



**ΣΧΟΛΗ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ
ΠΜΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΔΙΚΤΥΩΝ**

ΜΕΤΑΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

του Θωμά Πεταρούδα του Δημητρίου

«Εφαρμογή INTERNET OF THINGS με RASPBERRY PI και PYTHON »

Επιβλέπων:

Ελευθέριος Στεργίου Αναπληρωτής Καθηγητής ΔΕΠ

Άρτα, Φεβρουάριος 2024

ΕΠΙΤΡΟΠΗ ΑΞΙΟΛΟΓΗΣΗΣ

Κωνσταντίνος Αγγέλης

Σπυριδούλα Μαργαρίτη

Ελευθέριος Στεργίου

ΕΥΧΑΡΙΣΤΙΕΣ

Θα ήθελα να ευχαριστήσω «ντε προφούντις» τον επιβλέποντα Καθηγητή μου, κύριο Ελευθέριο Στεργίου, Αναπληρωτή Καθηγητή ΔΕΠ, για την καθοδήγηση που μου προσέφερε και τον χρόνο που διέθεσε δίνοντάς μου πολύτιμες συμβουλές και χρήσιμες οδηγίες για την περάτωση της πτυχιακής εργασίας.

Θα ήθελα επίσης να ευχαριστήσω θερμά όλους τους καθηγητές του , του Τμήματος Πληροφορικής και Τηλεπικοινωνιών για την επιστημονική και τεχνολογική συγκρότησή μου.

Θερμές ευχαριστίες οφείλω στους γονείς μου, στους συγγενείς μου καθώς και στους φίλους μου για την ηθική στήριξη κατά τη διάρκεια των μεταπτυχιακών σπουδών μου.

Περιεχόμενα

ΠΙΝΑΚΑΣ ΕΙΚΟΝΩΝ	vii
ΠΕΡΙΛΗΨΗ	1
ABSTRACT	2
1. Εισαγωγή	3
2. Βιβλιογραφική αναφορά.....	4
2.1 Επισκόπηση συστημάτων ασφαλείας.....	4
2.2 Επισκόπηση συστημάτων παρακολούθησης με κάμερα	6
2.2.1 Μέθοδοι αναγνώρισης κάμερας.....	8
2.2.2 Ανίχνευση κίνησης στις κάμερες.....	8
3.Μεθοδολογία.....	11
3.1 Αρχιτεκτονική του συστήματος.....	11
3.2 Υλικά.....	12
3.3 Λογισμικό	17
3.4 Σχεδίαση και Υλοποίηση του συστήματος.....	19
3.5 Διαδικασία συλλογής δεδομένων	19
3.6 Εύρος και περιορισμοί	20
4.Σχεδίαση συστήματος και υλοποίηση	22
4.1 Ρύθμιση του Raspberry Pi και ενσωμάτωση της κάμερας	22
4.2 Ενσωμάτωση αισθητήρων και πρωτόκολλο επικοινωνίας.....	22
4.3 Σύνδεση στην πλατφόρμα και ενσωμάτωση και διασύνδεση του συστήματος.....	23
4.4 Λειτουργικότητα συστήματος.....	24
4.4.1 Υλοποίηση κώδικα ανίχνευσης	27
4.4.2 Υλοποίηση κώδικα αποστολής email	38
5.Συλλογή δεδομένων και ανάλυση.....	42
5.1 Διαδικασία συλλογής δεδομένων	42
5.2 Τεχνικές ανάλυσης των δεδομένων	43
5.3 Παρουσίαση των αποτελεσμάτων	45
6.Συμπεράσματα και μελλοντικές αλλαγές	51
6.1 Σύνοψη των ευρημάτων.....	51
6.2 Προτάσεις για μελλοντικές αλλαγές.....	52

ΒΙΒΛΙΟΓΡΑΦΙΚΕΣ ΑΝΑΦΟΡΕΣ.....	54
Παράρτημα Α.....	58
Παράρτημα Β.....	83
Παράρτημα Γ.....	92
Παράρτημα Δ.....	96

ΠΙΝΑΚΑΣ ΕΙΚΟΝΩΝ

Εικόνα 1 Σύστημα ασφαλείας	4
Εικόνα 2 Κάμερες παρακολούθησης	5
Εικόνα 3 Σύστημα διαχείρισης συναγερμού	7
Εικόνα 4 Ανίχνευση κίνησης.....	9
Εικόνα 5 Anker Powerconf C200 A3369	13
Εικόνα 6 Raspberry Pi 4B	14
Εικόνα 7 Raspberry Pi PSU	15
Εικόνα 8 GPIO connection pins.....	16
Εικόνα 9 PIR sensor	17
Εικόνα 10 SMTP protocol	20
Εικόνα 11 Cayenne platform	23
Εικόνα 12 Blynk platform	24
Εικόνα 13 Διάγραμμα ροής.....	26
Εικόνα 14 Gaussian analysis.....	44
Εικόνα 15 Cayenne platform	45
Εικόνα 16 Cayenne Dashboard	46
Εικόνα 17 Απομακρυσμένη παρακολούθηση και έλεγχος	48
Εικόνα 18 Προσθήκη motion sensor μέσω cayenne	50
Εικόνα 19 Επιλογές για Alert	51
Εικόνα 20 Προετοιμασία κάρτας	58
Εικόνα 21 Επιλογή λογισμικού	59
Εικόνα 22 Επιλογή κάρτας	60
Εικόνα 23 Εκκίνηση διαμόρφωσης	60
Εικόνα 24 Προειδοποιητικό μήνυμα διαμόρφωσης.....	61
Εικόνα 25 Πρόοδος διαμόρφωσης.....	62
Εικόνα 26 Μήνυμα επιτυχημένης διαμόρφωσης.....	62
Εικόνα 27 Τοποθέτηση κάρτας στο Raspberry Pi	63
Εικόνα 28 Ρυθμίσεις εγκατάστασης- Πρώτη εκκίνηση Raspberry Pi OS.....	64
Εικόνα 29 Ρυθμίσεις εγκατάστασης- Επιλογή Χώρας	65
Εικόνα 30 Ρυθμίσεις εγκατάστασης- Καθορισμός Χώρας	66
Εικόνα 31 Ρυθμίσεις εγκατάστασης- Καθορισμός Κωδικού.....	66
Εικόνα 32 Ρυθμίσεις εγκατάστασης- ρυθμίσεις οθόνης.....	67
Εικόνα 33 Ρυθμίσεις εγκατάστασης- ρυθμίσεις WiFi.....	68
Εικόνα 34 Ρυθμίσεις εγκατάστασης- Ενημέρωση λογισμικού	69
Εικόνα 35 Ρυθμίσεις εγκατάστασης- Ενημέρωση λογισμικού μπάρα προόδου	69
Εικόνα 36 Ρυθμίσεις εγκατάστασης- Ενημέρωση λογισμικού μήνυμα προόδου.....	70
Εικόνα 37 Ρυθμίσεις εγκατάστασης- Μήνυμα ολοκλήρωσης.....	71
Εικόνα 38 Preferences- Appearance settings	72

