενου τερματισμού.
 - c. Add to sidebar: ώστε να εμφανίζεται το πρόσθετο στο πλευρικό μενού του Home Assistant για εύκολη πρόσβαση.
 4. Εκκίνηση του πρόσθετου μέσω της επιλογής «Start».
 5. *Προαιρετικά:* Έλεγχος των καταγραφών (logs) για επιβεβαίωση της ορθής εκκίνησης και λειτουργίας.
 6. Πρόσβαση στο περιβάλλον του *Studio Code Server* μέσω της επιλογής «Open web UI»

2.4.2 Advanced SSH & Web Terminal

Το συγκεκριμένο πρόσθετο (add-on) κρίνεται απαραίτητο και ένα ισχυρό εργαλείο για την απομακρυσμένη πρόσβαση, ρύθμιση και διαχείριση του Home Assistant, μέσω γραμμής εντολών. Σύμφωνα με το documentation του[39], πρόκειται για μια ενισχυμένη έκδοση του βασικού SSH add-on που προσφέρει η κοινότητα του HA, με έμφαση στην ασφάλεια, την ευχρηστία και την ευελιξία. Η χρήση του προσθέτου επιτρέπει την εκτέλεση βασικών εντολών συστήματος και πρόσβασης αρχείων ενώ επιπλέον περιλαμβάνει εργαλεία και δυνατότητες όπως:

- Προεγκατεστημένα εργαλεία όπως curl, wget, rsync, git, nmap, mosquito client, MariaDB/MySQL client, Awake(«wake on LAN»), nano, vim, tmux, καθώς και εργαλεία δικτύου.
- Πρόσβαση σε hardware όπως audio, uart/serial, gpio pins και dbus
- Δυνατότητα εγκατάστασης custom Alpine packages κατά την εκκίνηση, ώστε να είναι διαθέσιμα κάθε φορά που γίνεται login.

- Δυνατότητα εκτέλεσης custom εντολών κατά την εκκίνηση του προσθέτου, για εξατομίκευση του περιβάλλοντος shell.
- ZSH shell με προεγκατεστημένο το "Oh My ZSH" και επιλεγμένα plugins για βελτιωμένη εμπειρία χρήσης.
- Λειτουργία σε επίπεδο host δικτύου, επιτρέποντας το άνοιγμα θυρών ή την εκτέλεση μικρών υπηρεσιών (daemons).
- Πρόσβαση στο Docker instance του host συστήματος για προχωρημένες λειτουργίες διαχείρισης.

Η εγκατάσταση του add-on γίνεται το ίδιο εύκολα και με τα ίδια σχεδόν βήματα με αυτά του *Studio Code Server* (υποενότητα 2.4.1):

1. Αναζήτηση του πρόσθετου από την σελίδα αναζήτησης add-ons εσωτερικά του Home Assistant (<http://x.x.x.x:8123/hassio/store>).
2. Επιλογή του add-on και κλικ στην επιλογή «Install» για την εγκατάσταση.
3. Διαμόρφωση των πεδίων *username* και *password* για ασφαλή πρόσβαση μέσω ssh.
4. Προαιρετικά: Πριν την εκκίνηση του πρόσθετου, καλό είναι να ενεργοποιηθούν οι εξής επιλογές:
 - a. Start on boot: ώστε το πρόσθετο να εκκινείται αυτόματα με την εκκίνηση του συστήματος.
 - b. Watchdog: για αυτόματη επανεκκίνηση του πρόσθετου σε περίπτωση σφάλματος ή απρόσμενου τερματισμού.
 - c. Add to sidebar: ώστε να εμφανίζεται το πρόσθετο στο πλευρικό μενού του Home Assistant για εύκολη πρόσβαση.
 - d. Protection mode: για περιορισμό πρόσβασης σε ευαίσθητες λειτουργίες του συστήματος (για μη έμπειρους διαχειριστές-χρήστες).
5. Εκκίνηση του πρόσθετου μέσω της επιλογής «Start».
6. Προαιρετικά: Έλεγχος των καταγραφών (logs) για επιβεβαίωση της ορθής εκκίνησης και λειτουργίας.
7. Πρόσβαση στο περιβάλλον του *Web Terminal* μέσω της επιλογής «Open web UI»

2.5 Αισθητήρες και συσκευές έξυπνου ελέγχου

Το HA, όπως αναφέρθηκε και στην εισαγωγή, υποστηρίζει τη δυνατότητα σύνδεσης με ένα ευρύ φάσμα συσκευών, είτε μέσω cloud υπηρεσιών (π.χ. Tuya ή Sonoff συσκευές)

είτε μέσω τοπικού ελέγχου (π.χ. Shelly συσκευές)[21]. Δεδομένου ότι στόχος της παρούσας πτυχιακής είναι η αξιοποίηση στο έπακρο των δυνατοτήτων τοπικού ελέγχου, η επιλογή του εξοπλισμού έγινε με γνώμονα την ελαχιστοποίηση της εξάρτησης σε εξωτερικές υπηρεσίες και υποδομές cloud. Στο πλαίσιο αυτό, η πλειονότητα των συσκευών που επιλέχθηκαν, προέρχονται από την εταιρία Shelly, η οποία προσφέρει έξυπνα ρελέ και αισθητήρες που λειτουργούν αυτόνομα, χωρίς την ανάγκη για σύνδεση με το cloud. Οι συσκευές αυτές, είναι ιδιαίτερα ευέλικτες και με μια γρήγορη αναζήτηση στο διαδίκτυο, γίνεται γρήγορα αντιληπτό πως διακρίνονται στην smart home κοινότητα για την ευκολία στη ρύθμιση τους[9], ενώ υπάρχει και έτοιμο integration για την ενσωμάτωσή τους στο HA. Στη συνέχεια παρουσιάζονται αναλυτικά οι αισθητήρες και οι έξυπνες συσκευές που αξιοποιήθηκαν στην εργασία αυτή.

2.5.1 Shelly Motion 2

Το Shelly Motion 2, σύμφωνα με την επίσημη τεκμηρίωση της εταιρείας Shelly[10], αποτελεί έναν Wi-Fi αισθητήρα κίνησης, υψηλής ευαισθησίας και ιδιαίτερα χαμηλής κατανάλωσης, σχεδιασμένο για χρήση σε εσωτερικούς χώρους. Πέρα από την κύρια αυτή λειτουργία του, η συσκευή διαθέτει επίσης αισθητήρες θερμοκρασίας και φωτεινότητας, καθιστώντας περιττή την ανάγκη για επιπλέον μονάδες αισθητήρων και απλοποιώντας σημαντικά την εγκατάσταση και τη συντήρηση του συστήματος. Επιπλέον, ενσωματώνει ένα επιταχυνσιόμετρο για ανίχνευση κραδασμών και προστασία από παραβιάσεις (tamper detection), ενισχύοντας την ασφάλεια του συστήματος.

Η συσκευή τροφοδοτείται από επαναφορτιζόμενη μπαταρία 6500 mAh, η οποία προσφέρει ιδιαίτερα μεγάλη αυτονομία. Σύμφωνα με τις προδιαγραφές του, μπορεί να μείνει σε σύνδεση με το δίκτυο για έως και 3 χρόνια σε κατάσταση αναμονής ή περίπου για 12 έως 18 μήνες σε κανονική λειτουργία, με καθημερινή ανίχνευση κίνησης, χωρίς την ανάγκη επαναφόρτισης.



Εικόνα 6: Shelly Motion 2 device[10].

2.5.2 Μετεωρολογικός σταθμός GreenDigital

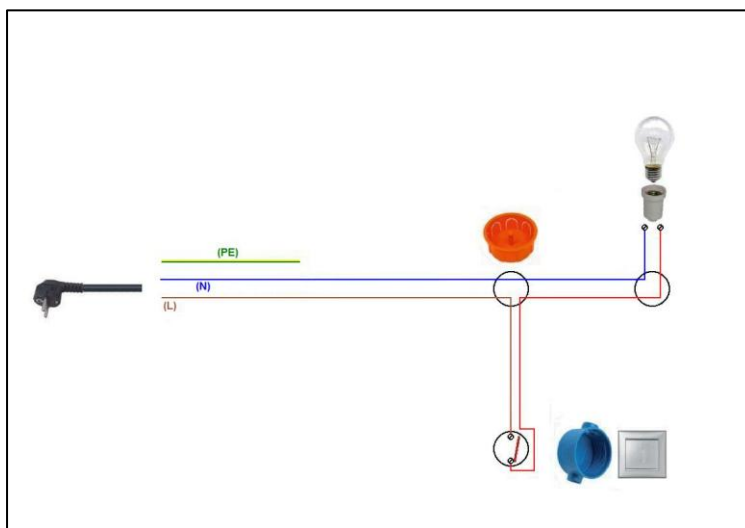
Το Τμήμα Ψηφιακών Συστημάτων του Πανεπιστημίου Θεσσαλίας, φιλοξενεί την ερευνητική ομάδα GreenDigital, η οποία ασχολείται με την ανάπτυξη ψηφιακών διδύμων με στόχο την αντιμετώπιση προκλήσεων που σχετίζονται με τη βιώσιμη ανάπτυξη[13]. Στο πλαίσιο των δραστηριοτήτων της, η ομάδα διατηρεί έναν πλήρως εξοπλισμένο μετεωρολογικό σταθμό, ικανό να καταγράφει και να μεταδίδει κρίσιμες μετρήσεις που σχετίζονται με τις τοπικές συνθήκες. Για τις ανάγκες της παρούσας μελέτης, μου παραχωρήθηκε η πρόσβαση στον σταθμό αυτόν με σκοπό οι μετρήσεις του να ενσωματωθούν στο περιβάλλον του Home Assistant, ώστε να αξιοποιηθούν τόσο σε επίπεδο αυτοματισμών όσο και στην λειτουργική παρουσίασή τους μέσω του αντίστοιχου dashboard (λεπτομέρειες για την ενσωμάτωσή τους στο Home Assistant, υπάρχουν στην ενότητα 3.4.3).



2.5.3 Shelly Plus 1PM & 2PM

Τα Shelly Plus 1PM και Shelly Plus 2PM, αποτελούν εξελιγμένα έξυπνα ρελέ, σχεδιασμένα για εφαρμογές οικιακού και επαγγελματικού εξοπλισμού. Όπως και το όνομά τους προδιαθέτει, το 1PM διαθέτει έναν ανεξάρτητο διακόπτη ρελέ με δυνατότητα παρακολούθησης κατανάλωσης ισχύος (Power Metering), ενώ το 2PM ενσωματώνει δύο ανεξάρτητα κανάλια ρελέ, επίσης με λειτουργία μέτρησης ενέργειας. Η συνδεσιμότητά τους με το τοπικό έξυπνο οικοσύστημα γίνεται μέσω Wi-Fi, ενώ η αρχική ζεύξη (pairing) μπορεί να ολοκληρωθεί πλήρως τοπικά, χωρίς την ανάγκη δημιουργίας λογαριασμών σε cloud υπηρεσίες, διευκολύνοντας τη διαδικασία εγκατάστασης[11][12].

Χρήσιμο είναι επίσης να αναφερθεί πως είναι ιδιαίτερα μικρά σε μέγεθος και σε συνδυασμό με τα ειδικά πλαίσια διακοπών Shelly που διαθέτει η εταιρία, μπορούν πολύ εύκολα να αντικαταστήσουν τους τυπικούς διακόπτες τοίχου. Το μοναδικό πρόβλημα που μπορεί να προκύψει - και πράγματι εμφανίζεται στις περισσότερες περιπτώσεις – αφορά την ανάγκη προσθήκης ουδέτερου αγωγού στο σημείο του διακόπτη. Η τροποποίηση αυτή αν και θεωρείται ελάχιστης κλίμακας[14], καθίσταται απαραίτητη λόγω της συνήθους πρακτικής στις ηλεκτρολογικές καλωδιώσεις διακοπών, στις οποίες δεν συνηθίζεται η καλωδίωση ενός κοινού διακόπτη να συνοδεύεται από τον ουδέτερο αγωγό του κάθε κυκλώματος



Εικόνα 8: Παραστατικό σχέδιο συνδεσμολογίας απλού διακόπτη[22].

3 Εγκατάσταση εξοπλισμού και ρύθμιση

3.1 Εγκατάσταση Shelly Motion 2

Η φυσική εγκατάστασή του Shelly Motion 2 αποδείχτηκε ιδιαίτερα εύκολη καθώς δεν απαιτήσε εξειδικευμένες τεχνικές γνώσεις ή εργαλεία. Το μόνο που χρειάστηκε ήταν να εντοπιστεί ένα κατάλληλο σημείο στον τοίχο, το οποίο να προσφέρει καθαρό οπτικό πεδίο προς τον χώρο ενδιαφέροντος. Η στερέωση πραγματοποιήθηκε εύκολα, χάρη στην αυτοκόλλητη βάση που διαθέτει η συσκευή, χωρίς την ανάγκη για τρύπες και βίδες στον τοίχο.

3.2 Εγκατάσταση Shelly Plus 1PM & 2PM

Αρχικά, έγινε ένας έλεγχος της υπάρχουσας ηλεκτρολογικής εγκατάστασης, πίσω από τους 5 τυπικούς διακόπτες ενδιαφέροντος (τρεις διπλοί τύπου μπουτόν για τα σκίαστρα και 2 μονοί για τα φώτα τύπου «αλέ ρετούρ»).

Αξίζει να σημειωθεί πως οι συσκευές που περιγράφονται λειτουργούν με τάση δικτύου 220V, γεγονός που συνεπάγεται αυξημένες απαιτήσεις ως προς την ασφάλεια κατά την εγκατάστασή τους. Η υλοποίηση τέτοιων παρεμβάσεων πρέπει να γίνονται από ή με συνοδεία αδειούχου και έμπειρου τεχνικού, καθώς η λανθασμένη διαχείριση μπορεί να επιφέρει σοβαρούς κινδύνους ηλεκτροπληξίας.

Μετά τον έλεγχο, κρίθηκε ότι το πιο κατάλληλο σημείο για την σύνδεση και την τοποθέτηση των Shelly ήταν στα κουτιά διακλάδωσης, ψηλά στον τοίχο, καθώς δεν χώραγαν πίσω από τους τυπικούς διακόπτες, παρά το μικρό τους μέγεθος σαν συσκευές. Ακολουθεί μια αναλυτική παρουσίαση της εγκατάστασης και ρύθμισής τους.



Εικόνα 9: Η καλωδίωση πίσω από τον ένα διπλό διακόπτη των δυτικών σκιάστρων, με την προσθήκη επιπλέον φάσης (μαύρο καλώδιο).

3.2.1 Εγκατάσταση ανατολικού σκιάστρου

Η εγκατάσταση του Shelly Plus 2PM στο ανατολικό σκίαστρο αποδείχτηκε η πιο απλή όλων. Αφού ανοίχθηκε το (μεγάλο) τετράγωνο κουτί διακλάδωσης, που υπήρχε στο σημείο, η συσκευή τοποθετήθηκε στο εσωτερικό του και συνδέθηκε με τα απαραίτητα καλώδια που ήταν ήδη διαθέσιμα. Η όλη διαδικασία δεν απαιτήσε καμία ιδιαίτερη τροποποίηση στην υπάρχουσα ηλεκτρολογική διάταξη, καθιστώντας την εγκατάσταση της συσκευής μια από τις πιο γρήγορες και απροβλημάτιστες.

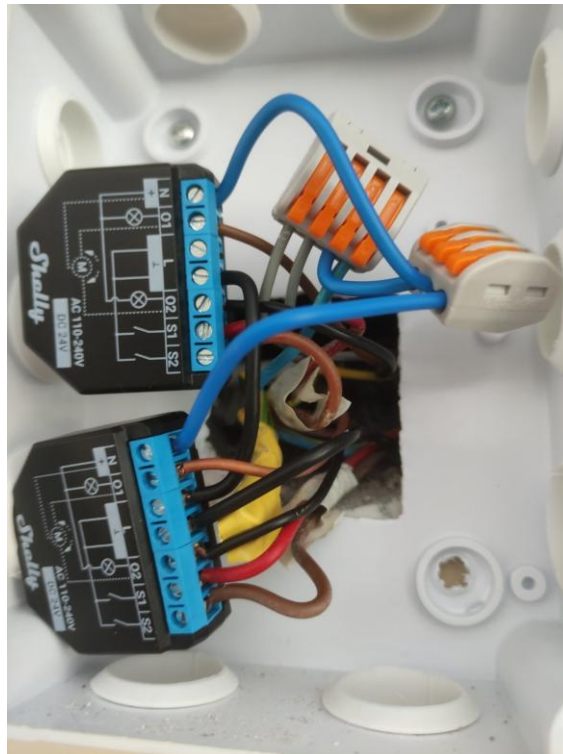
3.2.2 Εγκατάσταση δυτικών σκιάστρων

Αντίθετα με το ανατολικό, η εγκατάσταση των δύο Shelly Plus 2PM στα δυτικά σκιάστρα αποδείχτηκε πιο περίπλοκη, παρουσιάζοντας μάλιστα και αντίθετο πρόβλημα από εκείνο που συναντάται συνήθως σε παρόμοιες εγκαταστάσεις. Συγκεκριμένα, ενώ στις περισσότερες εγκαταστάσεις συνηθίζεται να απουσιάζει ο ουδέτερος αγωγός, στην προκείμενη απουσίαζε ο αγωγός της φάσης. Στο σημείο της διακλάδωσης υπήρχαν μόνο οι επιστροφές από τους διακόπτες, δύο ζεύγη φάσεων και ουδετέρων προς και από τους μηχανισμούς των σκιάστρων και ο ένας γενικός ουδέτερος «επιστροφής», καθιστώντας αδύνατη τη σύνδεση τους.

Η λύση δόθηκε με τη χρήση ατσαλίνας, μέσω της οποίας πραγματοποιήθηκε η όδευση πρόσθετου καλωδίου φάσης από το σημείο των διακοπών -όπου και τερματίζε-
έως το κουτί διακλάδωσης, σύμφωνα με τις απαιτήσεις της εγκατάστασης (βλ. Εικόνα

9). Αφού η διαδικασία ολοκληρώθηκε με ασφάλεια και επέτρεπε πλέον τη σωστή λειτουργία των συσκευών, το επόμενο βήμα ήταν η εγκατάστασή τους.