Εικόνα 39 Appearance settings- Desktop.....	73
Εικόνα 40 Appearance settings – Menu bar.....	73
Εικόνα 41 Appearance settings - System	74
Εικόνα 42 Appearance settings - Defaults.....	74
Εικόνα 43 Raspberry Pi Configuration- System	75
Εικόνα 44 Raspberry Pi Configuration- Interfaces.....	76
Εικόνα 45 Task Bar preferences - right click	76
Εικόνα 46 Panel preferences – Panel applets	77
Εικόνα 47 Panel preferences – add plug in CPU TEMP	78
Εικόνα 48 Panel preferences- add plug in Keyboard layout.....	79
Εικόνα 49 Keyboard layout Oprion	79
Εικόνα 50 Keyboard layout change language	80
Εικόνα 51 Keyboard layout- change layout option	81
Εικόνα 52 Programming- Thonny Python	82
Εικόνα 53 Cayenne platform - Notifications	96
Εικόνα 54 Cayenne platform - Notification Settings	97
Εικόνα 55 Cayenne platform - Triggers.....	97
Εικόνα 56 Cayenne platform - Χρονοπρογραμματισμός	98
Εικόνα 57 Cayenne platform - settings.....	98
Εικόνα 58 Cayenne platform - Προγραμματισμός προβολών	98
Εικόνα 59 Cayenne platform - Ροές Δεδομένων	99
Εικόνα 60 Cayenne platform - Προβολή δεδομένων	99
Εικόνα 61 Παρακολούθηση τοποθεσίας.....	100
Εικόνα 62 Τύποι παρακολούθησης στοιχείων	100
Εικόνα 63 Οπτικοποίηση στοιχείων.....	100
Εικόνα 64 Ενσωμάτωση συσκευών στην πλατφόρμα Cayenne	101
Εικόνα 65 Προσαρμοσμένα widget.....	101
Εικόνα 66 Sketch Files	102
Εικόνα 67 MQTT API	102
Εικόνα 68 Τύποι δεδομένων.....	103
Εικόνα 69 Raspberry pi 4	103
Εικόνα 70 GPIO θήρες του Raspberry Pi 4.....	105
Εικόνα 71 HC-SR501 PIR sensor	106
Εικόνα 72 HC-SR501PIR sensor	107
Εικόνα 73 Cayenne Features	108
Εικόνα 74 Cayenne platform - Εγκατάσταση απαραίτητων εργαλείων	108
Εικόνα 75 Cayenne platform - Διαδικασία εγκατάστασης.....	109
Εικόνα 76 Cayenne platform - Ολοκλήρωση εγκατάστασης.....	110
Εικόνα 77 Raspberry Pi Settings.....	110
Εικόνα 78 Raspberry Pi Settings (2).....	111

Εικόνα 79 Motion detection sensors	112
Εικόνα 80 Motion detection sensor settings.....	113
Εικόνα 81 Cayenne platform - Notifications	114
Εικόνα 82 Cayenne platform - Trigger selection	114
Εικόνα 83 Cayenne platform - email notification	115

ΠΕΡΙΛΗΨΗ

Η παρούσα εργασία περιλαμβάνει έρευνα σχετικά με την αποτελεσματικότητα και την ανταπόκριση ενός συστήματος παρακολούθησης με χρήση κάμερας που αναπτύσσεται σε μια καθορισμένη γεωγραφική περιοχή. Το υπό έλεγχο σύστημα αναλαμβάνει συνεχή παρακολούθηση της εν λόγω περιοχής, αξιοποιώντας αλγόριθμους ανίχνευσης κίνησης για να διακρίνει τις αλλαγές στο οπτικό πεδίο που καταγράφονται από τη συσκευή της κάμερας. Με την ανίχνευση κίνησης, αποστέλλει αμέσως ένα μήνυμα ηλεκτρονικού ταχυδρομείου (email) που περιέχει την αντιπαράθεση εικόνας που διακρίνεται, γεγονός που υποδηλώνει την ασυμφωνία μεταξύ του προηγούμενου και των επόμενων καρέ, ενδεικτικό της κίνησης. Ταυτόχρονα, το σύστημα αρχειοθετεί το στιγμιότυπο εικόνας που έλαβε μέσα στο μέσο αποθήκευσης μιας υπολογιστικής πλατφόρμας Raspberry Pi για μετέπειτα ανάλυση και αρχειακούς σκοπούς.

ABSTRACT

This study encompasses an inquiry into the efficacy and responsiveness of a camera surveillance infrastructure deployed within a designated geographic region. The system under scrutiny undertakes continuous monitoring of said area, leveraging motion detection algorithms to discern alterations within the visual field captured by the camera apparatus. Upon detection of movement, it promptly dispatches an email containing the discerned image juxtaposition denoting the discrepancy between antecedent and subsequent frames, indicative of motion. Simultaneously, the system archives the received image snapshot within the storage medium of a Raspberry Pi computing platform for subsequent analysis and archival purposes.

1. Εισαγωγή

Η έλευση των εξελιγμένων τεχνολογιών επιτήρησης έχει επιφέρει αξιοσημείωτες καινοτομίες στα ζητήματα αναφορικά με τα συστήματα ασφαλείας, προσφέροντας εξέχουσα επαγρύπνηση και δυνατότητες παρακολούθησης σε διαφορετικά περιβάλλοντα. Σε αυτό το πλαίσιο, η παρούσα μελέτη διερευνά την αξιοπιστία και την ανταπόκριση ενός συστήματος παρακολούθησης κάμερας που αναπτύσσεται σε μια συγκεκριμένη γεωγραφική περιοχή.

Κεντρικό στοιχείο της έρευνάς μας είναι η χρήση του Raspberry Pi, μιας ευέλικτης και συμπαγούς υπολογιστικής πλατφόρμας που έχει κερδίσει ευρεία αναγνώριση για την οικονομική προσιτότητα, την προσβασιμότητα και την προσαρμοστικότητά της σε διάφορους τεχνολογικούς τομείς. Ως εκπαιδευτικό εργαλείο για την προώθηση της εκπαίδευσης στην επιστήμη των υπολογιστών και της επάρκειας προγραμματισμού, το Raspberry Pi έχει εξελιχθεί σε βασικό συστατικό σε πολυάριθμες προσπάθειες έρευνας και ανάπτυξης, ιδιαίτερα στον τομέα των ενσωματωμένων συστημάτων και των εφαρμογών Internet of Things (IoT)[1], [2].

Η ώθηση πίσω από αυτήν την έρευνα πηγάζει από την επιτακτική ανάγκη αντιμετώπισης του εξελισσόμενου τοπίου των προκλήσεων ασφαλείας, όπου οι παραδοσιακές μεθοδολογίες επιτήρησης συχνά υστερούν στην παροχή έγκαιρων και ακριβών γνώσεων για δραστηριότητες που παρουσιάζουν ιδιαιτερότητες. Συγκεκριμένα, η μελέτη επιδιώκει να εξετάσει την αποτελεσματικότητα του συστήματος επιτήρησης όσον αφορά την ικανότητά του να εντοπίζει και να ειδοποιεί τα ενδιαφερόμενα μέρη για αντιληπτές απειλές ή διαταραχές εντός της παρακολουθούμενης περιοχής [3], [4].

Μία από τις κύριες ανησυχίες αφορά την απόκριση του συστήματος σε ενημερώσεις και συναγερμούς, κρίσιμα στοιχεία για τη διασφάλιση της ακεραιότητας και της αποτελεσματικότητας της συσκευής ασφαλείας. Επιπλέον, η μελέτη προσπαθεί να αξιολογήσει τη σκοπιμότητα χρήσης μιας μοναδικής κάμερας για ανίχνευση κίνησης, βελτιστοποιώντας έτσι τη χρήση των πόρων και ελαχιστοποιώντας την πολυπλοκότητα της υποδομής [5].

Σε αυτό το πλαίσιο, οι στόχοι της μελέτης είναι διπλοί. Πρώτον, να εξακριβωθεί η εγκυρότητα και η ορθότητα του συστήματος επιτήρησης για τον ακριβή εντοπισμό και την απόκριση σε αντιληπτές ανωμαλίες εντός της παρακολουθούμενης περιοχής. Δεύτερον, για την αξιολόγηση της χρονικής απόδοσης του συστήματος, ιδιαίτερα για τον εντοπισμό τυχόν καθυστερήσεων ή ζητημάτων λανθάνοντος χρόνου που μπορεί να εμποδίσουν την απόκρισή του σε πραγματικό χρόνο [6].

Εξετάζοντας αυτές τις βασικές πτυχές, αυτή η μελέτη προσπαθεί να συνεισφέρει εμπειρικές γνώσεις σχετικά με την αποτελεσματικότητα και τη λειτουργική δυναμική των

σύγχρονων συστημάτων επιτήρησης, ενημερώνοντας έτσι τις μελλοντικές εξελίξεις και βελτιστοποιήσεις στον τομέα της ασφάλειας και των τεχνολογιών παρακολούθησης.

2. Βιβλιογραφική αναφορά

2.1 Επισκόπηση συστημάτων ασφαλείας

Τα συστήματα ασφαλείας διαδραματίζουν κρίσιμο ρόλο στη διαφύλαξη περιουσιακών στοιχείων, ακινήτων και ατόμων από πιθανές απειλές και εισβολές. Καθώς οι τεχνολογικές εξελίξεις συνεχίζουν να επαναπροσδιορίζουν το τοπίο των λύσεων ασφαλείας, καθίσταται επιτακτική η κατανόηση των θεμελιωδών αρχών και στοιχείων που στηρίζουν τα σύγχρονα συστήματα ασφαλείας [6].

Στον πυρήνα του, ένα σύστημα ασφαλείας περιλαμβάνει ένα δίκτυο διασυνδεδεμένων συσκευών, αισθητήρων και εφαρμογών λογισμικού που έχουν σχεδιαστεί για να ανιχνεύουν, να αποτρέπουν και να ανταποκρίνονται σε παραβιάσεις ασφάλειας και απόπειρες μη εξουσιοδοτημένης πρόσβασης. Αυτά τα συστήματα έχουν σχεδιαστεί για να παρέχουν ολοκληρωμένη κάλυψη σε διάφορους τομείς, συμπεριλαμβανομένων οικιστικών, εμπορικών, βιομηχανικών και θεσμικών ρυθμίσεων [7], [8].



Εικόνα 1 Σύστημα ασφαλείας

Τα βασικά στοιχεία των συστημάτων ασφαλείας περιλαμβάνουν αισθητήρες ανίχνευσης εισβολής, κάμερες παρακολούθησης, μηχανισμούς ελέγχου πρόσβασης, συστήματα συναγερμού και σταθμούς παρακολούθησης. Αισθητήρες ανίχνευσης εισβολής, όπως ανιχνευτές κίνησης, αισθητήρες πόρτας/παραθύρου και ανιχνευτές θραύσης γυαλιού, χρησιμεύουν ως η πρώτη γραμμή άμυνας ανιχνεύοντας μη εξουσιοδοτημένες απόπειρες εισόδου ή ύποπτες δραστηριότητες σε μια ασφαλή περιοχή [7].