Λόγω περιορισμένου χώρου στο υπάρχον (μικρό) στρογγυλό κουτί διακλάδωσης και δεδομένου ότι έπρεπε να τοποθετηθούν δύο ανεξάρτητες συσκευές, κρίθηκε αναγκαίο να τοποθετηθεί επιπλέον ένα επιτοίχιο ηλεκτρολογικό κουτί, ακριβώς πάνω από το εντοιχισμένο. Εξασφαλίζοντας επαρκή χώρο και διασφαλίζοντας την σωστή, εύκολη και ασφαλή εγκατάστασή τους.



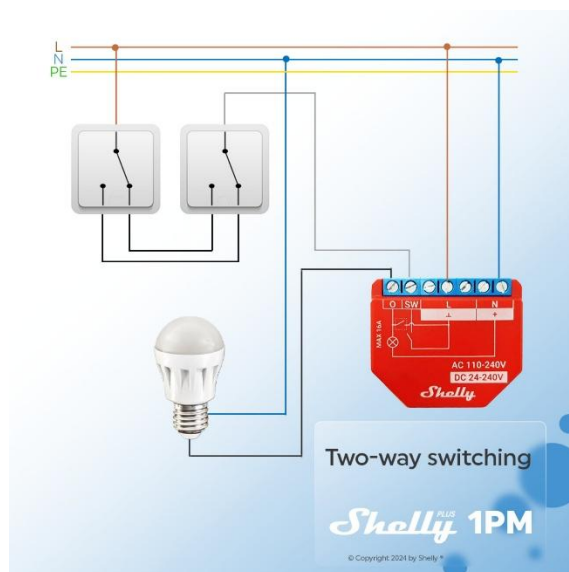
Εικόνα 10: Τα Shelly Plus 2PM των δυτικών σκιάστρων.



Εικόνα 11: Η τοποθέτηση των Shelly Plus 2PM των δυτικών σκιάστρων ψηλά στον τοίχο.

3.2.3 Εγκατάσταση σε φωτισμό

Η εγκατάσταση του Shelly Plus 1PM, το οποίο προοριζόταν για τον έλεγχο του φωτισμού, αναμενόταν να εμφανίσει ιδιαιτερότητες λόγω της υφιστάμενης καλωδίωσης τύπου «αλέ ρετούρ». Για την ορθή τοποθέτηση έγινε μια απόπειρα αναγνώρισής της, μελετώντας τα κουτιά διακλάδωσης πάνω ακριβώς από τον κάθε διακόπτη (όπως και με τα Shelly Plus 2PM). Μετά τον έλεγχο, διαπιστώθηκε πως ένα από τα δύο κουτιά διακλάδωσης περιείχε όλα τα απαιτούμενα καλώδια. Τη φάση πριν τον πρώτο σε σειρά διακόπτη, τον ουδέτερο και την επιστροφή από τον τελευταίο σε σειρά διακόπτη «αλέ ρετούρ». Έτσι, με μοναδικό εμπόδιο τον περιορισμένο χώρο εσωτερικά του κουτιού διακλάδωσης, τοποθετήθηκε ένα επιτοίχιο αντίστοιχο ηλεκτρολογικό κουτί, πάνω από το υφιστάμενο και η εγκατάσταση μπόρεσε και ολοκληρώθηκε με επιτυχία.



Εικόνα 12: Η συνδεσμολογία του Shelly Plus 1PM[16].

3.3 Σύνδεση Shelly συσκευών στο WiFi και βασική ρύθμιση

Πριν από οποιαδήποτε ενσωμάτωση στο Home Assistant, απαιτήθηκε η αρχική σύνδεση όλων των συσκευών στο WiFi και συγκεκριμένα στο ειδικό IoT WiFi δίκτυο του Τμήματος Ψηφιακών Συστημάτων. Αυτό πραγματοποιήθηκε μέσω του προσωπικού access point (AP) που διαθέτει κάθε συσκευή κατά την πρώτη ενεργοποίησή της. Με τη σύνδεση στο AP, ο χρήστης αποκτά πρόσβαση σε ένα πλήρως τοπικό self-hosted Shelly web interface, χωρίς την ανάγκη για σύνδεση ή εγγραφή σε κάποια cloud υπηρεσία. Πιο συγκεκριμένα, η διαδικασία σύνδεσης, για κάθε μία συσκευή, περιλαμβάνει τα εξής βήματα:

1. Σύνδεση στο WiFi που εκπέμπει η συσκευή (π.χ. ShellyPlus2PM-xxxxx).
2. Πλοήγηση στη default διεύθυνση ip 192.168.33.1, μέσω κάποιου browser, η οποία οδηγεί στο τοπικό self-hosted dashboard της συσκευής.
3. Εντοπισμός και επιλογή των ρυθμίσεων δικτύου της συσκευής.
4. Εισαγωγή των απαιτούμενων στοιχείων για τη σύνδεση στο WiFi με στατική IP, ώστε να εξασφαλίζεται η σταθερή σύνδεση στο δίκτυο και η αξιόπιστη πρόσβαση στο Home Assistant, και
5. Αποθήκευσή τους.

Με την ολοκλήρωση της διαδικασίας, η συσκευή είναι πλέον προσβάσιμη μέσω της στατικής και τοπικής IP διεύθυνσης που της δηλώσαμε[17].

Παράλληλα, η σύνδεσή τους έγινε και δοκιμαστικά μέσω της εφαρμογής Shelly Cloud App, με την οποία εντοπίστηκαν πολύ εύκολα οι συσκευές μέσω Bluetooth. Ωστόσο αυτή η προσέγγιση απαιτεί δημιουργία λογαριασμού στο cloud της Shelly, κάτι που έρχεται σε αντίθεση με τον στόχο της παρούσας εργασίας (για πλήρη τοπική λειτουργία χωρίς εξωτερικές cloud εξαρτήσεις), οπότε απορρίφθηκε σαν μέθοδος.

Ακόμα, για τη χρονικά συντονισμένη λειτουργία τους, δηλώθηκε στις ρυθμίσεις τους ο «ntp.uth.gr» SNTP server, σύμφωνα με οδηγίες της ομάδας τεχνικής υποστήριξης του τμήματος, ο οποίος αποτελεί επίσημο εξυπηρετητή χρόνου του Πανεπιστημίου Θεσσαλίας. Ενώ επίσης απαραίτητο ήταν, για λόγους ασφαλείας και ελέγχου πρόσβασης, να δημιουργηθούν credentials εισόδου για κάθε συσκευή, ώστε να περιορίζεται η πρόσβαση μόνο σε εξουσιοδοτημένους χρήστες.

Τέλος, έπρεπε να περαστούν οι κατάλληλες ρυθμίσεις λειτουργίας ανά τύπο συσκευής, όπως περιγράφεται στις παρακάτω υποενότητες.

3.3.1 Shelly Motion 2

Προκειμένου να επιτευχθεί η σωστή ενσωμάτωση του Shelly Motion 2 στο Home Assistant, ήταν απαραίτητο να ρυθμιστεί το πρωτόκολλο COIOT. Συγκεκριμένα, στο αντίστοιχο πεδίο των ρυθμίσεων δικτύου της συσκευής, εκχωρήθηκε η IP διεύθυνση του Home Assistant, συνοδευόμενη από την πύλη (port) 5683, η οποία αποτελεί την προκαθορισμένη πόρτα επικοινωνίας για το συγκεκριμένο πρωτόκολλο (CoAP/COIOT).

Η ρύθμιση αυτή είναι απαραίτητη ώστε το Shelly Integration του Home Assistant να μπορεί να εντοπίσει και να αναγνωρίσει τη συσκευή στο τοπικό δίκτυο. Αν και υπάρχει η εναλλακτική δυνατότητα ενσωμάτωσης μέσω MQTT, το COIOT θεωρείται πιο άμεσο και απλοποιεί τη διαδικασία σύνδεσης, σύμφωνα και με την επίσημη τεκμηρίωση του Integration[15].

3.3.2 Shelly Plus 2PM

Στην παρούσα εγκατάσταση, τα Shelly Plus 2PM ρυθμίστηκαν να λειτουργούν σε «Cover mode». Η επιλογή του συγκεκριμένου τρόπου λειτουργίας επιτρέπει στο Home Assistant, όσο και στο ίδιο το UI των Shelly, να αναγνωρίζουν τις συσκευές ως ελεγκτές μηχανισμού σκίαστρων, εμφανίζοντας έτσι τα κατάλληλα κουπιά ελέγχου (άνοιγμα, κλείσιμο, παύση) και την αντίστοιχη κατάσταση (state) λειτουργίας.

Για τον φυσικό χειρισμό των σκιάστρων, συνεχίζουν και είναι διαθέσιμοι οι διπλοί διακόπτες τύπου μπουτόν, που υπήρχαν ήδη, ρυθμισμένοι σε switch mode, ενώ ακόμη ενεργοποιήθηκε η λειτουργία για factory reset μέσω αυτών, ώστε να υπάρχει εναλλακτική μέθοδος επαναφοράς σε περίπτωση δυσλειτουργίας ή ανάγκης επαναρύθμισης των συσκευών. Η ρύθμιση αυτή είναι ιδιαίτερα χρήσιμη, καθώς η βασική μέθοδος επαναφοράς απαιτεί φυσική πρόσβαση στο μικρό κουμπί που βρίσκεται στο πίσω μέρος του κάθε Shelly Plus.



Εικόνα 13: Το κουμπί reset στο πίσω μέρος ενός Shelly plus 2PM.

Τέλος, κατά τις δοκιμές της εγκατάστασης, παρατηρήθηκαν κάποιες αναντιστοιχίες μεταξύ των ενδείξεων και εντολών του web interface και της λειτουργίας των φυσικών διακοπών, ως προς τη φορά κίνησης των σκιάστρων. Για παράδειγμα, η επιλογή «πάνω» στο UI μπορεί να ενεργοποιούσε την κάθοδο του σκιάστρου, ενώ το αντίστοιχο κουμπί στον τοίχο λειτουργούσε σωστά. Για την αντιμετώπιση αυτών των διαφορών, αξιοποιήθηκαν δύο σημαντικές ρυθμίσεις του Shelly Plus 2PM. Οι:

- *Reverse directions*, η οποία καθορίζει την ερμηνεία του Shelly για την κατεύθυνση λειτουργίας του μοτέρ (πάνω/κάτω), και
- *Swap switch inputs*, η οποία ελέγχει ποιο ρελέ ενεργοποιείται από κάθε κουμπί του φυσικού διακόπτη.

Με τον κατάλληλο συνδυασμό των παραπάνω ρυθμίσεων, επιτεύχθηκε πλήρης αντιστοίχιση μεταξύ της λειτουργίας μοτέρ, των ενδείξεων στο web interface των Shelly και της φυσικής λειτουργίας των διακοπών, εξασφαλίζοντας συνεπή και ορθή συμπεριφορά του συστήματος.

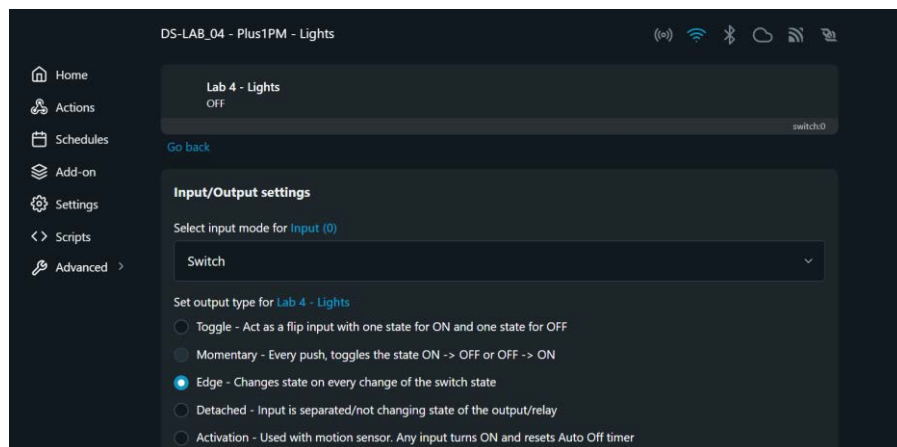
Ακολουθεί πίνακας που παρουσιάζει παραδείγματα συνδυασμών των ρυθμίσεων Reverse directions και Swap switch inputs στο Shelly Plus 2PM, όπως και το αποτέλεσμα που προκύπτει στη λειτουργία και έλεγχο των σκιάστρων.

Πίνακας 1: Συνδυασμοί ρυθμίσεων κινήσεων και αποτελέσματα λειτουργίας, Shelly Plus 2PM

Reverse directions	Swap switch inputs	Shelly outputs προς μοτέρ	Διακόπτες στον τοίχο προς Shelly	Αποτέλεσμα
OFF	OFF	Σωστά	Σωστά	Όλα σωστά
OFF	OFF	Αντίστροφα	Σωστά	Λάθος στο UI
ON	OFF	Σωστά	Αντίστροφα	Λάθος στους διακόπτες
ON	ON	Σωστά	Σωστά	Όλα σωστά
OFF	ON	Αντίστροφα	Σωστά	UI και διακόπτες συμφωνούν, αλλά τα σκιάστρα κινούνται ανάποδα από τις εντολές τους

3.3.3 Shelly Plus 1PM

Στο Shelly Plus 1PM, δεν χρειάστηκε ιδιαίτερα να γίνουν εξατομικευμένες ρυθμίσεις. Η μόνη εξατομικευμένη ρύθμιση που χρειάστηκε ήταν να δηλωθεί το input, από τους φυσικούς διακόπτες, ως «Switch» και output type επιλογή «Edge». Με αυτόν τον τρόπο κάθε αλλαγή κατάστασης (πάτημα διακόπτη) προκαλεί εναλλαγή της κατάστασης του ρελέ στο Shelly, επιτρέποντας το άναμμα ή το σβήσιμο του φωτισμού ανεξαρτήτως της προηγούμενης κατάστασης.



Εικόνα 14: Ρυθμίσεις εισόδου για τον έλεγχο φωτισμού στο Shelly web interface.

Η επιλογή «Edge» είναι η μόνη που ανταποκρίνεται σωστά όταν είναι επιθυμητό να μπορεί να εναλλάσσεται ο φωτισμός και από τον φυσικό διακόπτη και από το Home Assistant, γιατί είναι η μόνη ρύθμιση η οποία δεν επηρεάζεται από τις φυσικές θέσεις του φυσικού διακόπτη και έτσι παραμένει συνεπής στη λειτουργία του. Ο παρακάτω πίνακας παρουσιάζει αναλυτικά τις διαθέσιμες επιλογές -ρύθμισης- εξόδου και εξηγεί γιατί δεν είναι κατάλληλες:

Πίνακας 2: Σύγκριση τύπων εξόδου (Output Type) στο Shelly Plus 1PM.

Επιλογή Output Type	Πώς λειτουργεί	Περιορισμοί σε φυσικό διακόπτη και παράλληλη χρήση web interface
Edge	Ανιχνεύει κάθε αλλαγή κατάστασης του διακόπτη και εναλλάσσει την κατάσταση του ρελέ.	Δεν εξαρτάται από τη θέση του διακόπτη και λειτουργεί σωστά ακόμη και όταν η κατάσταση του φωτισμού αλλάξει από το web ή κάποιον αυτοματισμό.
Toggle	Αντιστοιχεί μία θέση του διακόπτη στο ON και την άλλη στο OFF.	Αν η κατάσταση του φωτισμού αλλάξει από το web σε «on» και η θέση του διακόπτη είναι σε «off», το «πάτημα» του διακόπτη σε «on» δεν θα κλείσει τα φώτα γιατί είναι ήδη ανοιχτά. Για να κλείσουν, θα πρέπει να «ξαναπατηθεί» ο διακόπτης ώστε να βρεθεί στη θέση «off».
Detached	Το input δεν επηρεάζει καθόλου την έξοδο (relay).	Δεν επιτρέπει τον έλεγχο του φωτισμού από τον φυσικό διακόπτη, αλλά συνεχίζει και λαμβάνει το state της θέσης του. Αυτή η ρύθμιση είναι ιδιαίτερα ενδιαφέρουσα καθώς επιτρέπει την ανάπτυξη πιο εξειδικευμένων αυτοματισμών και πιο σύνθετα σενάρια ελέγχου.
Activation	Ενεργοποιεί το ρελέ και ξεκινά χρονοδιακόπτη για αυτόματο σβήσιμο.	Σχεδιασμένο για αισθητήρες κίνησης. Δεν προσφέρει χειροκίνητο έλεγχο μέσω διακόπτη.

3.4 Ενσωμάτωση στο Home Assistant

3.4.1 Shelly Motion 2

Σε αντίθεση με την απλότητα της φυσικής εγκατάστασης της συσκευής, η ενσωμάτωση της στο Home Assistant παρουσίασε ιδιαιτερότητες. Το έτοιμο integration, αν και φιλικό στη χρήση, φαίνεται να είναι προγραμματισμένο με τέτοιον τρόπο ώστε να καταγράφει τα δεδομένα μόνο όταν παρατηρούνται έντονες αλλαγές στις τιμές των μετρήσεων. Αυτό δημιούργησε περιορισμούς και έντονο προβληματισμό, καθώς για την αξιόπιστη υλοποίηση αυτοματισμών, κρίθηκε απαραίτητη η ακριβής δειγματοληψία σε επίπεδο λεπτού.

Η λύση δόθηκε με την χρήση του API της συσκευής και την ανάπτυξη ενός μικρού Python script. Το script, εκτελείται από το Home Assistant κάθε 1 λεπτό και είναι προγραμματισμένο να στέλνει μια κλήση GET σε συγκεκριμένο endpoint, σύμφωνα και με το documentation της εταιρίας[23]. Ως απάντηση, λαμβάνει ένα JSON με τις πιο πρόσφατες μετρήσεις. Με αυτόν τον τρόπο, μπορούμε να έχουμε ακριβείς τιμές στο HA κάθε περίπου 2 λεπτά.

Ο λόγος που η χρονική καθυστέρηση μπορεί να κυμανθεί από 1 έως 2 λεπτά, εξηγείται από τον τρόπο εκτέλεσης της όλης επικοινωνίας. Συγκεκριμένα:

1. Το Shelly Motion 2, ανανεώνει τις μετρήσεις του, εσωτερικά, κάθε 1 λεπτό.
2. Το Python script εκτελείται από το Home Assistant επίσης κάθε 1 λεπτό.
3. Αν το script «τύχει» και εκτελεστεί λίγο πριν την ανανέωση της συσκευής, θα παραλάβει τα δεδομένα του προηγούμενου λεπτού και η νέα τιμή θα καταγραφεί στον επόμενο κύκλο.

Συνεπώς, το κατώτερο χρονικό διάστημα ενημέρωσης είναι το 1 λεπτό, ενώ το ανώτερο μπορεί να φτάσει τα 2 λεπτά.

Παρακάτω αναλύεται τμηματικά ο κώδικας του `get_lab04_lux.py` Python script συνδυαστικά με τον `yaml` κώδικα που χρειάζεται για να δημιουργήσει sensor entities για τις μετρήσεις:

Ανάγνωση διαπιστευτηρίων από αρχείο `secrets.yaml`:

```
# Load credentials from YAML file
with open("secrets.yaml", "r") as file:
    secrets = yaml.safe_load(file)

username = secrets["DSlab04_username"]
password = secrets["DSlab_password"]
```

Ορισμός του endpoint και αποστολή αιτήματος GET

```
url = "http://10.120.0.40/status"
response = requests.get(url, auth=HTTPBasicAuth(username, password))
```

Ανάγνωση και φιλτράρισμα των δεδομένων JSON

```
data = response.json()
filtered = {
    "lux": data["lux"]["value"],
    "illumination": data["lux"]["illumination"],
    "temp": data["tmp"]["value"],
    "unixtime": data["unixtime"],
    "time": data["time"]
}
```

Εκτύπωση των φιλτραρισμένων δεδομένων

```
print(json.dumps(filtered, indent=1))
```

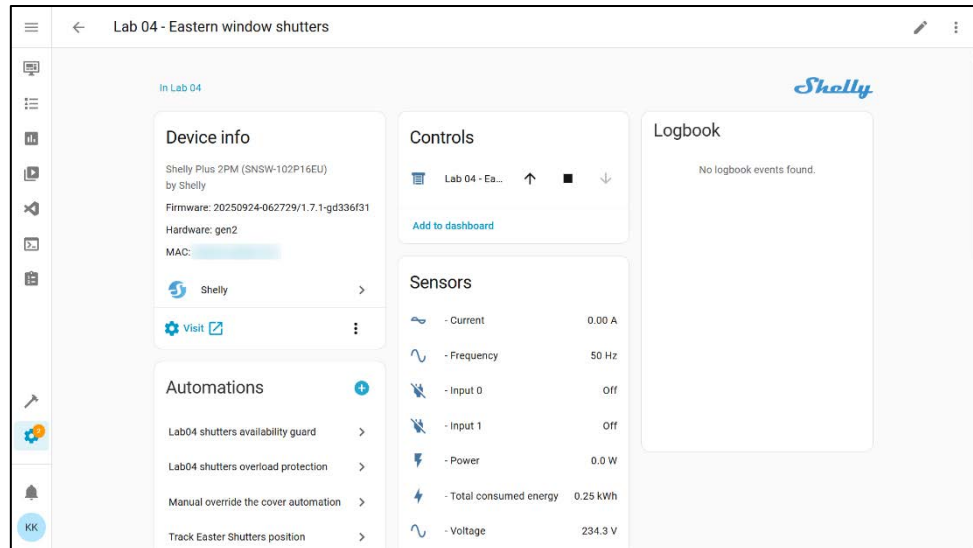
Δημιουργία αισθητήρα τύπου command_line στο Home Assistant

```
- sensor:
  name: Lab04 luminosity
  unique_id: xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx # (UUID - Hidden)
  command: "python3 /config/get_lab04_lux.py"
  device_class: ILLUMINANCE
  state_class: measurement
  unit_of_measurement: "lx"
  scan_interval: 60 # Fetch every 1 minutes
  value_template: "{{ value_json.lux | int }}"
  json_attributes:
    - lux
    - illumination
    - temp
    - time
    - unixtime
```

3.4.2 Shelly Plus 1PM & 2PM

Η ενσωμάτωση των Shelly Plus 1PM & 2PM στο Home Assistant, αποδείχτηκε ιδιαίτερα ομαλή, καθώς τα συγκεκριμένα μοντέλα υποστηρίζονται πλήρως από το επίσημο integration του Home Assistant. Μετά την αρχική ρύθμιση δικτύου και την απόδοση στατικών IP, οι συσκευές αναγνωρίστηκαν αυτόματα από την πλατφόρμα και καταγράφηκαν ως οντότητες. Κάθε οντότητα αντιπροσωπεύει τόσο τα ρελέ ελέγχου όσο και τις λειτουργίες μέτρησης κατανάλωσης. Ενώ επιπλέον, κάθε συσκευή συνοδεύεται από οντότητες που παρέχουν χρήσιμες πληροφορίες για την κατάσταση και τη λειτουργία της, όπως η τρέχουσα τάση του συστήματος, η ένταση του δικτύου

WiFi (σε dBm), η ημερομηνία τελευταίας ενημέρωσης firmware, η εσωτερική θερμοκρασία της συσκευής, κ.ά.. Οι οντότητες αυτές επιτρέπουν όχι μόνο την αξιόπιστη παρακολούθηση της λειτουργίας τους αλλά και τον προγραμματισμό πιο περίπλοκων αυτοματισμών.



Εικόνα 15: Διεπαφή ελέγχου και παρακολούθησης των ανατολικών ρολών μέσω Home Assistant.

3.4.3 Μετεωρολογικός σταθμός GreenDigital

Για την ενσωμάτωση των μετρήσεων του μετεωρολογικού σταθμού GreenDigital και τη δημιουργία αισθητήρων (sensor) με βάση αυτών, υλοποιήθηκε ένας μηχανισμός βασισμένος σε δύο αισθητήρες τύπου *command_line* [40], οι οποίοι εκτελούν περιοδικά ένα κοινό Python script με ονομασία *data_fetcher_v3.py*.

Το script αυτό έχει ως κύρια λειτουργία του την ανάκτηση δεδομένων από ένα εξωτερικό αρχείο δεδομένων τύπου CSV από την online διεύθυνση <https://greendigital.uth.gr/data/meteo48h.csv.txt>, την επεξεργασία των δεδομένων αυτών και την έξοδό τους σε μορφή JSON. Έτσι, οι αισθητήρες, μέσω του *value_template* και των *json_attributes*, διαβάζουν τις επιθυμητές τιμές από το JSON και τις καθιστούν διαθέσιμες στο οικοσύστημα του Home Assistant.

Παρακάτω παρουσιάζεται τμηματικά ο κώδικας του *data_fetcher_v3.py* Python script:

Εισαγωγή βιβλιοθηκών και ρύθμιση καταγραφών (logging)

```
import pandas as pd
import json
import sys
```

```

import os
import logging
from logging import getLogger
import datetime

# Configure default logging with timestamp
logger = getLogger(__name__)
logging.basicConfig(
    level=logging.INFO,
    format="%(asctime)s [%(levelname)s] [ha_meteo_uth] %(name)s: %(message)s",
    datefmt="%Y-%m-%d %H:%M:%S"
)

```

Ορισμός αρχείου κατάστασης

```

# Text file name which stores the last fetched unix time and status
LAST_STATUS = "/config/ha_meteo_uth/latest_status.json"

```

Το αρχείο *latest_status.json* χρησιμοποιείται για την αποθήκευση της τελευταίας χρονικής στιγμής (Unix time) κατά την οποία έγινε επιτυχής ανάκτηση δεδομένων, καθώς και ενός μετρητή επαναλήψεων (*match_count*).

Ανάγνωση προηγούμενης κατάστασης από αρχείο

```

def read_status():
    # Reading the status from the JSON file
    try:
        with open(LAST_STATUS, "r") as f:
            return json.load(f)
    except Exception as e:
        logger.error(f"The last fetched time is invalid or missing ({e}),")
        logger.info("Setting default value...")
        return {"last_time_fetched": 0, "match_count": -1} # Default values if file
doesn't exist

```

Η συνάρτηση προσπαθεί να διαβάσει το αρχείο *latest_status.json*. Αν αυτό δεν υπάρχει ή είναι κατεστραμμένο, επιστρέφονται προκαθορισμένες τιμές (*last_time_fetched* = 0, *match_count* = -1), ώστε να συνεχιστεί η εκτέλεση χωρίς σφάλμα.

Μέθοδος αποθήκευσης κατάστασης στο αρχείο

```

def write_status(status):
    logger.info(f"Saving last fetched status to {LAST_STATUS}")
    # Write the status to the JSON file
    with open(LAST_STATUS, "w") as f:
        json.dump(status, f, indent=2)

```

Αποθηκεύει την ενημερωμένη κατάσταση στο αρχείο *latest_status.json*, ώστε να είναι διαθέσιμη στην επόμενη εκτέλεση του script.

Συνάρτηση εκκίνησης `main_job()`

Η συνάρτηση `main_job()` αποτελεί το σημείο εκκίνησης του script.

```
def main_job():  
    # Reading the status from the JSON file  
    status = read_status()  
  
    # Fetching the data  
    data_fetch(status)  
  
main_job()
```

Ανάκτηση και επεξεργασία δεδομένων

Το πρώτο στάδιο της συνάρτησης `data_fetch()` ανακτώνται οι τιμές `last_time_fetched` και `match_count` από το αντικείμενο `status`, το οποίο έχει προηγουμένως φορτωθεί από το αρχείο `latest_status.json` κατά την εκκίνηση του script. Η τιμή `last_fetch` αντιπροσωπεύει τη χρονική στιγμή (timestamp) της τελευταίας επιτυχούς ανάκτησης δεδομένων, ενώ ο `match_count` καταγράφει πόσες φορές έχει επαναληφθεί μέτρηση με το ίδιο timestamp. Χρησιμοποιώντας αυτόν τον counter το script ελέγχει πόσες φορές έχει δεχτεί την ίδια μέτρηση ώστε να δηλώσει πως κάτι πάει λάθος και να μην εξάγει JSON δεδομένα, έτσι ώστε να προστατεύσει το ιστορικό δεδομένων-μετρήσεων του ΗΑ.

```
# Get the last fetched time  
last_fetch = int(float(status["last_time_fetched"]))  
match_count = status["match_count"]
```

Το αρχείο CSV ανακτάται από την καθορισμένη διεύθυνση URL, η οποία φιλοξενεί τις μετρήσεις του μετεωρολογικού σταθμού GreenDigital, και «διαβάζεται» μέσω της βιβλιοθήκης `pandas`, η οποία εγκαταστάθηκε μέσω του terminal Add-on στον φάκελο `config/deps`.

```
CSV_URL = "https://greendigital.uth.gr/data/meteo48h.csv.txt"  
csv_data=pd.read_csv(CSV_URL)
```

Έπειτα, ορίζονται σε πίνακα οι στήλες που περιέχουν τις μετρήσεις ενδιαφέροντος και είναι επιθυμητό να διαβαστούν αργότερα από τη μεταβλητή `csv_data` με τα δεδομένα από το CSV. Η επιλογή περιορίζεται σε συγκεκριμένα πεδία ώστε να μειωθεί το υπολογιστικό φορτίο και να διασφαλιστεί η συνέπεια των δεδομένων.

```
# Selecting the relevant columns from the pulling csv file  
columns = ['Unix_time', 'Temp_Out', 'In_Temp', 'Low_Temp', 'Hi_Temp', 'Out_Hum',  
'In_Hum', 'Dew_Pt', 'In_Dew', 'Solar_Rad']  
values = csv_data[columns]
```

Τέλος, όλες οι τιμές μετατρέπονται σε αριθμητικές, με τις τυχόν μη έγκυρες τιμές (π.χ. κενά, συμβολικά ή αλφαριθμητικά) να μετατρέπονται σε NaN, ώστε να μπορούν να εντοπιστούν και να αντιμετωπιστούν ανάλογα σε επόμενο στάδιο.

```
# Converting non-numeric values to NaN
values = values.apply(pd.to_numeric, errors='coerce')
```

Έλεγχος και διόρθωση μη έγκυρων τιμών

Αφού έχουν επιλεγεί και μετατραπεί οι στήλες του αρχείου CSV σε αριθμητικές τιμές, το script προχωρά στον έλεγχο για την ύπαρξη μη έγκυρων ή ελλιπών δεδομένων (NaN). Η συνθήκη `values.isna().values.any()` ελέγχει αν υπάρχουν κενές ή μη αριθμητικές τιμές σε οποιοδήποτε κελί του πίνακα. Αν εντοπιστούν τέτοιες τιμές, αυτές αντικαθίστανται με 0 μέσω της μεθόδου `fillna(0)`.

```
# Checking for NaN values and replace them with 0
if values.isna().values.any():
    values = values.fillna(0) # Replacing NaNs with 0
    logger.error(" NaN or broken values found and replaced with 0!")
```

Ανάκτηση τελευταίας γραμμής CSV

Αφού έχουν καθαριστεί τα δεδομένα και αντικατασταθεί τυχόν μη έγκυρες τιμές, το script προχωρά στην ανάκτηση της πιο πρόσφατης μέτρησης

```
# Getting the lasts rows
last_row = values.iloc[-1]
```

και στη συνέχεια, το timestamp της μέτρησης *Unix_time* μετατρέπεται σε αναγνώσιμη μορφή

```
# Convert unix to readable format
date = datetime.datetime.utcfromtimestamp(last_row["Unix_time"]).strftime('%d %B %Y')
time =
datetime.datetime.utcfromtimestamp(last_row["Unix_time"]).strftime('%H:%M:%S')
```

Δημιουργία μεταβλητής data

Εφόσον έχει ανακτηθεί η τελευταία γραμμή δεδομένων και έχουν υπολογιστεί οι αντίστοιχες χρονικές πληροφορίες *Date* και *Time*, το script προχωρά στη δημιουργία αυτοσχέδιας «τελευταίας γραμμής» με την προσθήκη αυτών των δεδομένων σε μια μεταβλητή *data*.

```
data = {**{col: last_row[col] for col in columns}, "Date": date, "Time": time}
```

Έλεγχος επαναλαμβανόμενων μετρήσεων

Κρίσιμο στοιχείο της λειτουργίας του script αποτελεί η δυνατότητα να εντοπίζει αν η τελευταία μέτρηση που ανακτήθηκε από το αρχείο CSV είναι νέα ή επαναλαμβανόμενη. Αυτό γίνεται με σύγκριση της χρονικής σήμανσης *Unix_time* της τελευταίας γραμμής με την τιμή *last_fetch* που έχει αποθηκευτεί από την προηγούμενη εκτέλεση, όπου ελέγχει αν έχει διαβάσει την ίδια μέτρηση πάνω από τρεις φορές (ξεκινάει να μετράει από -1 και ελέγχει για ≥ 2). Σε κάθε εκτέλεση, ενημερώνει κατάλληλα και το αντικείμενο status με τις νέες τιμές

```
if last_row["Unix_time"] == last_fetch:
    logger.warning(f"The last known data is fetched {match_count + 1} times already (Unix Time: {last_fetch})!")
    if match_count >= 2:
        logger.error("The returned data have been set to 'unavailable'.\nCheck if the source data is updated.")
        data = {**{col: "unavailable" for col in columns}, "Date": "unavailable", "Time": "unavailable"}
    else:
        # Increment of the match_count
        status["match_count"] += 1

if last_row["Unix_time"] != last_fetch:
    # If no match, reset the match count
    status["match_count"] = -1
    # Updating the last fetched time
    status["last_time_fetched"] = last_row["Unix_time"]
```

Αποθήκευση κατάστασης και έξοδος JSON

Φτάνοντας στο τέλος του script, αφού έχει ολοκληρωθεί και προσπεραστεί ο έλεγχος για επαναλαμβανόμενες μετρήσεις, το script προχωρά στην αποθήκευση της κατάστασης και στην έξοδο των δεδομένων.

```
# Writing the updated status to the file
write_status(status)

json_output = json.dumps(data, indent=12)
print(json_output)
```

Με αυτό ολοκληρώνεται η πλήρης εκτέλεση του Python *data_fetcher_v3.py* script και τα δεδομένα, ως JSON πλέον, είναι έτοιμα να χρησιμοποιηθούν στη δημιουργία των δύο αισθητήρων.