Οι κάμερες παρακολούθησης αντιπροσωπεύουν ένα άλλο αναπόσπαστο στοιχείο των συστημάτων ασφαλείας, προσφέροντας δυνατότητες παρακολούθησης και οπτικής επαλήθευσης σε πραγματικό χρόνο. Τα προηγμένα συστήματα κάμερας αξιοποιούν τις τεχνολογίες απεικόνισης υψηλής ευκρίνειας, τις λειτουργίες νυχτερινής όρασης με υπέρυθρη ακτινοβολία και τις λειτουργίες pan-tilt-zoom για να παρέχουν ολοκληρωμένη κάλυψη και επίγνωση της κατάστασης σε διάφορα περιβάλλοντα [9].



Εικόνα 2 Κάμερες παρακολούθησης

Μηχανισμοί ελέγχου πρόσβασης, συμπεριλαμβανομένων συσκευών ανάγνωσης καρτών-κλειδιών, βιομετρικών σαρωτών και ηλεκτρονικών κλειδαριών, ρυθμίζουν τα σημεία εισόδου και εξόδου, διασφαλίζοντας ότι το εξουσιοδοτημένο προσωπικό έχει πρόσβαση ενώ περιορίζει τα μη εξουσιοδοτημένα άτομα. Τα συστήματα συναγερμού χρησιμεύουν ως ηχητικά και οπτικά αποτρεπτικά μέσα, εκπέμποντας ειδοποιήσεις και ειδοποιήσεις ως απόκριση σε παραβιάσεις ασφαλείας ή προκαθορισμένα ερεθίσματα.

Οι κεντρικοί σταθμοί παρακολούθησης χρησιμεύουν ως το νευρικό κέντρο των συστημάτων ασφαλείας, συγκεντρώνοντας δεδομένα από διάφορους αισθητήρες και κάμερες και διευκολύνοντας την ανάλυση σε πραγματικό χρόνο, τη λήψη αποφάσεων και τον συντονισμό απόκρισης. Μέσω της ενοποίησης με πλατφόρμες που βασίζονται σε cloud και εφαρμογές για κινητές συσκευές, τα συστήματα ασφαλείας επιτρέπουν την απομακρυσμένη παρακολούθηση, διαχείριση και έλεγχο, δίνοντας τη δυνατότητα στους χρήστες να επιβλέπουν τις λειτουργίες ασφαλείας από οπουδήποτε και ανά πάσα στιγμή.

Συμπερασματικά, τα συστήματα ασφαλείας αντιπροσωπεύουν ένα πολύπλευρο οικοσύστημα διασυνδεδεμένων τεχνολογιών και λειτουργιών με στόχο τον μετριασμό των κινδύνων και την ενίσχυση της ασφάλειας. Με την κατανόηση των αρχών και των στοιχείων των συστημάτων ασφαλείας, οι ενδιαφερόμενοι μπορούν να λάβουν τεκμηριωμένες αποφάσεις σχετικά με το σχεδιασμό, την υλοποίηση και τη διαχείριση ισχυρών λύσεων ασφαλείας προσαρμοσμένων στις συγκεκριμένες ανάγκες και απαιτήσεις τους.

2.2 Επισκόπηση συστημάτων παρακολούθησης με κάμερα

Τα συστήματα παρακολούθησης κάμερας αποτελούν ένα πολύ σημαντικό κομμάτι της σύγχρονης υποδομής ασφάλειας, παρέχοντας δυνατότητες οπτικής παρακολούθησης, καταγραφής και ανάλυσης σε διάφορα περιβάλλοντα. Καθώς η επικράτηση των τεχνολογιών επιτήρησης συνεχίζει να αυξάνεται, καθίσταται επιτακτική ανάγκη να διερευνηθούν οι θεμελιώδεις αρχές και λειτουργίες που ορίζουν τα συστήματα παρακολούθησης με κάμερα [10].

Στην ουσία, ένα σύστημα παρακολούθησης κάμερας περιλαμβάνει ένα δίκτυο διασυνδεδεμένων καμερών που αναπτύσσονται στρατηγικά για να καταγράψουν και να παρακολουθούν δραστηριότητες σε μια καθορισμένη περιοχή. Αυτές οι κάμερες αξιοποιούν προηγμένες τεχνολογίες απεικόνισης, συμπεριλαμβανομένης της ανάλυσης υψηλής ευκρίνειας, του μεγάλου δυναμικού εύρους και της ευαισθησίας χαμηλού φωτισμού, για να προσφέρουν καθαρό, λεπτομερές υλικό κάτω από διάφορες συνθήκες φωτισμού.

Τα συστήματα παρακολούθησης σε κάμερα εξυπηρετούν πολλούς σκοπούς που κυμαίνονται από την αποτροπή εγκληματικής δραστηριότητας και βανδαλισμού μέχρι την ενίσχυση της επίγνωσης της κατάστασης και τη διευκόλυνση των ερευνών. Σε εμπορικά και βιομηχανικά περιβάλλοντα οι κάμερες παρακολούθησης διαδραματίζουν σημαντικό ρόλο στην παρακολούθηση των λειτουργιών, στη διασφάλιση της συμμόρφωσης με τα πρωτόκολλα ασφαλείας και στην προστασία περιουσιακών στοιχείων και πόρων από κλοπή ή κακή χρήση.

Τα βασικά χαρακτηριστικά των συστημάτων παρακολούθησης κάμερας περιλαμβάνουν δυνατότητες ανίχνευσης κίνησης, λειτουργίες απομακρυσμένης

παρακολούθησης και ελέγχου και μηχανισμούς αρχειοθέτησης και ανάκτησης για καταγεγραμμένα πλάνα. Οι αλγόριθμοι ανίχνευσης κίνησης επιτρέπουν στις κάμερες να ανιχνεύουν και να παρακολουθούν αυτόματα την κίνηση εντός του οπτικού τους πεδίου, ενεργοποιώντας ειδοποιήσεις και ειδοποιήσεις ως απόκριση σε ύποπτες δραστηριότητες ή προκαθορισμένες ενεργοποιήσεις [11].