Όσο αφορά λοιπόν τους αισθητήρες, ο ένας αισθητήρας αφορά την θερμοκρασία εξωτερικού χώρου και χρησιμοποιεί την τιμή *Temp_Out* ως την κύρια τιμή-μέτρηση του sensor μέσω του *value_template*. Παράλληλα, στο πεδίο *json_attributes*, εισάγονται και

άλλες τιμές από το JSON που επιστρέφει που επιστρέφει το Python script όπως *In_Temp*, *Solar_Rad*, *Date*, *Time*, οι οποίες δεν εμφανίζονται απευθείας στον αισθητήρα αλλά μπορούν να αξιοποιηθούν για τη δημιουργία επιμέρους αισθητήρων μέσω *template sensors* ή *helpers*. Ο δεύτερος αισθητήρας αφορά αποκλειστικά την υγρασία εξωτερικού χώρου, με κύρια τιμή την *Out_Hum*, ενώ η *In_Hum* εισάγεται αντίστοιχα ως attribute για περαιτέρω χρήση από το HA.

```
- sensor:
  name: Temperature Outside
  unique_id: xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx # (UUID - Hidden)
  command: "python3 /config/ha_meteo_uth/data_fetcher_v3.py"
  device_class: temperature
  state_class: measurement
  unit_of_measurement: "°C"
  scan_interval: 600 # Fetch every 10 minutes
  value_template: "{{ value_json.Temp_Out }}"
  json_attributes:
    - Temp_Out
    - In_Temp
    - In_Dew
    - Low_Temp
    - Hi_Temp
    - Dew_Pt
    - Solar_Rad
    - Date
    - Time
    - Unix_time

- sensor:
  name: Humidity Outside
  unique_id: xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx # (UUID - Hidden)
  command: "python3 /config/ha_meteo_uth/data_fetcher_v3.py"
  device_class: humidity
  state_class: measurement
  unit_of_measurement: "%"
  scan_interval: 600 # Fetch every 10 minutes
  value_template: "{{ value_json.Out_Hum | int }}"
  json_attributes:
    - Out_Hum
    - In_Hum
```

Η επιλογή να χρησιμοποιείται ένας αισθητήρας για πολλές τιμές μέσω των *json_attributes* έγινε σκόπιμα, ώστε να αποφευχθεί η δημιουργία πολλών *command line* αισθητήρων, οι οποίοι θα εκτελούσαν το ίδιο script ανεξάρτητα. Μια τέτοια προσέγγιση θα οδηγούσε σε υπερβολικό αριθμό αιτημάτων προς τον εξωτερικό διακομιστή που φιλοξενεί το CSV, με αποτέλεσμα πιθανή υπερφόρτωση ή ακόμα και περιορισμό πρόσβασης. Αντίθετα, η τρέχουσα υλοποίηση επιτρέπει την εκτέλεση του script δύο φορές ανά κύκλο ενημέρωσης, με τα δεδομένα να διαμοιράζονται εσωτερικά στο Home Assistant.

4 Αυτοματισμοί και ρουτίνες

4.1 Σκίαστρα

Η λειτουργία των σκιάστρων στο εργαστήριο βασίζεται σε ένα σύστημα εκτίμησης και ελέγχου θέσης, το οποίο περιλαμβάνει έναν αυτοματισμό εκτίμησης θέσης για κάθε σκίαστρο, ένα script ευθυγράμμισης των εκτιμώμενων θέσεων μεταξύ τους και τον κεντρικό αυτοματισμό λειτουργίας τους, που είναι υπεύθυνος να ρυθμίζει την κατάσταση-θέση τους, βάσει περιβαλλοντικών συνθηκών.

4.1.1 Αυτοματισμός εκτίμησης θέσης

Ο αυτοματισμός εκτίμησης θέσης των σκιάστρων, έχει σχεδιαστεί με βασική του λειτουργία -όπως λέει ο τίτλος- την εκτίμηση των θέσεων των σκιάστρων. Γίνεται χρήση του όρου «εκτίμηση» διότι, το υλικό του μηχανισμού δεν επιτρέπει το πλήρες καλιμπράρισμα το οποίο γίνεται κανονικά μέσω των Shelly[24], εξαιτίας της απουσίας κατάλληλων τελικών διακοπών. Συγκεκριμένα, για να πραγματοποιηθεί το καλιμπράρισμα μέσω των Shelly, χρήσιμη θα ήταν η ύπαρξη φυσικών διακοπών (ή κάποιο αντίστοιχο μηχανικό ή ψηφιακό σύστημα) στα άκρα της διαδρομής – δηλαδή στο κάθε τέρμα ανοίγματος και κλεισίματος του μηχανισμού. Οι διακόπτες αυτοί, όταν «πατηθούν», μηδενίζουν την κατανάλωση σε watt του μηχανισμού, με αποτέλεσμα το Shelly να μπορεί να εντοπίσει τις ακραίες θέσεις λειτουργίας, καθώς βασίζεται στην ανίχνευση απότομων μεταβολών στην κατανάλωση για τον προσδιορισμό τους[25] και τελικά, για την ολοκλήρωση του καλιμπραρίσματος.



Εικόνα 16: Ένας από τους μηχανισμούς σκίασης, με επισήμανση του σημείου-κουμπιού μηδενισμού ηλεκτρικής κατανάλωσης.

Στην παρούσα εγκατάσταση υπάρχει μόνο ένας τέτοιος φυσικός διακόπτης στο τέρμα του κλεισίματος. Κατά συνέπεια, δεν μπορεί να γίνει υπολογισμός με ακρίβεια της συνολικής διαδρομής του μηχανισμού. Για την αντιμετώπιση του περιορισμού αυτού, αναπτύχθηκε ο συγκεκριμένος αυτοματισμός, ο οποίος εκτιμά τη θέση του σκιάστρου βάσει του χρόνου λειτουργίας του και υλοποιείται ξεχωριστά για κάθε σκίαστρο, καθώς κάθε σκίαστρο είναι μια αυτοτελή οντότητα.

Ο αυτοματισμός λειτουργεί ως εξής:

Κάθε φορά που ενεργοποιείται ο μηχανισμός ενός σκιάστρου, ο αυτοματισμός ενεργοποιείται και ξεκινάει να καταγράφει το είδος της εντολής (άνοιγμα ή κλείσιμο) και τη διάρκεια της. Στην περίπτωση ανοίγματος, ο αυτοματισμός αυξάνει κατά 1 βήμα το *input_number* entity helper (βοηθητικές οντότητες του Home Assistant) [26] το οποίο διατηρεί την εκτιμώμενη τιμή θέσης στην κλίμακα 0 (τελείως κλειστό) με 10 (τελείως ανοιχτό) και με βήμα (step) ίσο με 2 μονάδες, κάθε 2 δευτερόλεπτα. Η διαδικασία αυτή επαναλαμβάνεται όσο το σκίαστρο παραμένει σε κατάσταση «ανοίγματος». Αντίστοιχα, υ-

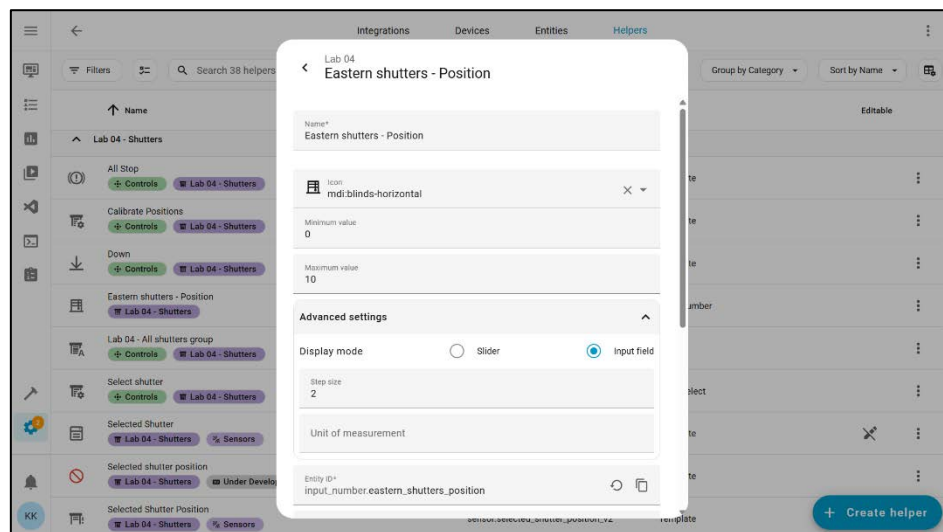
πολογίζεται και η θέση του κατά το κλείσιμο, όπου παράλληλα ελέγχει και την κατανάλωση ισχύος του κινητήρα, καθώς ο κινητήρας διαθέτει το φυσικό κουμπί που διακόπτει την παροχή ρεύματος όταν το σκίαστρο φτάσει στην κατώτατη θέση (βλ. εικόνα 16). Συνεπώς, η στιγμή που η ισχύς της κατανάλωσης γίνεται ίση με 0 Watt (ή μικρότερη από 0.10 όπως διατυπώνεται στον αυτοματισμό), αποτελεί αξιόπιστο δείκτη ότι το σκίαστρο έχει πλήρως κλείσει και βρίσκεται στη θέση «0».

Η παραπάνω λογική λειτουργίας παρουσιάζεται συνοπτικά στον ακόλουθο ψευδοκώδικα ενώ οι πλήρες κώδικες YAML, με ονόματα αυτοματισμών “Track Easter Shutters position”, “Track western shutters 1 position” και “Track western shutters 2 position” (ένας αυτοματισμός για κάθε σκίαστρο).

```

Αν σκίαστρο = "opening" και Διάρκεια_κατάστασης > 1 τότε
  Όσο σκίαστρο = "opening" επανάλαβε κάθε 2 δευτερόλεπτα:
    Θέση = Θέση + Βήμα_θέσης
Τέλος_Αν

Αν σκίαστρο = "closing" και Διάρκεια_κατάστασης > 1 τότε
  Όσο σκίαστρο = "closing" επανάλαβε κάθε 2 δευτερόλεπτα:
    Θέση = Θέση - Βήμα_θέσης
  Αν Ισχύς < 0.1 W τότε
    Θέση = 0
Τέλος_Αν
  
```



Εικόνα 17: Οι ρυθμίσεις στο UI για τον input_number helper[26] του ανατολικού σκιάστρου.

4.1.2 Ρουτίνα/Script ευθυγράμμισης σκιάστρων

Σκοπός της ρουτίνας είναι η επικαιροποίηση μιας όσο πιο σίγουρης αλλά εκτιμώμενης πάντα θέσης για κάθε σκίαστρο, όπως αυτή έχει υπολογιστεί από τον αντίστοιχο αυτοματισμό εκτίμησης θέσης (βλ. ενότητα 4.1.1), και τελικά η ευθυγράμμιση τους με βάση την κάθε μία από αυτές.

Ως ρουτίνα/script, δεν εκτελείται αυτόματα, αλλά απαιτεί ρητή κλήση είτε από τον χρήστη, είτε από κάποιον αυτοματισμό, είτε από άλλη εκτελούμενη ρουτίνα. Για την πρώτη περίπτωση, έχει προστεθεί κουμπί εκτέλεσης στο dashboard του ΗΑ, επιτρέποντας άμεση και εύκολη ενεργοποίηση από τον χρήστη. Ενώ επιπλέον, η ρουτίνα καλείται από τον κεντρικό αυτοματισμό λειτουργίας των σκιάστρων (βλ. ενότητα 4.1.3).

Ιδιαίτερα σημαντικό είναι το script να εκτελείται μετά την ολοκλήρωση χειροκίνητων κινήσεων των σκιάστρων, είτε αυτές γίνονται μέσα από το ΗΑ είτε από τους φυσικούς διακόπτες στους τοίχους του εργαστηρίου. Αυτό εξασφαλίζει ότι:

- Τα σκιάστρα παραμένουν ορθά ευθυγραμμισμένα, αποφεύγοντας προβλήματα στην οπτική και μηχανική τους θέση.
- Οι καταγραφές παραμένουν συνεπείς, καθώς χειροκίνητες κινήσεις χωρίς επακόλουθη εκτέλεση της ρουτίνας, ενδέχεται να καταγραφούν ως stopped, ακόμη κι αν τα σκιάστρα είναι όντως πλήρως ανοιχτά ή κλειστά. Αυτό μπορεί να προκαλέσει δυσλειτουργίες σε αυτοματισμούς και ασυμβατότητες στην απεικόνιση των καταστάσεων στο dashboard, επηρεάζοντας τη συνολική αξιοπιστία του συστήματος.

Για το δεύτερο, ιδιαίτερη σημασία έχει η κατάσταση closed, καθώς είναι το μόνο σημείο στο οποίο μπορεί να πραγματοποιηθεί λογικό και πραγματικό καλιμπράρισμα. Στη θέση αυτή, όπως αναφέρθηκε και στην ενότητα 4.1.1, η κατανάλωση ισχύος του κινητήρα μηδενίζεται, επιτρέποντας ασφαλή και αξιόπιστη αναγνώριση της τελικής θέσης χωρίς τον κίνδυνο υπερφόρτωσης ή βλάβης του μοτέρ.

Αντίθετα, η κατάσταση κατά το άνοιγμα ορίζεται ως stopped από τη ρουτίνα για λόγους καλής λειτουργίας του μοτέρ, ώστε να μη συνεχιστεί η κίνηση πέρα του επιθυμητού χρονικού σημείου. Παρότι τα Shelly Plus 2PM είναι ρυθμισμένα με εσωτερικό χρονικό όριο ανοίγματος τα 10 δευτερόλεπτα και σταματάει αυτόματα, η εντολή stop προστίθεται για να διασφαλιστεί ότι η κίνηση τερματίζει ακριβώς όταν η εκτιμώμενη θέση φτάσει στο όριο. Αποφεύγοντας την υπερκατανάλωση και την πιθανότητα ζημίας του κινητήρα.

Η χρήση της stopped κατάστασης στο άνοιγμα δεν επηρεάζει αρνητικά την απεικόνιση στο dashboard του Home Assistant, καθώς η stopped κατάσταση σε επίπεδο shelly, συνδέεται απεικονίστηκε με την open σε επίπεδο HA. Έτσι η οπτική αναπαράσταση παραμένει σωστή.

Συνοπτική λογική λειτουργίας σε ψευδοκώδικα:

```
Ανάκτησε τις εκτιμώμενες θέσεις των σκιάστρων:
    east = θέση ανατολικού σκιάστρου
    west1 = θέση δυτικού σκιάστρου 1
    west2 = θέση δυτικού σκιάστρου 2
    positions = [east, west1, west2]

Αν το κεντρικό σκίαστρο είναι ανοιχτό τότε:

    Αν οι θέσεις positions των σκιάστρων είναι διαφορετικές μεταξύ τους
    ΚΑΙ το άθροισμα των θέσεων είναι μεγαλύτερο από 15 τότε:

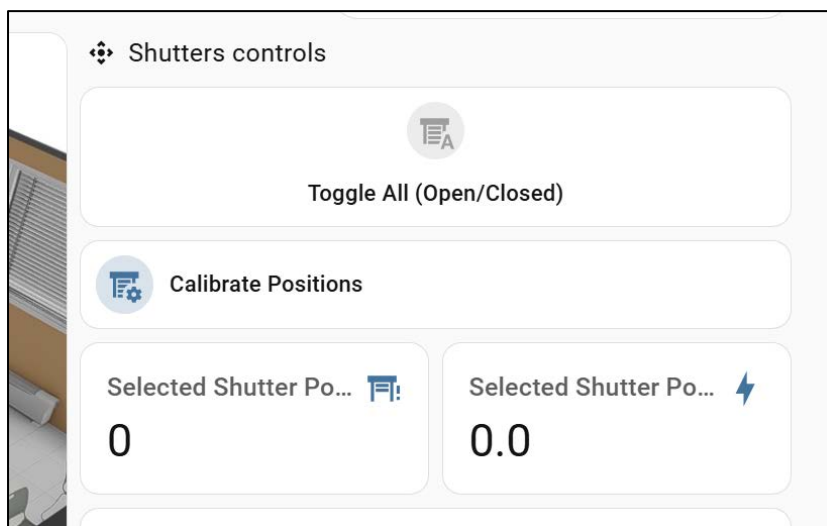
        Παράλληλα για κάθε σκίαστρο:
            Αν η θέση του είναι μικρότερη από 9 τότε:
                Ξεκίνα άνοιγμα σκιάστρου
                Περίμενε μέχρι η θέση του να γίνει μεγαλύτερη από 9
                ή να περάσουν 10 δευτερόλεπτα
                // Η αναμονή των 10 δευτερολέπτων είναι τυπική τιμή ασφαλείας.
                // Δεν αναμένεται να ενεργοποιηθεί, καθώς η συνθήκη θέσης καλύ-
                πτει το trigger.
                Σταμάτησε την κίνηση του σκιάστρου

            Εμφάνισε ειδοποίηση με τις προηγούμενες θέσεις των σκιάστρων

        Αλλιώς αν:
            Οι θέσεις είναι διαφορετικές μεταξύ τους
            Ή όλες οι θέσεις είναι ίδιες και τουλάχιστον μία είναι 0 τότε:

                Κλείσε όλα τα σκίαστρα του εργαστηρίου 4 // Lab04 shutters group
                Περίμενε μέχρι το entity του group να δηλωθεί ως "closed" ή να περάσουν
                10 δευτερόλεπτα
                // Η αναμονή των 10 δευτερολέπτων είναι τυπική τιμή ασφαλείας.
                // Δεν αναμένεται να ενεργοποιηθεί, καθώς η συνθήκη θέσης καλύ-
                πτει το trigger.

            Εμφάνισε ειδοποίηση με τις προηγούμενες θέσεις των σκιάστρων
```



Εικόνα 18: Το κουμπί calibrate positions όπου γίνεται εμφανές στη Lab4 διεπαφή χρήστη (ενότητα 5.1).

4.1.3 Κύριος αυτοματισμός και ανάλυση των συνθηκών του

Ο αυτοματισμός στο Εργαστήριο 4 υλοποιήθηκε με στόχο τη δυναμική και έξυπνη διαχείριση των σκιάστρων, με σκοπό τη βελτιστοποίηση των συνθηκών φωτισμού και φυσικής θερμοκρασίας του χώρου. Η λειτουργία του βασίζεται στη συνεχή παρακολούθηση των αισθητήρων θερμοκρασίας, φωτεινότητας και ηλιακής ακτινοβολίας και στην (εκτιμώμενη) θέση των σκιάστρων.

Ο αυτοματισμός ενεργοποιείται περιοδικά, κάθε 2 λεπτά, και εκτελεί μια σειρά από ελέγχους ώστε να αποφασίσει αν πρέπει να ανοίξει, κλείσει ή διατηρήσει τη θέση των σκιάστρων. Αρχικά, γίνεται ένα χρονικό καλιμπράρισμα των σκιάστρων μέσω του script της ενότητας 4.1.2, ώστε να συγχρονιστεί η θέση τους. Έπειτα, η λογική του βασίζεται σε τρία βασικά σενάρια λειτουργίας:

1. *Ανοίγμα σκιάστρων* όταν οι συνθήκες φωτισμού και θερμοκρασίας είναι χαμηλές, ενώ οι τιμή της ηλιακής ακτινοβολίας είναι υψηλή.
2. *Κλείσιμο σκιάστρων* όταν υπάρχει υπερβολική φωτεινότητα εσωτερικά ή αυξημένη θερμοκρασία.
3. *Αυτόματο κλείσιμο κατά τη διάρκεια της νύχτας* για λόγους ασφάλειας.