Εικόνα 3 Σύστημα διαχείρισης συναγερμού

Οι λειτουργίες απομακρυσμένης παρακολούθησης και ελέγχου δίνουν στους χρήστες πρόσβαση σε ζωντανές ροές κάμερας, προσαρμόζουν τις ρυθμίσεις της κάμερας και ελέγχουν εγγεγραμμένο υλικό από οποιαδήποτε συσκευή με δυνατότητα σύνδεσης στο διαδίκτυο, διευκολύνοντας την επίγνωση της κατάστασης και τον συντονισμό απόκρισης σε πραγματικό χρόνο. Οι λύσεις αποθήκευσης που βασίζονται στο cloud παρέχουν επεκτάσιμες και ασφαλείς επιλογές αποθήκευσης για αρχειοθετημένα πλάνα, διασφαλίζοντας την ακεραιότητα των δεδομένων και την προσβασιμότητα για εκτεταμένες περιόδους [12].

Συνοπτικά, τα συστήματα παρακολούθησης κάμερας αντιπροσωπεύουν ένα κρίσιμο στοιχείο μιας ολοκληρωμένης υποδομής ασφάλειας, επιτρέποντας σε οργανισμούς και άτομα να παρακολουθούν, να αναλύουν και να ανταποκρίνονται αποτελεσματικά σε απειλές και συμβάντα για την ασφάλεια. Αξιοποιώντας τις δυνατότητες προηγμένων τεχνολογιών απεικόνισης και λειτουργιών απομακρυσμένης παρακολούθησης, τα συστήματα παρακολούθησης κάμερας ενδυναμώνουν τους ενδιαφερόμενους φορείς να

βελτιώσουν την ασφάλεια, να μετριάσουν τους κινδύνους και να προστατεύσουν περιουσιακά στοιχεία και πόρους σε ένα συνεχώς εξελισσόμενο τοπίο ασφάλειας.

2.2.1 Μέθοδοι αναγνώρισης κάμερας

Οι μέθοδοι αναγνώρισης κάμερας ενσωματώνουν εξελιγμένους αλγορίθμους και υπολογιστικά πλαίσια που έχουν σχεδιαστεί για να διακρίνουν και να παρακολουθούν σχολαστικά την κίνηση μέσα στο εκτεταμένο πεδίο του οπτικού πεδίου μιας κάμερας. Αυτό το τεχνολογικό επίτευγμα χρησιμεύει ως βάση των συστημάτων επιτήρησης, των πλαισίων ασφαλείας και των πλατφορμών ανίχνευσης κίνησης, δημιουργώντας μια αλλαγή παραδείγματος στις δυνατότητες παρακολούθησης σε πραγματικό χρόνο.

Είναι θεμελιώδους σημασίας η μέθοδος αναγνώρισης κάμερας, η οποία βρίσκεται στην περίπλοκη ανάλυση των ροών βίντεο που καταγράφονται από τη συσκευή της κάμερας, όπου οι αποχρώσεις στη σκηνή εξετάζονται σχολαστικά σε διαδοχικά καρέ. Μέσω μιας σειράς μεθοδολογιών αιχμής όπως η αφαίρεση φόντου, η ανάλυση οπτικής ροής και οι αλγόριθμοι ανίχνευσης αντικειμένων, το σύστημα διακρίνει ανεπαίσθητες αλλαγές στο οπτικό τοπίο, οριοθετώντας έτσι τα μοτίβα κίνησης με αξιοσημείωτη ακρίβεια [13], .

Η επακόλουθη ανίχνευση κίνησης προαναγγέλλει επακόλουθες προκαθορισμένες ενέργειες ή ειδοποιήσεις, σχολαστικά βαθμονομημένων στις ανάγκες του παρακολουθούμενου περιβάλλοντος. Είτε πρόκειται για έναρξη εγγραφής βίντεο, ταχεία διάδοση ειδοποιήσεων ή για αλληλεπίδραση με πρωτόκολλα συναγερμού, το σύστημα πλοηγείται απρόσκοπτα στη δυναμική αλληλεπίδραση μεταξύ συμβάντων κίνησης και ενεργειών απόκρισης.

Η εξέλιξη της αναγνώρισης κίνησης από τις κάμερες αποτελεί απόδειξη για τις συνεργιστικές προόδους στην όραση υπολογιστών και τα παραδείγματα μηχανικής μάθησης. Αυτές οι μετασηματιστικές τεχνολογίες προσφέρουν πρωτοφανή επίπεδα ακρίβειας και ανθεκτικότητας στην ανίχνευση κίνησης, ακόμη και σε ένα περιβάλλον περίπλοκων και δυναμικών περιβαλλοντικών ρυθμίσεων. Επιπλέον, η συμβολή της τεχνολογικής αποτελεσματικότητας και των καινοτομιών υλικού, που επισημαίνεται από κάμερες υψηλής ανάλυσης και εξελιγμένες μονάδες επεξεργασίας, ενισχύει την αποτελεσματικότητα και την ευελιξία των συστημάτων αναγνώρισης κίνησης σε αξιοσημείωτα επίπεδα.

2.2.2 Ανίχνευση κίνησης στις κάμερες

Η ανίχνευση κίνησης στις κάμερες αναφέρεται στην τεχνολογία που επιτρέπει σε μια κάμερα να εντοπίζει και να καταγράφει καρέ βίντεο όπου υπάρχει κίνηση εντός του

οπτικού της πεδίου. Αυτή η λειτουργία χρησιμοποιείται συνήθως σε συστήματα ασφαλείας, οικιακή επιτήρηση, παρακολούθηση άγριων ζώων και διάφορες άλλες εφαρμογές [14], [15].



Εικόνα 4 Ανίχνευση κίνησης

Ακολουθεί ένα γενικό περίγραμμα του τρόπου λειτουργίας της ανίχνευσης κίνησης στις κάμερες:

1. Σύγκριση καρτέ

Η κάμερα καταγράφει διαδοχικά καρτέ της τροφοδοσίας βίντεο. Οι αλγόριθμοι ανίχνευσης κίνησης συγκρίνουν αυτά τα καρτέ για να εντοπίσουν τις αλλαγές μεταξύ τους.