Η προσέγγιση αυτή επιτρέπει προοδευτική ρύθμιση της θέσης των σκιάστρων, κατά 2 μονάδες τη φορά, ώστε να επιτυγχάνεται λεπτομερής έλεγχος του φυσικού φωτισμού και της θερμικής άνεσης στον χώρο.

Ακολουθεί ψευδοκώδικας που αποτυπώνει τη λογική του αυτοματισμού.

Κάθε 2 λεπτά:

Εκτέλεση script: `calibrate_shutters_positions`

Αναμονή μέχρι να ολοκληρωθεί η εκτέλεση του script

Αν (Θερμοκρασία < 20°C) ΚΑΙ (Ηλιακή Ακτινοβολία > 200 W/m²) ΚΑΙ (Φωτεινότητα < 270 lux) ΚΑΙ (Θέση Σκιάστρων < 9):

Ανοιγμα σκιάστρων

Αναμονή 2 δευτερόλεπτα

Σταμάτημα σκιάστρων

Αλλιώς αν (Φωτεινότητα > 350 lux) ΚΑΙ (Ηλιακή Ακτινοβολία > 200 W/m²) ΚΑΙ (Θερμοκρασία > 27°C):

Κλείσιμο σκιάστρων

Αναμονή 2 δευτερόλεπτα

Σταμάτημα σκιάστρων

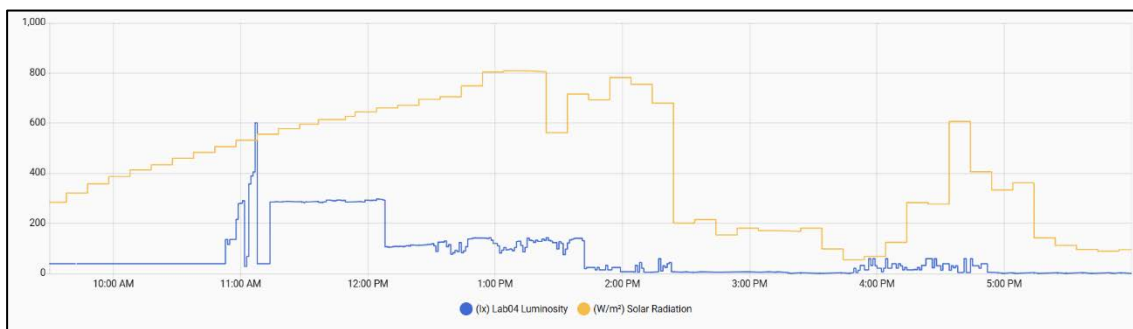
Αλλιώς αν (Ωρα μεταξύ 21:00 και 07:00) ΚΑΙ (Σκιάστρα δεν είναι ήδη κλειστά):

Κλείσιμο σκιάστρων

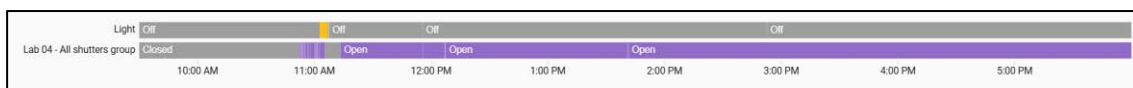
Η επιλογή των ορίων για τις τιμές των αισθητήρων δεν είναι αυθαίρετη, αλλά βασίζεται σε βιβλιογραφικά πρότυπα και μελέτες. Συγκεκριμένα, για τη φωτεινότητα η τιμή των 270 lux ως κατώφλι για άνοιγμα και των 320 lux για κλείσιμο, προκύπτει από την γενική απαίτηση για περίπου 300 lux σε αίθουσες διδασκαλίας με χρήση υπολογιστών, σύμφωνα με σχετικές πληροφορίες από την ιστοσελίδα thorlux.com [27] και τις προδιαγραφές του ΦΕΚ 221Α/1994 [28], που αφορούν τις προδιαγραφές ασφαλείας κατά την εργασία σε χώρους γραφείων με χρήση ηλ. υπολογιστών. Οι τιμές αυτές, με βάση και τις δοκιμές που έγιναν κατά τη διάρκεια αυτής της πτυχιακής, φαίνεται να εξασφαλίζουν επαρκή οπτική άνεση, χωρίς υπερβολική φωτεινότητα. Αντίστοιχα, το ΦΕΚ[28], περιγράφει πως η θερμοκρασία σε τέτοιους χώρους θα πρέπει να βρίσκεται μέσα στο εύρος των 20°C με 27°C, ως όριο μεταξύ άνετης και δυσάρεστης θερμικής κατάστασης, καθώς τιμές εκτός αυτού του φάσματος, ενδέχεται να προκαλέσουν αίσθηση θερμικής δυσφορίας, ειδικά σε χώρους με έντονη εισροή ηλιακής ακτινοβολίας.

Η θέση των σκιάστρων λαμβάνεται επίσης υπόψη, ώστε να αποφεύγεται η περιττή ενεργοποίηση του μηχανισμού σκίασης, όταν τα σκιάστρα είναι ήδη πλήρως ανοιχτά ή κλειστά, συμβάλλοντας έτσι στην αποδοτικότερη λειτουργία του συστήματος και στη μείωση της φθοράς των μηχανισμών. Τέλος, η ηλιακή ακτινοβολία με όριο τα 200 W/m² θεωρήθηκε ως σημείο μετάβασης προς υψηλή ενεργειακή φόρτιση του χώρου, επηρεάζοντας τόσο τις θερμικές συνθήκες όσο και τα επίπεδα, φυσικού, φωτισμού.

Τα γραφήματα που ακολουθούν, βασίζονται σε δεδομένα της 3^{ης} Οκτωβρίου και χρησιμοποιούνται για δοκιμή των ορίων που έχουν τεθεί για την ενεργοποίηση του αυτοματισμού των σκιάστρων. Ο αυτοματισμός δεν ήταν ενεργοποιημένος αλλά τα σκίαστρα παρέμειναν χειροκίνητα ανοιχτά σχεδόν όλη την ημέρα, σε διάφορες θέσεις, ενώ ο τεχνητός φωτισμός ενεργοποιήθηκε για κάποια λεπτά το πρωί, ώστε να καταγραφεί η μέγιστη τιμή φωτεινότητας που μπορεί να προσφέρει ο φωτισμός του χώρου. Το γεγονός ότι τα σκίαστρα ήταν ήδη ανοιχτά γύρω στις 11:00am, δεν επηρέασε σημαντικά τη μέτρηση, καθώς ο τεχνητός φωτισμός υπερκάλυπτε τον φυσικό τη συγκεκριμένη μέρα και ώρα. Όλη η παρακάτω καταγραφή επιτρέπει την αξιολόγηση της συμπεριφοράς των αισθητήρων φωτεινότητας και ηλιακής ακτινοβολίας σε πραγματικές συνθήκες, επιβεβαιώνοντας ότι τα επιλεγμένα όρια ανταποκρίνονται λειτουργικά στις ανάγκες του χώρου.

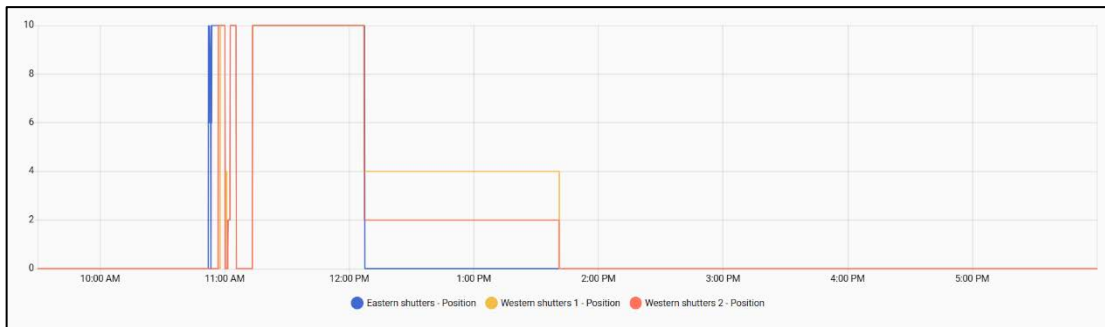


Εικόνα 19: Φωτεινότητα και Ηλιακή Ακτινοβολία



Εικόνα 20: Κατάσταση Φωτισμού και Σκιάστρων,
(με bug μετά τις 1:30pm)

Αξίζει να σημειωθεί ότι στην Εικόνα 20, μετά τις 13:30, τα σκίαστρα εμφανίζονται ως ανοιχτά ενώ στην πραγματικότητα δεν είναι. Το φαινόμενο αυτό αναδεικνύει την ανάγκη για επαναβαθμονόμηση (calibrate) τους μετά από χειροκίνητη παρέμβαση (βλ. υποενότητα 4.1.2), ώστε να διασφαλίζεται η ακρίβεια των δεδομένων.



Εικόνα 21: Θέσεις σκιάστρων

4.2 Συμπύκνωση στα τζάμια

Η συγκεκριμένη υλοποίηση πραγματοποιήθηκε σε διαφορετικό χώρο από αυτόν του «Εργαστήριο 4» (που αποτελεί τον βασικό χώρο ενδιαφέροντος), καθώς ο συγκεκριμένος χώρος δεν διαθέτει αισθητήρα εσωτερικής υγρασίας, στοιχείο απαραίτητο για τον υπολογισμό της υγρασίας δρόσου. Η αρχική ανάπτυξη του μηχανισμού ωστόσο έγινε και ολοκληρώθηκε, με το σκεπτικό ότι θα εγκαθιστούνταν σχετικός αισθητήρας και στο εργαστήριο, κάτι που τελικά δεν πραγματοποιήθηκε. Παρόλα αυτά, η παρακάτω υλοποίηση εντάσσεται στο ίδιο τεχνολογικό και λειτουργικό πλαίσιο του έργου, αξιοποιώντας τις ίδιες αρχές σχεδίασης και τα εργαλεία του Home Assistant και δεδομένης της χρηστικότητάς της, αξίζει να αναλυθεί παρακάτω και να εξεταστεί η ενσωμάτωσή της στην έξυπνη αίθουσα διδασκαλίας, στο μέλλον.

Έτσι, μέρος των αυτοματισμών του συστήματος είναι και ο παρακάτω απλός αλλά λειτουργικός μηχανισμός παρακολούθησης και ειδοποίησης σχετικά με την πιθανότητα εμφάνισης υγρασίας στις εσωτερικές επιφάνειες των παραθύρων. Η υγρασία αυτή οφείλεται στη συμπύκνωση των υδρατμών του αέρα, ένα φυσικό φαινόμενο που μπορεί να οδηγήσει σε προβλήματα υγρασίας, μούχλας και φθοράς των δομικών υλικών[32].

Η λογική εκτίμηση της πιθανότητας συμπύκνωσης υλοποιείται μέσω ενός binary template αισθητήρα (**binary_sensor.water_condensation_risk**), ο οποίος συγκρίνει τη θερμοκρασία δρόσου με την εσωτερική θερμοκρασία του χώρου. Αν η θερμοκρασία δρόσου είναι ίση ή μεγαλύτερη από την εσωτερική θερμοκρασία, ο αισθητήρας ενεργοποιείται με **state: "wet"**, υποδεικνύοντας πιθανή εμφάνιση συμπύκνωσης.

Ακολουθεί η υλοποίηση του binary sensor template σε Jinja2:

```
- binary_sensor:
  - name: "Water Condensation Risk"
    unique_id: xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx # (UUID - Hidden)
    device_class: moisture
```

```

icon: mdi:snowflake-thermometer
state: >-
    {{ (states('sensor.dew_temperature') | float) >= (states('sensor.temperature_in-
side') | float) }}

```

Και η λογική του αυτοματισμού, η οποία μπορεί να περιγράψει με τον εξής ψευδο-κώδικα:

```

Αν ο αισθητήρας συμπύκνωσης είναι ενεργός ("Wet"):
    Στείλε ειδοποίηση: "Warning: The dew point is below freezing! Water conden-
sation may appear"
Αλλιώς:
    Στείλε ειδοποίηση: "There is no water condensation on windows anymore."

```

Παράλληλα, για να υπάρχει και ορατή στο dashboard η παραπάνω πληροφορία-μήνυμα, δημιουργήθηκε και ένας αισθητήρας ακόμα (**sensor.condensation_message**). Ο αισθητήρας αυτός εμφανίζει τα αντίστοιχα μηνύματα με αυτά του αυτοματισμού και με βάση την ίδια λογική λειτουργίας.

Η υλοποίηση του αισθητήρα σε Jinja2 έχει ως εξής:

```

- name: "Condensation Message"
  unique_id: xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx # (UUID - Hidden)
  icon: mdi:snowflake-thermometer
  state: >-
    {% set dew_temp = states("sensor.dew_temperature") %}
    {% set in_temp = states("sensor.temperature_inside") %}
    {% if dew_temp | float >= in_temp | float %}
      Warning: The dew point is below freezing! Water condensation may appear
    {% else %}
      There is no water condensation on windows
    {% endif %}

```

Για να λειτουργήσουν όμως όλα τα παραπάνω, χρειάζεται να υπολογιστεί κάπου και η θερμοκρασία δρόσου. Για αυτό, δημιουργήθηκαν ακόμα 2 template αισθητήρες που υπολογίζουν τη θερμοκρασία δρόσου με δύο τρόπους, όπως περιγράφονται παρακάτω.

1. Στον πρώτο τρόπο υπολογισμού, εφαρμόζεται μια προσεγγιστική μέθοδος[33] για την εκτίμηση της θερμοκρασίας δρόσου, βασισμένη στον εμπειρικό τύπο

$$T_{dew} \approx T_{in} - \frac{(100 - RH)}{5}$$

Όπου T_{in} είναι η εσωτερική θερμοκρασία (°C) του χώρου και όπου RH (%) η σχετική υγρασία.

Η αντίστοιχη υλοποίηση στο Home Assistant πραγματοποιείται μέσω του παρακάτω Jinja2 template:

```
- name: "Dew Approximate"
  unique_id: xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx # (UUID - Hidden)
  device_class: temperature
  state_class: measurement
  icon: mdi:thermometer-alert
  unit_of_measurement: "°C"
  state: >-
    {{ ( states('sensor.temperature_inside') | float - ((100 -
states('sensor.humidity_inside') | int) / 5)) | float )}}
```

2. Στον δεύτερο τρόπο υπολογισμού, βασιζόμαστε στον τύπο Magnus-Tetens, ο οποίος προσφέρει μεγαλύτερη ακρίβεια [31]. Ο τύπος αυτός χρησιμοποιείται ευρέως στη μετεωρολογία για τον υπολογισμό της θερμοκρασίας δρόσου με υψηλή ακρίβεια. Η μαθηματική του μορφή είναι:

$$T_{dew} = \frac{\beta \cdot \gamma(T_{in}, RH)}{\alpha - \gamma(T_{in}, RH)}$$

Όπου:

- $\gamma(T_{in}, RH) = \frac{\alpha \cdot T_{in}}{\beta + T_{in}} + \ln\left(\frac{RH}{100}\right)$
- T_{in} : η εσωτερική θερμοκρασία (°C)
- RH: η σχετική υγρασία (%)
- $a = 17.27$, $b = 237.7$: εμπειρικές σταθερές

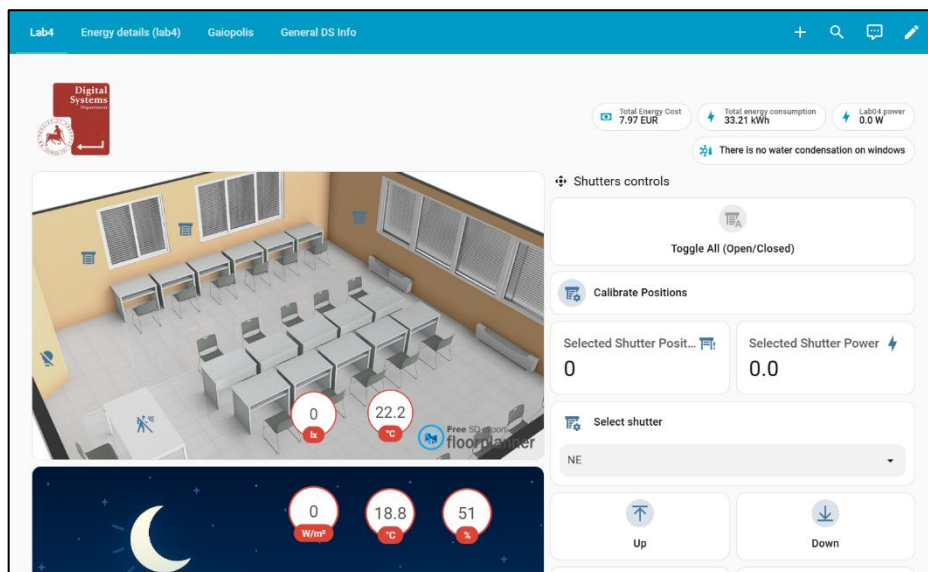
Η υλοποίηση του υπολογισμού μέσω Jinja2 template φαίνεται παρακάτω:

```
- name: "Dew Temperature"
  unique_id: xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx # (UUID - Hidden)
  device_class: temperature
  state_class: measurement
  icon: mdi:thermometer-alert
  unit_of_measurement: "°C"
  state: >-
    {{ (( (237.7 * (((17.27*(states('sensor.temperature_inside') |
float))/(237.7+(states('sensor.temperature_inside') | float)))+log(((states('sensor.hu-
midity_inside') | int)/100),e))) / (17.27 - (((17.27*(states('sensor.temperature_in-
side') | float))/(237.7+(states('sensor.temperature_inside') |
float)))+log(((states('sensor.humidity_inside') | int)/100),e)))) | float) | round(1,
"ceil", 0) )}}
```


5 Διεπαφή Χρήστη

Η υλοποίηση του συστήματος για την έξυπνη αίθουσα διδασκαλίας συνοδεύτηκε από τη δημιουργία μίας προσαρμοσμένης διεπαφής χρήστη (ή dashboard ή web interface) στο Home Assistant, το οποίο επιτρέπει την παρακολούθηση και τον έλεγχο συσκευών και την ανάλυση των δεδομένων σε πραγματικό χρόνο, με στόχο την εργονομία, την όμορφη οπτική αναπαράσταση τους και την ευκολία χρήσης από τον τελικό χρήστη.