2. Αλλαγές εικονοστοιχείων

Αναλύονται οι αλλαγές στα εικονοστοιχεία μεταξύ διαδοχικών καρτέ. Οποιαδήποτε σημαντική μεταβολή μεταξύ των καρτέ υποδεικνύει πιθανή κίνηση.

3. Ρύθμιση κατώφλιου

Ορίζεται ένα κατώφλι για να καθορισθεί ποιο επίπεδο μεταβολής στις τιμές των εικονοστοιχείων συνιστά κίνηση. Εάν η διαφορά μεταξύ των καρτέ ξεπεράσει αυτό το κατώφλι, επισημαίνεται ως κίνηση.

4. *Περιοχές ενδιαφέροντος (ROI)*

Ορισμένα προηγμένα συστήματα επιτρέπουν τον ορισμό συγκεκριμένων περιοχών εντός της προβολής της κάμερας (ROI), όπου η ανίχνευση κίνησης είναι πιο κρίσιμη. Για παράδειγμα, μπορείτε να εστιάσετε την ανίχνευση κίνησης σε εισόδους ή πολύτιμα αντικείμενα.

5. *Ειδοποίηση ή εγγραφή*

Όταν ανιχνεύεται κίνηση, το σύστημα κάμερας μπορεί να ενεργοποιήσει διάφορες ενέργειες, όπως αποστολή ειδοποιήσεων, ενεργοποίηση συναγερμού, έναρξη εγγραφής ή λήψη στιγμιότυπων.

6. *Ρυθμιζόμενη ευαισθησία*

Πολλά συστήματα προσφέρουν τη δυνατότητα ρύθμισης των επιπέδων ευαισθησίας. Αυτή η λειτουργία βοηθά στην ελαχιστοποίηση των ψευδών συναγερμών που προκαλούνται από μικρές κινήσεις, όπως ταλαντευόμενα κλαδιά ή διερχόμενα οχήματα.

7. *Προηγμένες τεχνικές*

Ορισμένα εξελιγμένα συστήματα χρησιμοποιούν προηγμένους αλγορίθμους, όπως η μηχανική μάθηση ή η τεχνητή νοημοσύνη, για τη βελτίωση της ακρίβειας, μειώνοντας τα ψευδώς θετικά και αρνητικά αποτελέσματα.

Υπάρχουν διάφορες μέθοδοι που χρησιμοποιούνται για την ανίχνευση κίνησης:

- Διαφοροποίηση καρέ: Συγκρίνει τις τιμές των εικονοστοιχείων μεταξύ των καρέ για τον εντοπισμό αλλαγών.
- Αφαίρεση φόντου: Απομονώνει το φόντο από τα κινούμενα αντικείμενα.
- Ανάλυση οπτικής ροής: Παρακολουθεί την κίνηση παρατηρώντας τη ροή των εικονοστοιχείων μεταξύ των καρέ.
- Μοντέλα βαθιάς μάθησης: Χρησιμοποιούν νευρωνικά δίκτυα για την αναγνώριση μοτίβων που σχετίζονται με την κίνηση.

Οι περισσότερες σύγχρονες κάμερες ασφαλείας και συστήματα παρακολούθησης διαθέτουν δυνατότητες ανίχνευσης κίνησης. Αυτά τα συστήματα προσφέρουν συχνά προσαρμόσιμες ρυθμίσεις για την προσαρμογή της ευαισθησίας, τον καθορισμό ζωνών ανίχνευσης και την ενσωμάτωση με άλλες συσκευές ή λογισμικό για ειδοποιήσεις ή καταγραφή.

Είναι σημαντικό να προσαρμόζετε αυτές τις ρυθμίσεις ανάλογα με το περιβάλλον, ώστε να ελαχιστοποιείτε τους ψευδείς συναγεμμούς, διασφαλίζοντας παράλληλα την αξιόπιστη καταγραφή των πραγματικών συμβάντων κίνησης.

3. Μεθοδολογία

Το κεφάλαιο της μεθοδολογίας περιγράφει τη δομημένη προσέγγιση που καθοδηγεί τη διερεύνηση της αξιοπιστίας και της λειτουργικότητας του συστήματος παρακολούθησης κάμερας. Παρέχει ένα μεθοδολογικό πλαίσιο, που περιγράφει τις συστηματικές διαδικασίες συλλογής, ανάλυσης και ερμηνείας δεδομένων με στόχο την ολοκληρωμένη αξιολόγηση των μετρήσεων απόδοσης και της λειτουργικής δυναμικής του συστήματος εντός του καθορισμένου γεωγραφικού πλαισίου.

3.1 Αρχιτεκτονική του συστήματος

Η αρχιτεκτονική του συστήματος της υποδομής επιτήρησης κάμερας βασίζεται στις ισχυρές δυνατότητες και την ευελιξία της υπολογιστικής πλατφόρμας Raspberry Pi. Στον πυρήνα του, το Raspberry Pi λειτουργεί ως η κεντρική μονάδα επεξεργασίας που ελέγχει πολλαπλές λειτουργίες επιτήρησης, αξιοποιώντας τη συμπαγή μορφή και την υπολογιστική του ικανότητα για να διευκολύνει τις λειτουργίες παρακολούθησης και ανίχνευσης σε πραγματικό χρόνο [16].

Η αρχιτεκτονική περιλαμβάνει ένα πρότυπο σχεδίασης αρθρωτών, που περιλαμβάνει διασυνδεδεμένα στοιχεία υλικού και λογισμικού βελτιστοποιημένα για απρόσκοπτη ενοποίηση και αποτελεσματική χρήση πόρων. Η πλακέτα Raspberry Pi χρησιμεύει ως το κύριο υπόστρωμα υλικού, με σύστημα Broadcom system-on-chip (SoC) που ενσωματώνει έναν τετραπύρρηνο επεξεργαστή ARM Cortex-A72, που συμπληρώνεται από ενσωματωμένη μνήμη, περιφερειακές διεπαφές και δυνατότητες επέκτασης [17].