Το dashboard, οργανώνεται σε τέσσερις βασικές καρτέλες (tabs), οι οποίες αντιστοιχούν σε διαφορετικές θεματικές ενότητες του συστήματος. Κάθε καρτέλα μπορεί να περιλαμβάνει διαδραστικά στοιχεία, γραφήματα, αισθητήρες και ελεγκτές, με στόχο την πλήρη εποπτεία και διαχείριση των λειτουργιών του χώρου. Στις επόμενες ενότητες παρουσιάζεται αναλυτικά το περιεχόμενο και η λειτουργία κάθε καρτέλας, με στόχο την ανάδειξη της πρακτικής αξίας και τεχνολογικής ολοκλήρωσης του συστήματος.

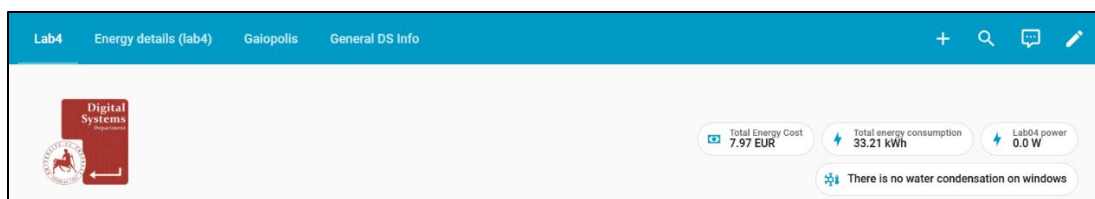


Εικόνα 22: Η αρχική/βασική προβολή του HA dashboard, στη σελίδα ελέγχου του εργαστηρίου.

5.1 Lab4 – Εικονική Κάτοψη και Έλεγχος Συσκευών

Η καρτέλα για το Lab4, αποτελεί την αρχική σελίδα του dashboard την οποία αντικρίζει ο χρήστης σε κάθε log-in του. Είναι σχεδιασμένη ώστε να προσφέρει μια συνολική εικόνα του χώρου συνδυάζοντας έλεγχο και λειτουργικότητα με ψηφιακές αναπαραστάσεις.

Το επάνω μέρος της καρτέλας (header) περιλαμβάνει το λογότυπο του Τμήματος και μια σειρά από «badges» που εμφανίζουν γενικές πληροφορίες. Οι πληροφορίες αυτές παρουσιάζουν χρήσιμα στοιχεία στον χρήστη, σύντομα, χωρίς να πιάνουν χώρο στον βασικό κορμό της σελίδας.



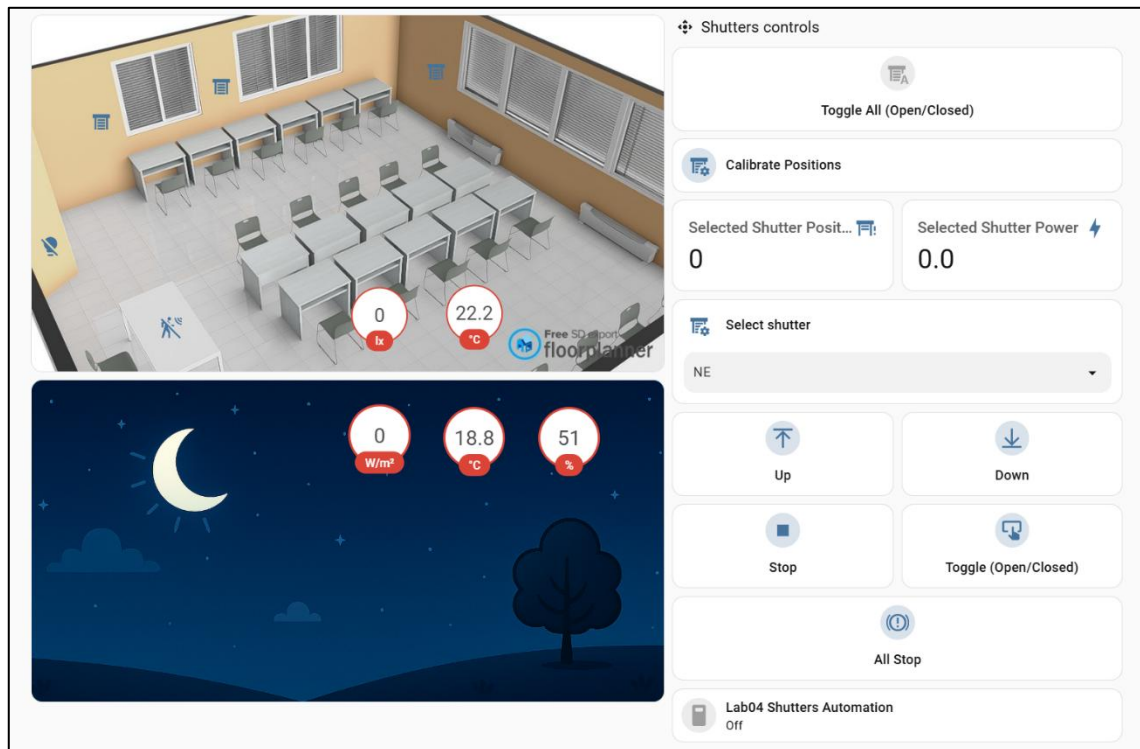
Εικόνα 23: Το Header του Lab4 tab

Έπειτα, το κεντρικό τμήμα του Lab4 UI είναι αφιερωμένο στον έλεγχο των σκιάστρων και περιλαμβάνει δύο βασικές περιοχές. Η πρώτη είναι μια διαδραστική κάτοψη του χώρου, που δημιουργήθηκε με το online εργαλείο Floorplanner[37], όπου απεικονίζονται τα σκίαστρα με εικόνες που αλλάζουν δυναμικά ανάλογα με την κατάστασή τους (δηλαδή αλλάζουν και στην εικόνα ως ανοιχτά ή κλειστά). Ακόμα στην κάτοψη, έχουν ενσωματωθεί εικονίδια για τον άμεσο έλεγχο των σκιάστρων, καθώς και για την παρακολούθηση των αισθητήρων της θερμοκρασίας, φωτεινότητας, κίνησης και φωτισμού.

Ακριβώς από κάτω της κάτοψης, βρίσκεται μια αφαιρετική απεικόνιση του εξωτερικού περιβάλλοντος, η οποία προσαρμόζεται ανάλογα με τις συνθήκες του εξωτερικού - φυσικού - φωτισμού. Εμφανίζει νυχτερινό ουρανό όταν δεν υπάρχει ηλιακό φως και πρωινό ουρανό (ηλιοφάνεια) όταν ο solar radiation αισθητήρας ανεβάσει κατάλληλα την τιμή του. Επίσης στην αφαιρετική εικόνα, υπάρχουν εικονίδια με τις τιμές εξωτερικής θερμοκρασίας, υγρασίας και ηλιακής ακτινοβολίας.

Στη δεξιά πλευρά της των δύο διαδραστικών εικόνων, βρίσκονται τα πλήκτρα ελέγχου των σκιάστρων. Ο χρήστης μπορεί να επιλέξει πιο σκίαστρο επιθυμεί να ελέγξει ώστε να το ανοίξει, να το κλείσει, να το σταματήσει ή να εναλλάξει την κατάστασή του. Ή μπορεί να επιλέξει κάποιο από τα άλλα κουμπιά που υπάρχουν με δυνατότητα ομαδικού ελέγχου όλων των σκιάστρων, βαθμονόμησης θέσης (βλ. ρουτίνα ενότητα 4.1.2), καθώς

και ενεργοποίησης ή απενεργοποίησης του αυτοματισμού σκίασης. Επιπλέον, εμφανίζονται πεδία όπου δείχνουν τη θέση και την κατανάλωση ισχύος κάθε στιγμή, για το επιλεγμένο σκίαστρο.

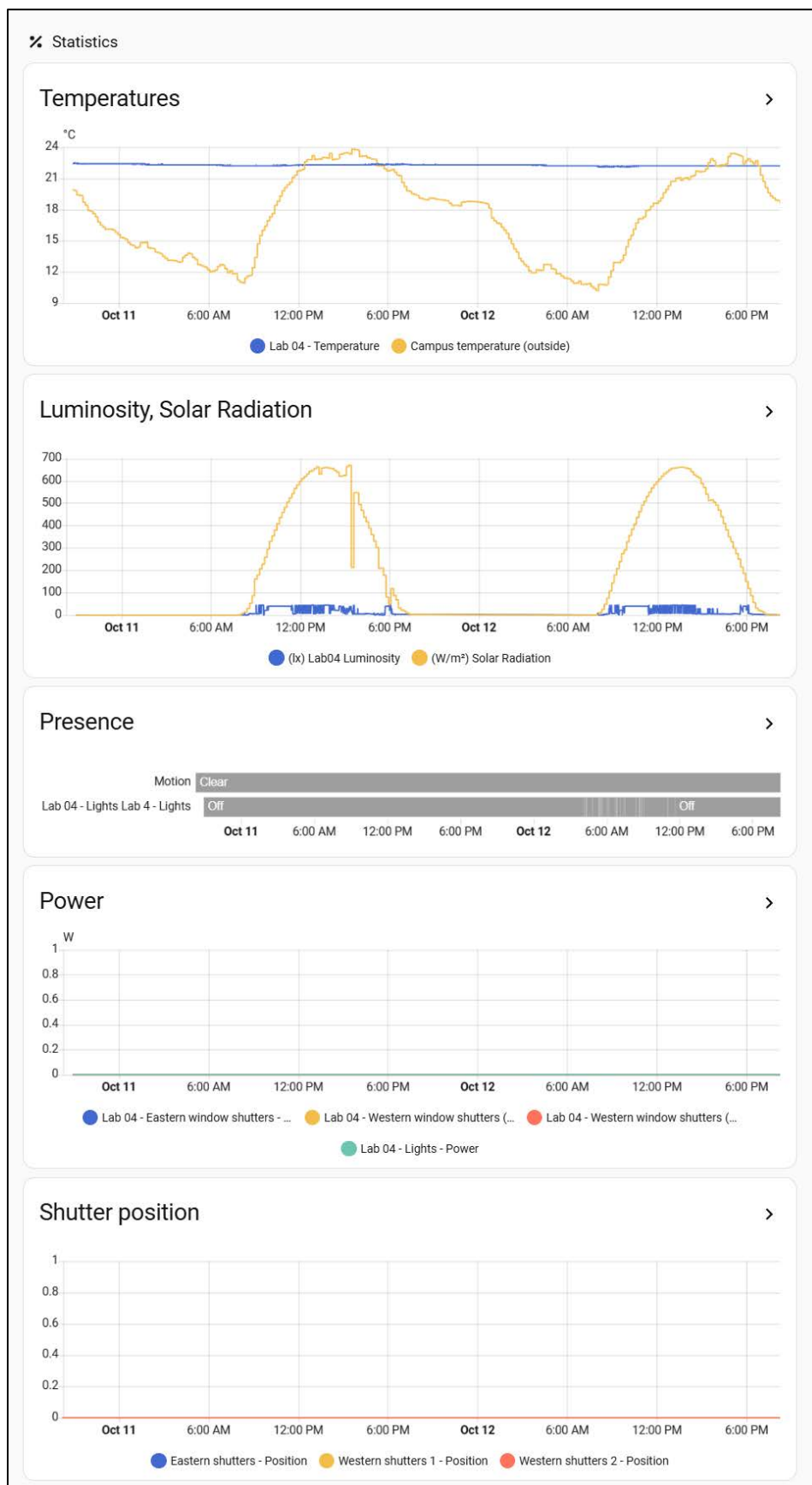


Εικόνα 24: Η διαδραστική απεικόνιση του χώρου και τα controls των σκιάστρων

Το τέταρτο και τελευταίο τμήμα της καρτέλας είναι αφιερωμένο στα στατιστικά. Περιλαμβάνει ιστορικά γραφήματα για τις τελευταίες 48 ώρες, οργανωμένα στις θεματικές ενότητες:

- *θερμοκρασίες (εσωτερική και εξωτερική),*
- *φωτεινότητα και ηλιακή ακτινοβολία, όπου χρειάστηκε να δημιουργηθούν template helpers που να αντιγράφουν κάθε φορά τις μετρήσεις, χωρίς όμως να τους δηλωθεί κάποια μονάδα μέτρησης. (παρά μόνο στον τίτλο/ονομά τους), ώστε να μπορούν να μουν στο ίδιο γράφημα,*
- *παρουσία (ανίχνευση κίνησης),*
- *κατανάλωση ισχύος από σκίαστρα και φωτισμό,*
- *καθώς και την εκτίμηση των θέσεων των σκιάστρων.*

Τα γραφήματα αυτά προσφέρουν στον χρήστη μια αναλυτική εικόνα της λειτουργίας του συστήματος με χρονική διάσταση. Ενώ το γεγονός ότι είναι το ένα δίπλα στο άλλο, μπορεί να βοηθήσει στην έκβαση συμπερασμάτων από αυτά.

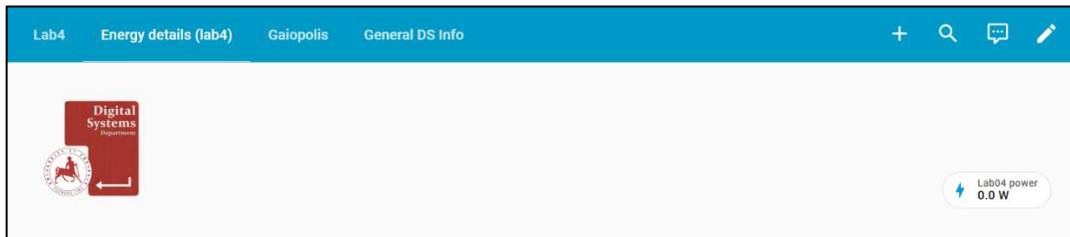


Εικόνα 25: Τα στατιστικά δεδομένα του Lab4 που παρουσιάζονται μέσω γραφημάτων

5.2 Energy details - Παρακολούθηση Ενέργειας και Κόστους

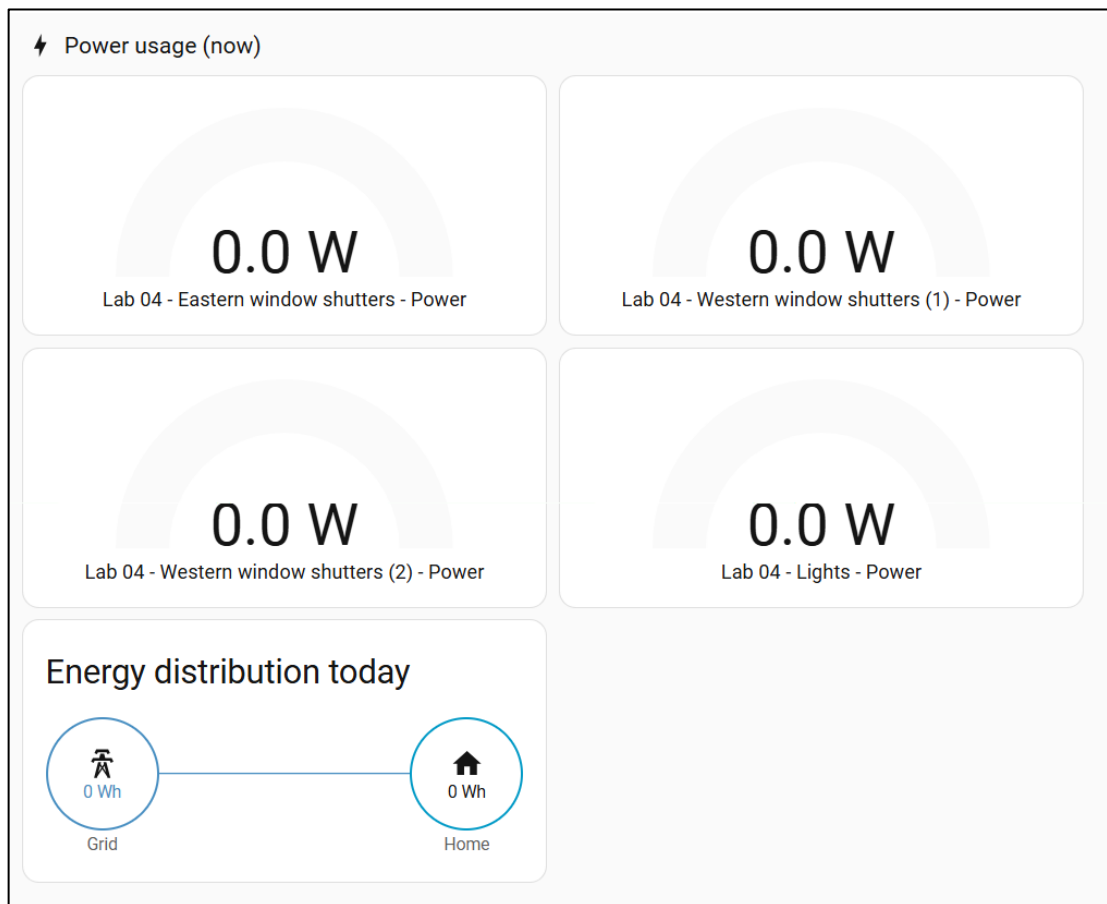
Αντίστοιχα με την ενότητα 5.1, η καρτέλα (Tab) «Energy details (lab4)» ακολουθεί την παρόμοια σχεδιαστική φιλοσοφία με διακριτά τμήματα τα οποία περιλαμβάνουν:

- Header με το λογότυπο του τμήματος και ένα badge με την συνολική ενεργή κατανάλωση του έξυπνου χώρου διδασκαλίας