Διευκολύνοντας την αισθητηριακή είσοδο και την απόκτηση δεδομένων, η αρχιτεκτονική ενσωματώνει κάμερες υψηλής ανάλυσης που συνδέονται με την πλακέτα Raspberry Pi μέσω τυποποιημένων διεπαφών όπως CSI (Camera Serial Interface), επιτρέποντας τη λήψη και την επεξεργασία ροών οπτικών δεδομένων σε πραγματικό χρόνο. [18] Επιπλέον, το σύστημα ενσωματώνει αισθητήρες κίνησης και περιβαλλοντικούς αισθητήρες για να αυξήσει την επίγνωση της κατάστασης και να βελτιώσει τις δυνατότητες ανίχνευσης της υποδομής επιτήρησης.

Οι λειτουργίες αποθήκευσης και αρχειοθέτησης δεδομένων διευκολύνονται μέσω της ενσωμάτωσης μέσων αποθήκευσης, συμπεριλαμβανομένων των καρτών microSD ή των μονάδων flash USB, επιτρέποντας τη διατήρηση και ανάκτηση των καταγεγραμμένων

εικόνων και δεδομένων αισθητήρων για ανάλυση και αποδεικτικούς σκοπούς. Οι δυνατότητες δικτύωσης πραγματοποιούνται μέσω επιλογών σύνδεσης Ethernet ή ασύρματης σύνδεσης, διευκολύνοντας την απομακρυσμένη πρόσβαση και τον έλεγχο του συστήματος επιτήρησης μέσω τοπικών δικτύων ή του διαδικτύου.

Η στοίβα λογισμικού περιλαμβάνει εξαρτήματα λογισμικού ανοιχτού κώδικα και αποκλειστικού λογισμικού, συμπεριλαμβανομένου του λειτουργικού συστήματος Raspbian, προσαρμοσμένα για ενσωματωμένα υπολογιστικά περιβάλλοντα και βελτιστοποιημένα για την αρχιτεκτονική Raspberry Pi [19]. Λογισμικό εφαρμογών, συμπεριλαμβανομένων αλγορίθμων ανίχνευσης κίνησης, ρουτίνες επεξεργασίας εικόνας και πρωτοκόλλων επικοινωνίας δικτύου, υλοποιούνται για τη διευκόλυνση της ανάλυσης σε πραγματικό χρόνο, της ανίχνευσης συμβάντων και των λειτουργιών ειδοποίησης.

Συνολικά, η αρχιτεκτονική του συστήματος ενσωματώνει ένα καταναμημένο υπολογιστικό παράδειγμα, όπου το Raspberry Pi χρησιμεύει ως πλέγμα υπολογισμού και ελέγχου, ενορχηστρώνοντας την απρόσκοπτη ενοποίηση στοιχείων υλικού και λογισμικού για την υλοποίηση μιας ισχυρής και ανταποκρινόμενης υποδομής παρακολούθησης κάμερας προσαρμοσμένη στις ανάγκες των σύγχρονων εφαρμογών ασφαλείας .

3.2 Υλικά

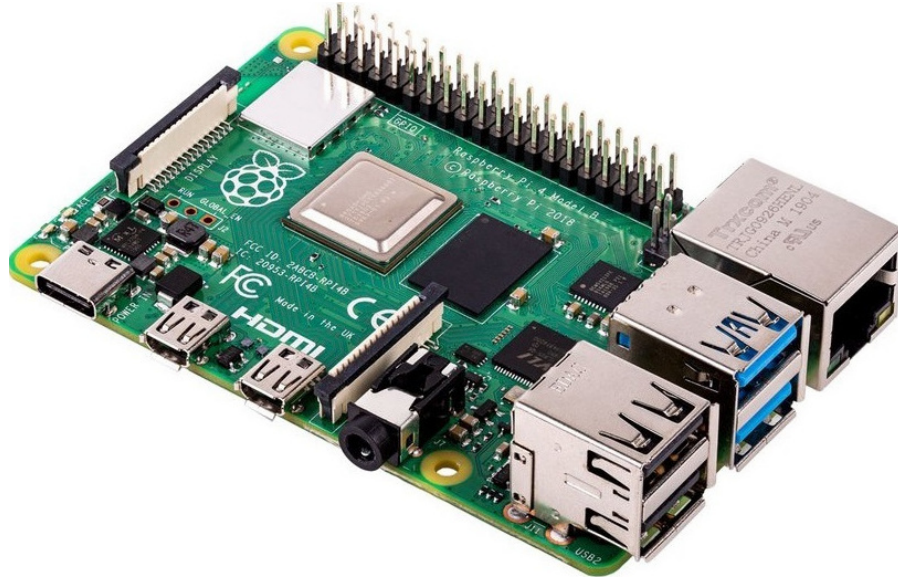
Το σύστημα παρακολούθησης κάμερας απαιτεί μια ολοκληρωμένη σειρά υλικών, καθένα από τα οποία διαδραματίζει κρίσιμο ρόλο στη λειτουργική ανάπτυξη και λειτουργικότητα της υποδομής. Τα ακόλουθα εξαρτήματα αποτελούν τα βασικά υλικά που απαιτούνται για τη συναρμολόγηση και τη διαμόρφωση του συστήματος:

- **Web Camera:** Μια κάμερα web υψηλής ανάλυσης χρησιμεύει ως η κύρια συσκευή εισαγωγής αισθητηρίων, διευκολύνοντας τη λήψη και τη μετάδοση ροών οπτικών δεδομένων εντός της παρακολουθούμενης περιοχής. Οι προδιαγραφές της κάμερας, όπως η ανάλυση, ο ρυθμός καρέ και το οπτικό πεδίο, επηρεάζουν τις δυνατότητες ανίχνευσης και την ποιότητα απεικόνισης του συστήματος [20].