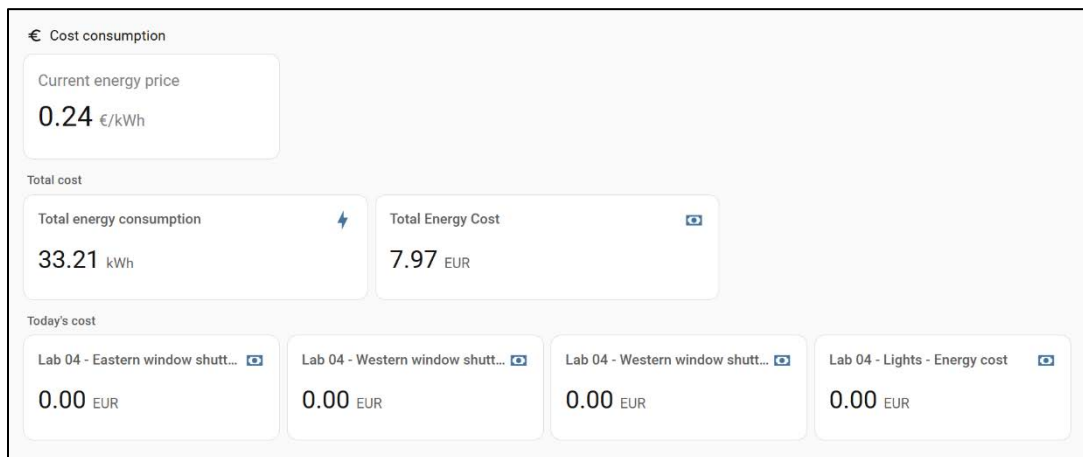


Εικόνα 26: Το Header του Energy tab

- Entity Cards που απεικονίζουν την τρέχουσα ισχύ, την κατανομή ενέργειας και το κόστος κατανάλωσης

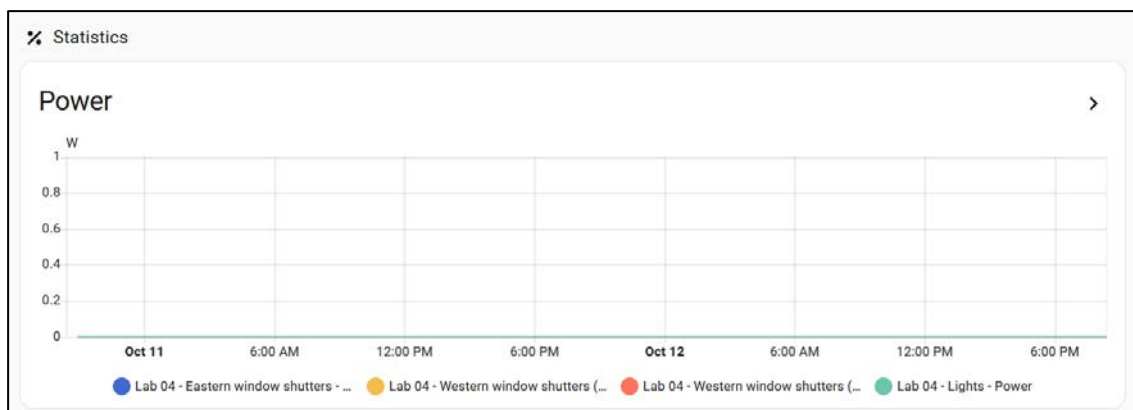


Εικόνα 27: Η διαδραστική απεικόνιση της τρέχουσας ισχύς και το ημερήσιο σύνολο

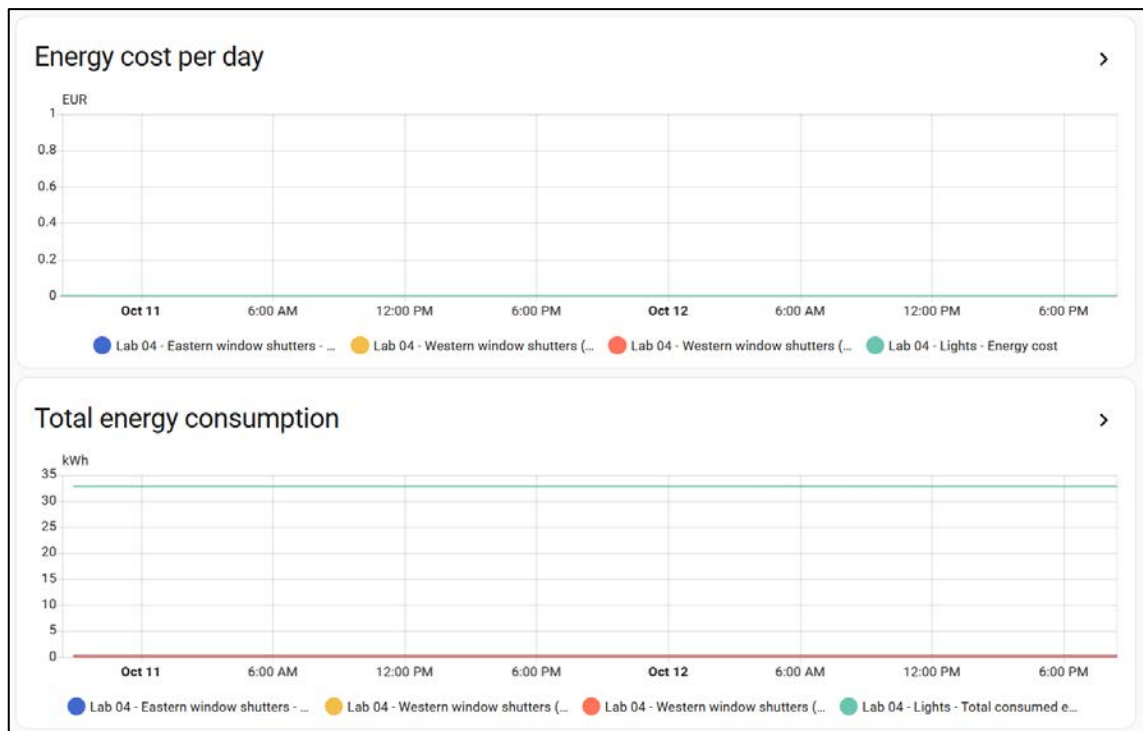


Εικόνα 28: Απεικόνιση δεδομένων κόστους κατανάλωσης

- Τέλος, το section των στατιστικών στο τέλος της σελίδας, στο οποίο παρουσιάζονται τα παραπάνω δεδομένα με τη μορφή γραφημάτων. Πιο συγκεκριμένα, παρουσιάζονται οι καταγραφές κατανάλωση ισχύος, το ημερήσιο κόστος και τη συνολική ενεργειακή χρήση.



Εικόνα 29: Τα στατιστικά δεδομένα του Energy details (lab4), που παρουσιάζονται μέσω γραφημάτων



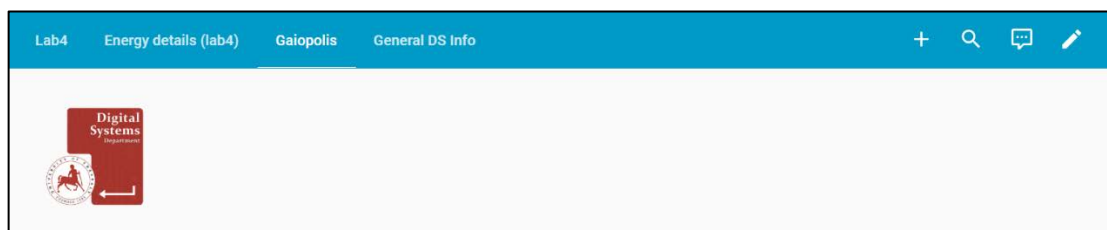
Εικόνα 30: Τα υπόλοιπα δύο γραφήματα που παρουσιάζονται στο Energy details (lab4)

5.3 Γαίopolis – Δεδομένα και μετρήσεις από τον Πανεπιστημιακό χώρο

Η καρτέλα Γαίopolis ήταν από τις πρώτες που σχεδιάστηκαν στο πλαίσιο του dashboard, πριν την προμήθεια και εγκατάσταση του εξοπλισμού για το Lab4. Αξιοποιεί υπάρχον υποδομή του Πανεπιστημίου, και συγκεκριμένα τον μετεωρολογικό σταθμό της ομάδας GreenDigital, ο οποίος παρέχει αξιόπιστα δεδομένα για το εξωτερικό και εσωτερικό περιβάλλον του ευρύτερου χώρου του τμήματος.

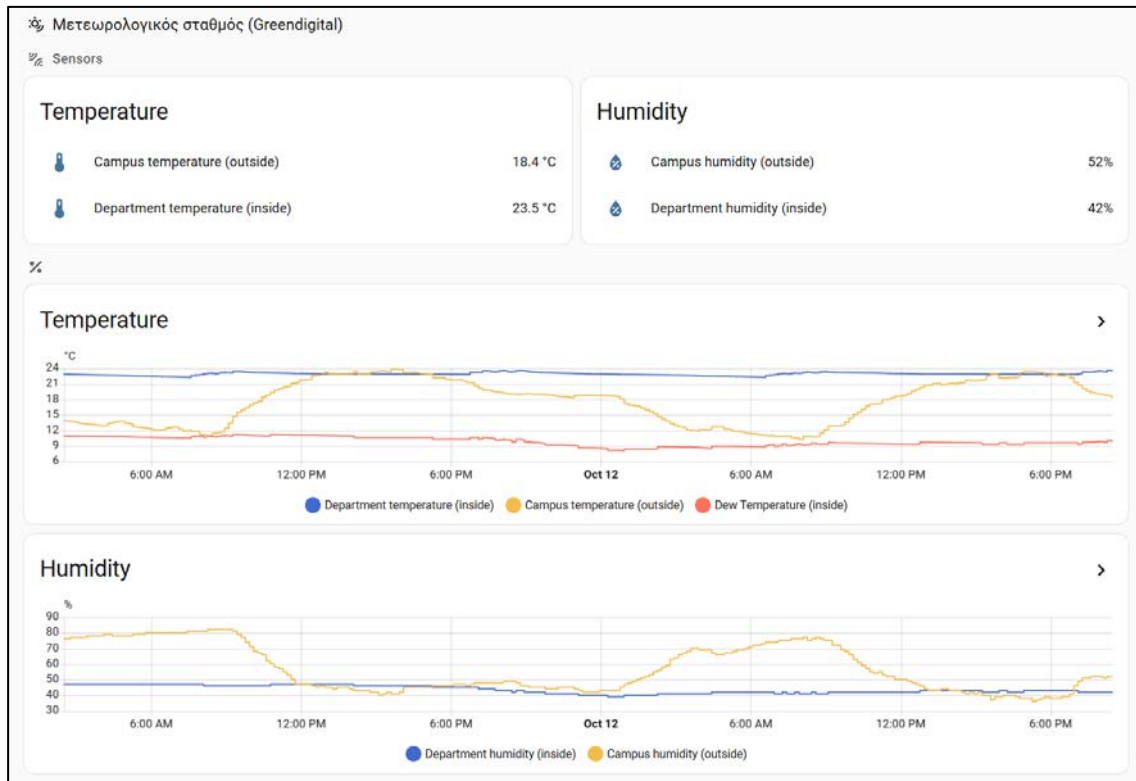
Η διεπαφή της καρτέλας ακολουθεί παρόμοια λογική με τις προηγούμενες δύο και σε αυτή ο χρήστης μπορεί να βρει:

- Για άλλη μια φορά το Header, όπου στη συγκεκριμένη σελίδα περιλαμβάνει μόνο το λογότυπο του Τμήματος



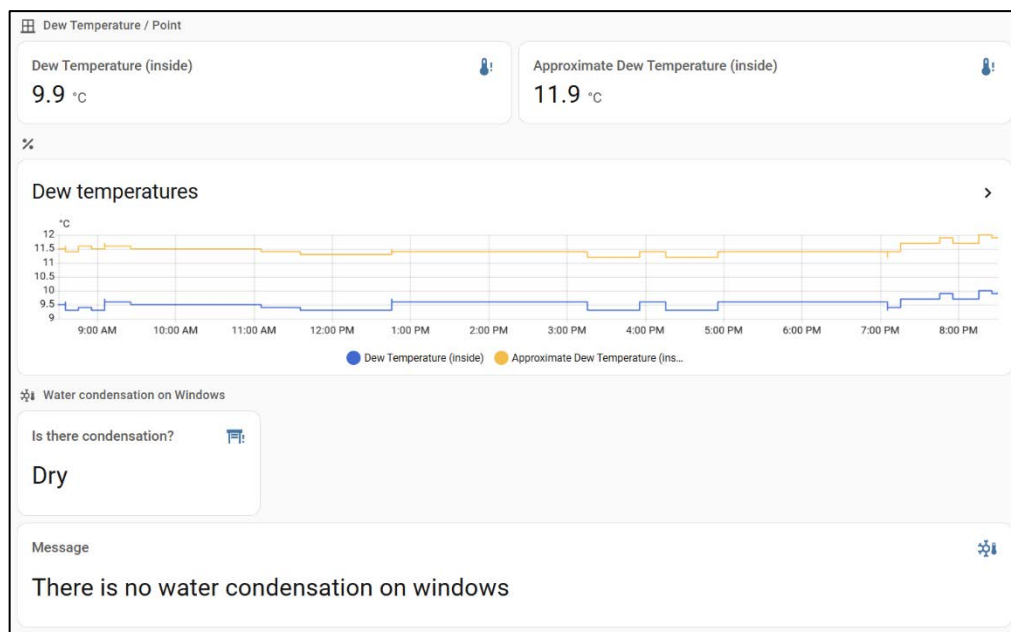
Εικόνα 31: Το Header του Gaiopolis tab

- Entity cards τα οποία παρουσιάζουν τις μετρήσεις θερμοκρασίας του εσωτερικού και εξωτερικού χώρου και συνοδεύονται από τα γραφήματά τους



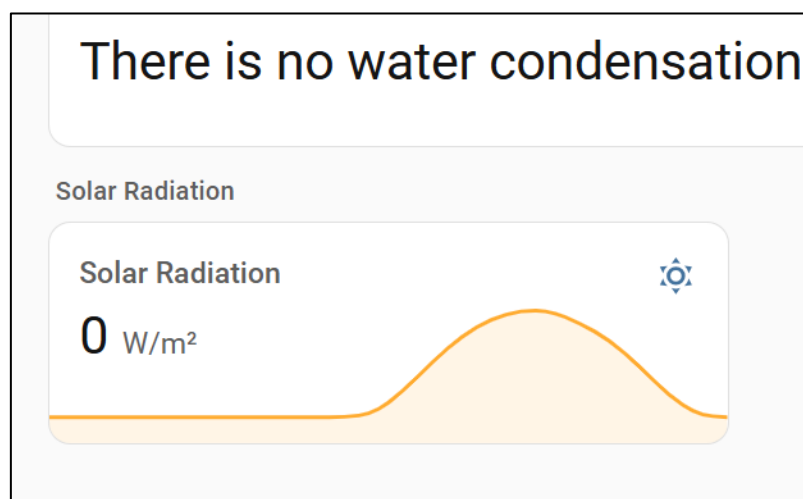
Εικόνα 32: Τα δεδομένα του μετεωρολογικού σταθμού στο web interface του HA

- Entity cards για την παρουσίαση των υπολογισμών της θερμοκρασίας δρόσου και των στατιστικών γραφημάτων της. Αν η θερμοκρασία δρόσου είναι μεγαλύτερη της εσωτερικής θερμοκρασίας θα εμφανιστεί υγρασία στα τζάμια. Για αυτό, υπάρχει και κατάλληλο μήνυμα στη διεπαφή για να ενημερώνει τον χρήστη.



Εικόνα 33: Μετρήσεις, στατιστικά δεδομένα και ειδοποιήσεις σχετικά με τη θερμοκρασία δρόσου, στο HA

- Τέλος, στη σελίδα υπάρχει και ένα entity sensor card, για την παρουσίαση της τιμής της ηλιακής ακτινοβολίας με μία μικρή και απλή οπτική αναπαράσταση που δείχνει «χονδρικά» την μορφή της γραμμής που θα είχε το γράφημα του, αν παρουσιαζόταν και αυτό σε αυτή τη σελίδα.



Εικόνα 34: Το entity sensor card για τις τιμές του Solar Radiation στο Gaiopolis Tab

5.4 General DS Information – Γενικότερες πληροφορίες σχετικά με το Τμήμα Ψηφιακών Συστημάτων

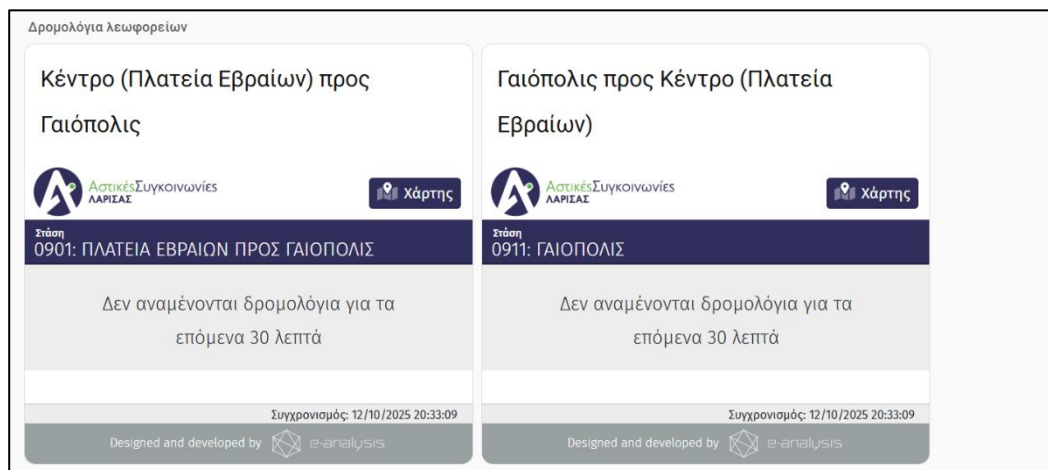
Η καρτέλα General DS Info (Information), είναι μία μοναδική διεπαφή χρήστη, καθώς ενσωματώνει λειτουργίες που ενισχύουν τη χρηστικότητα του dashboard για τους χρήστες του πανεπιστημίου - και κυρίως τους φοιτητές. Σε αυτή, ο χρήστης μπορεί να έχει πρόσβαση σε:

- Ένα custom Header με κουμπιά (buttons cards) για άμεση πρόσβαση σε χρήσιμες υπηρεσίες του Τμήματος ΨΣ.



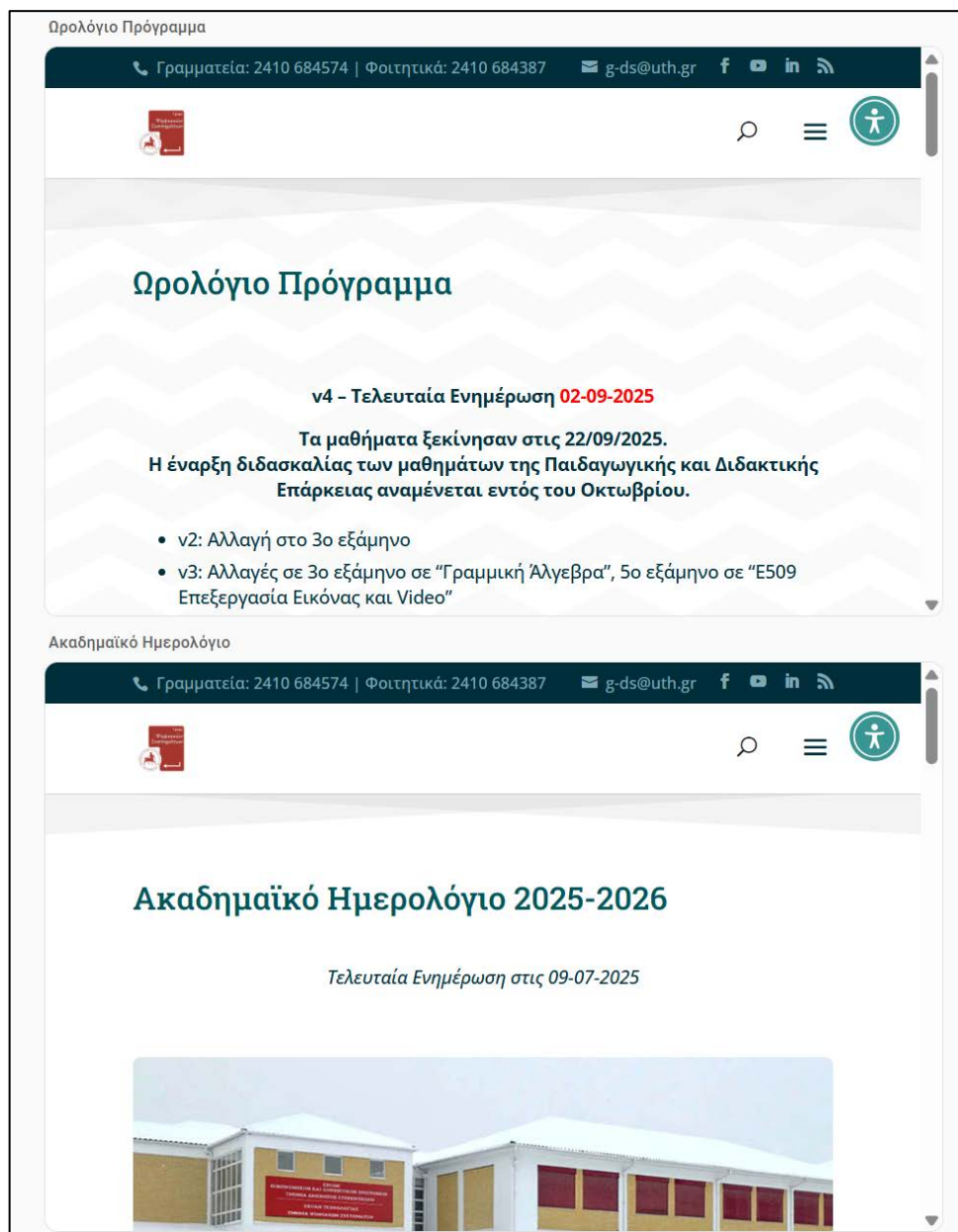
Εικόνα 35: Το header της σελίδας General DS info

- Στο live πρόγραμμα των δρομολογίων για λεωφορεία που κατευθύνονται από το κέντρο της Λάρισας προς τον χώρο του Πανεπιστημίου αλλά και για αυτά που αναχωρούν από αυτόν, τοποθετημένα σε webpage cards, ώστε να φορτώνει τη σχετική ιστοσελίδα και να παρουσιάζεται η πληροφορία στον χρήστη.



Εικόνα 36: Ο πίνακας με τα δρομολόγια των λεωφορείων στη συγκεκριμένη διεπαφή

- Στο ωρολόγιο πρόγραμμα των μαθημάτων του Τμήματος καθώς και στο ακαδημαϊκό του ημερολόγιο, τα οποία παρουσιάζονται στον χρήστη μέσω webpage card, αντίστοιχα.



Εικόνα 37: Το ωρολόγιο πρόγραμμα και το Ακαδημαϊκό ημερολόγιο του ΨΣ, τοποθετημένα σε webpage cards στο ΗΑ

6 Συμπεράσματα

Η παρούσα πτυχιακή εργασία, είχε ως αντικείμενο τον σχεδιασμό και την υλοποίηση μιας έξυπνης αίθουσας διδασκαλίας, βασισμένη στο λογισμικό Home Assistant OS (HAOS/HA) και χώρο υλοποίησης το «Εργαστήριο 4» του Τμήματος Ψηφιακών Συστημάτων του Πανεπιστημίου Θεσσαλίας. Βασικός στόχος ήταν η αξιοποίηση έξυπνων συσκευών, αισθητήρων και αυτοματισμών, ώστε να μετατραπεί ένας ήδη σύγχρονος χώρος διδασκαλίας, σε έναν πιο λειτουργικό, αποδοτικό και ενεργειακά ευφυή χώρο.

Η εργασία εστίασε στην πρακτική εφαρμογή του λογισμικού HA, στην εγκατάσταση του απαραίτητου, για τους στόχους της, εξοπλισμού και την ανάπτυξη σεναρίων αυτοματισμού. Πιο συγκεκριμένα, υλοποιήθηκε η:

1. Εγκατάσταση του Home Assistant, ως VM σε Proxmox server του τμήματος.
2. Εγκατάσταση των συσκευών Shelly και η ενσωμάτωσή τους στο HA.
3. Ενσωμάτωση των δεδομένων του μετεωρολογικού σταθμού GreenDigital στο HA, με χρήση ειδικού script.
4. Δημιουργία ειδικής διεπαφής χρήστη για τον έλεγχο της αίθουσας διδασκαλίας και την παρακολούθηση των μετρήσεων και άλλων δεδομένων.
5. Ανάπτυξη αυτοματισμών για τον έλεγχο των σκιάστρων και του υπολογισμού την εκτιμώμενης θέσης τους καθώς και για την ενημέρωσή μας για την κατάσταση της συμπύκνωσης του αέρα στον χώρο.

Κατά την υλοποίηση ωστόσο των παραπάνω, προέκυψαν και ορισμένες τεχνικές προκλήσεις. Η πιο απαιτητική ήταν η απουσία της φάσης στα δυτικά σκιάστρα, με αποτέλεσμα να χρειάζεται τροποποίηση η ηλεκτρολογική υποδομή του σημείου, για την εγκατάσταση των Shelly Plus 2PM.

Επιπλέον, η ενσωμάτωση του Shelly Motion 2 παρουσίασε περιορισμούς στον τρόπο με τον οποίο το Shelly integration ενσωμάτωνε τις μετρήσεις της στο Home Assistant, καθώς φαίνεται να μην παρείχε αρκετά συχνή δειγματοληψία (πιθανότατα για εξοικονόμηση ρεύματος). Το πρόβλημα επιλύθηκε με την ανάπτυξη Python script, το οποίο αντλούσε δεδομένα και μετρήσεις κατευθείαν από τη συσκευή μέσω API κλήσεων, εξασφαλίζοντας την ακρίβεια και τη σταθερότητα που απουσίαζε.

Αναστοχαζόμενος την πορεία της εργασίας, η γενική μου προσέγγιση γύρω από τους στόχους και τις υλοποιήσεις θα παρέμενε γενικά ίδια ακόμα και αν η υλοποίηση γινόταν σήμερα - εφόσον συνεχίζουν να υφίστανται τα ίδια εργαλεία και εξοπλισμός με τις ίδιες προδιαγραφές με αυτές κατά την ανάπτυξη και ολοκλήρωση της πτυχιακής. Το μόνο που θα επέλεγα να αλλάξω θα ήταν να ενισχύσω τα virtualized χαρακτηριστικά του HA καθώς και την προσθήκη κατάλληλων hubs ή κατάλληλου hardware ώστε να έχει υποστήριξη και για συσκευές που βασίζονται στο πρωτόκολλο επικοινωνίας Zigbee. Καθώς το Zigbee αποτελεί εξαιρετική επιλογή για αισθητήρες και συσκευές που λειτουργούν με μπαταρίες ως μία τεχνολογία με μικρές απαιτήσεις από κατανάλωση ρεύματος. Ενώ ακόμα δημιουργούν ένα mesh δίκτυο που εξασφαλίζει αξιόπιστη επικοινωνία μεταξύ όλων των αντίστοιχων συσκευών.[35]

Η πτυχιακή αυτή σίγουρα άξιζε τον κόπο, καθώς πρόσφερε βαθύτερη - πρακτική - κατανόηση των τεχνολογιών IoT και μου ανέδειξε τρόπους έξυπνης δόμησης και αυτοματισμού ενός χώρου. Ταυτόχρονα, μου ενίσχυσε σημαντικά της δεξιότητές προγραμματισμού και επίλυσης προβλημάτων. Κυρίως όμως αποτέλεσε μια εμπειρία που ενίσχυσε την επιθυμία μου να ενισχύσω να ασχολούμαι με τον χώρο των έξυπνων κτηρίων μέσω του Home Assistant, βλέποντάς το όχι μόνο ως ένα πεδίο τεχνολογικής εφαρμογής, αλλά και ως ουσιαστικό βήμα προς πιο βιώσιμες και ενεργειακά αποδοτικές δομές.

Τέλος, η πτυχιακή αυτή σίγουρα αξίζει να συνεχιστεί και πιο συγκεκριμένα θα μπορούσε να της προστεθεί

1. Έλεγχος θέρμανσης και ψύξης.
2. Τοποθέτηση επιπλέον αισθητήρων (όπως εσωτερικής υγρασίας), με ίδιο πρωτόκολλο επικοινωνίας με αυτά της παραπάνω εγκατάστασης (WiFi) ή διαφορετικό (Zigbee, Z-Wave, Thread, Matter).
3. Αναγνώριση παρουσίας, προσώπων, κλπ.. μέσω μοντέλων AI (τεχνητής νοημοσύνης) όπως με το Frigate software [34].
4. Ενδιαφέρον θα είχε ίσως και η δημιουργία DIY αισθητήρων με χρήση ESP32 και το espHome integration του Home Assistant, με σκοπό δημιουργίας αισθητήρων που προσαρμόζονται απόλυτα στις ανάγκες μας, αντί να προσαρμόζουμε τις ανάγκες μας στις έτοιμες λύσεις εμπορικών συσκευών.

Βιβλιογραφία

- [1] <https://www.taxheaven.gr/news/> - Επιδότηση ηλεκτρικής ενέργειας για τον Δεκέμβριο - Αναλυτικά τα νέα μέτρα. [25/11/2022]
- [2] <https://www.iotforall.com/evolution-building-management-systems> - The Evolution of Building Management Systems: From Limited Control to a Fully Integrated Digital Twin | IoT For All. [02/12/2024]
- [3] <https://smartr.build/insights/building-management-systems-critical-component-efficient-operations> - Building management systems: A critical component for efficient operations.
- [4] <https://www.theverge.com/24087882/smart-home-hub-apple-google-alexa> - Smart home hubs: what they are and why you need one. [07/03/2024]
- [5] <https://github.com/home-assistant/assets/tree/master/logo> - Home Assistant logo. [05/10/2025]
- [6] <https://www.pocket-lint.com/what-is-home-assistant-how-does-it-work/> - What is Home Assistant, how does it work, and what do you need to get started?. [08/09/2023]
- [7] <https://www.home-assistant.io/installation/#about-installation-types> - Home Assistant installation types [05/10/2025]
- [8] <https://www.proxmox.com/en/about/company-details/media-kit> - Proxmox company logo. [05/10/2025]
- [9] <https://hometechhacker.com/mastering-shelly-products-with-home-assistant-a-comprehensive-guide/> - Mastering Shelly Products with Home Assistant: A Comprehensive Guide. [18/05/2025]
- [10] <https://kb.shelly.cloud/knowledge-base/shelly-motion-2> - Shelly Motion 2. [05/10/2025]
- [11] <https://kb.shelly.cloud/knowledge-base/shelly-plus-2pm> - Shelly Plus 2PM. [05/10/2025]

- [12] <https://kb.shelly.cloud/knowledge-base/shelly-plus-1pm> - Shelly Plus 1PM. [05/10/2025]
- [13] <https://greendigital.uth.gr/measurement/#pm-station> - GreenDigital Environmental measurements. [05/10/2025]
- [14] <https://www.shelly.com/products/shelly-wall-switch-with-1-button-white> - Shelly eshop, Shelly Wall Switch with 1 Button. [05/10/2025]
- [15] <https://www.home-assistant.io/integrations/shelly/> - Home Assistant Shelly integration. [05/10/2025]
- [16] https://community.shelly.cloud/gallery/image/512-plus-1pm-two-way-switching_enpng/ - Plus 1PM Two-way switching_en.png. [26/09/2024]
- [17] ShellyGuide, <https://www.youtube.com/watch?v=XzUUJGwJ5Zg>, 05 – Shelly Smart Control app - How to add your first Shelly device (YouTube βίντεο). [05/10/2025]
- [18] <https://www.nabucasa.com/about/> - Nabu Casa, About us. [06/10/2025]
- [19] <https://randomnerdtutorials.com/projects-home-automation/> - 30+ Home Automation Projects. [06/10/2025]
- [20] <https://www.home-assistant.io/installation/alternative> - Alternative Install Home Assistant Operating System. [06/10/2025]
- [21] <https://energytalk.co.za/t/shelly-or-sonoff/1366/9> - Shelly or Sonoff (Discussion). [12/2021 – 10/2022]
- [22] <https://aesop.iep.edu.gr/node/18302/2759> - Ηλεκτρολογία (Ε.Ε.) (Επαγγελματικό Λύκειο), Απλός ηλεκτρικός διακόπτης - εργαστηριακό μέρος. [06/10/2025]
- [23] <https://shelly-api-docs.shelly.cloud/gen1/#shelly-motion-2-settings-coiot> - Shelly Motion 2: /status. [06/10/2025]
- [24] <https://kb.shelly.cloud/knowledge-base/shelly-plus-2-pm-device-smart-control-cover-mode> - Shelly Plus 2 PM Device Smart Control - Cover Mode. [06/10/2025]
- [25] <https://shelly-api-docs.shelly.cloud/gen2/ComponentsAndServices/Cover/#calibration-kb> – Shelly Components and Services – Cover, Calibration KB. [07/10/2025]
- [26] <https://www.home-assistant.io/docs/tools/> Home – Documentation – Tools. [07/10/2025]

- [27] <https://www.thorlux.com/applications/education/classroom> - ThorluxLighting, Classrooms. [07/10/2025]
- [28] Ελληνική Δημοκρατία, ΕΦΗΜΕΡΙΣ ΤΗΣ ΚΥΒΕΡΝΗΣΕΩΣ, 221Α-94, Π.Δ 398/94, Αθήνα [17/12/1994] (Available online) <https://www.elinyae.gr/sites/default/files/2019-07/221A-94.pdf>.
- [30] <https://ypen.gov.gr/energeia/statistika-stoicheia/> - Ενέργεια-Στατιστικά στοιχεία, ΥΠΕΝ. [08/10/2025]
- [31] https://www.renkeer.com/dew-point/#elementor-toc__heading-anchor-7 - How to measure dew point? - Using Calculation Formulas. [05/06/2025]
- [32] <https://tsialos.gr/> - Συμπύκνωση υδρατμών στο τζάμι των παραθύρων στέγης (Blog). [07/12/2023]
- [33] <https://www.cordulus.com/glossary/dew-point> - What is a Dew point?. [30/09/2025]
- [34] <https://frigate.video/> - Frigate – Home. [08/10/2025]
- [35] <https://www.aqara.com/eu/blog-eu/what-is-zigbee-and-how-does-it-work-2/> - What Is Zigbee And How Does It Work?. [08/10/2025]
- [36] <https://www.aqara.com/eu/blog-eu/zigbee-vs-thread-vs-matter-whats-the-difference/> - Zigbee vs. Thread vs. Matter: What's The Difference?. [08/10/2025]
- [37] <https://floorplanner.com/> - Floorplanner.com. [08/10/2025]
- [38] <https://github.com/hassio-addons/addon-vscode> - Home Assistant Community Add-on: Studio Code Server. [08/10/2025]
- [39] <https://github.com/hassio-addons/addon-ssh> - Home Assistant Community Add-on: Advanced SSH & Web Terminal. [08/10/2025]
- [40] https://www.home-assistant.io/integrations/command_line/ - Command line integration. [08/10/2025]

Παράρτημα Α

Όλος ο κώδικας που αναπτύχθηκε ή/και χρησιμοποιήθηκε στην παρούσα πτυχιακή εργασία, είναι διαθέσιμος στο ακόλουθο GitHub αποθετήριο:

<https://github.com/konnostech81/Uth-Smart-Classroom-HomeAssistant-Thesis-2025>

Το αποθετήριο περιλαμβάνει:

- Μέρος του φακέλου *config* του Home Assistant.
- Τον *yaml/jinja2* κώδικα που χρησιμοποιήθηκε σε μεμονωμένα *templates* τύπου *helpers*.
- Τη λίστα με τα *Add-Ons* που χρησιμοποιήθηκαν