Εικόνα. 5 Anker Powerconf C200 A3369

- Raspberry Pi 4B: Ο υπολογιστής μονής πλακέτας Raspberry Pi 4B χρησιμεύει ως κεντρική μονάδα επεξεργασίας και κέντρο ελέγχου της υποδομής επιτήρησης. Διαθέτοντας τετραπύρηνο επεξεργαστή ARM Cortex-A72, ενσωματωμένη μνήμη και περιφερειακές διεπαφές, το Raspberry Pi 4B παρέχει την υπολογιστική ικανότητα που είναι απαραίτητη για την επεξεργασία δεδομένων σε πραγματικό χρόνο, την ανάλυση και την επικοινωνία δικτύου [21].



Εικόνα 6 Raspberry Pi 4B

- Τροφοδοτικό: Μια αξιόπιστη μονάδα τροφοδοσίας είναι απαραίτητη για την αδιάλειπτη παροχή ρεύματος στο Raspberry Pi 4B και στα βοηθητικά εξαρτήματα. Η μονάδα τροφοδοσίας πρέπει να συμμορφώνεται με τις απαιτήσεις τάσης και ρεύματος που καθορίζονται από το Raspberry Pi Foundation για να διασφαλιστεί η βέλτιστη απόδοση και η σταθερότητα του συστήματος [21].



Εικόνα 7 Raspberry Pi PSU

- **Οθόνη:** Μια οθόνη υψηλής ευκρίνειας χρησιμεύει ως η οπτική διεπαφή εξόδου για τη διαμόρφωση και την παρακολούθηση της λειτουργίας του συστήματος επιτήρησης. Η οθόνη διευκολύνει την αλληλεπίδραση του χρήστη και τις ρυθμίσεις παραμέτρων, παρέχοντας ανατροφοδότηση σε πραγματικό χρόνο για την κατάσταση του συστήματος και τις ειδοποιήσεις.
- **Καλώδιο HDMI Mini:** Ένα καλώδιο HDMI mini επιτρέπει την απρόσκοπτη σύνδεση μεταξύ του Raspberry Pi 4B και της οθόνης, διευκολύνοντας την έξοδο βίντεο υψηλής ευκρίνειας και τον συγχρονισμό της οθόνης.
- **Πληκτρολόγιο και ποντίκι:** Οι συσκευές εισόδου, όπως το πληκτρολόγιο και το ποντίκι, είναι απαραίτητες για την αλληλεπίδραση με τον χρήστη και τη διαμόρφωση του συστήματος. Αυτά τα περιφερειακά επιτρέπουν στους χρήστες να πλοηγούνται στο γραφικό περιβάλλον χρήστη, τις εντολές εισόδου και να διαμορφώνουν τις ρυθμίσεις του συστήματος σύμφωνα με τις λειτουργικές απαιτήσεις.
- **GPIO Connection Pins:** Τα pin εισόδου/εξόδου γενικής χρήσης (GPIO) διευκολύνουν τη διασύνδεση εξωτερικών αισθητήρων, ενεργοποιητών και περιφερειακών συσκευών με το Raspberry Pi 4B. Τα GPIO pins επιτρέπουν αμφίδρομη επικοινωνία και έλεγχο, επιτρέποντας την ενσωμάτωση

αισθητήρων κίνησης, αισθητήρων περιβάλλοντος και άλλων περιφερειακών συσκευών στο σύστημα επιτήρησης [22].



Εικόνα. 8 GPIO connection pins

- Πιθανός αισθητήρας PIR (παθητικός αισθητήρας υπέρυθρων): Σε σενάρια που απαιτούν δυνατότητες ανίχνευσης κίνησης, ένας παθητικός αισθητήρας υπέρυθρων (PIR) μπορεί να διασυνδέεται με το Raspberry Pi 4B μέσω pins GPIO. Ο αισθητήρας PIR ανιχνεύει αλλαγές στην υπέρυθρη ακτινοβολία εντός του οπτικού του πεδίου, επιτρέποντας την ανίχνευση κίνησης και την ενεργοποίηση μηχανισμών συναγερμού εντός του συστήματος επιτήρησης [23].



Εικόνα 9 PIR sensor

Αυτά τα υλικά αποτελούν συλλογικά τη θεμελιώδη υποδομή του συστήματος παρακολούθησης κάμερας, παρέχοντας τα απαραίτητα στοιχεία υλικού για την απόκτηση δεδομένων, την επεξεργασία και την επικοινωνία εντός του παρακολουθούμενου περιβάλλοντος.

3.3 Λογισμικό

Η υποενότητα λογισμικού περιλαμβάνει το σύνολο των εργαλείων λογισμικού και πλαισίων που χρησιμοποιούνται για τη διαμόρφωση, τη λειτουργία και τη διαχείριση του συστήματος επιτήρησης κάμερας που έχει αναπτυχθεί στην πλατφόρμα Raspberry Pi 4B. Κεντρικό στοιχείο στη στοίβα λογισμικού είναι το Raspberry Pi OS, ένα λειτουργικό σύστημα που βασίζεται στο Debian και έχει βελτιστοποιηθεί για την αρχιτεκτονική Raspberry Pi. Το Raspberry Pi OS παρέχει ένα φιλικό προς το χρήστη και ελαφρύ υπολογιστικό περιβάλλον προσαρμοσμένο για ενσωματωμένες εφαρμογές, διευκολύνοντας την απρόσκοπτη ενσωμάτωση με την πλατφόρμα υλικού Raspberry Pi.

Το Raspberry Pi OS προσφέρει ένα σύστημα από προεγκατεστημένα πακέτα λογισμικού και βοηθητικά προγράμματα, συμπεριλαμβανομένων εργαλείων διαμόρφωσης συστήματος, περιβαλλόντων προγραμματισμού και εφαρμογών πολυμέσων. Ο αρθρωτός

σχεδιασμός του λειτουργικού συστήματος και η εκτεταμένη αποθήκη λογισμικού επιτρέπουν στους χρήστες να προσαρμόζουν και να επεκτείνουν τη λειτουργικότητα του συστήματος επιτήρησης σύμφωνα με συγκεκριμένες λειτουργικές απαιτήσεις [24].

Τα βασικά χαρακτηριστικά του Raspberry Pi OS περιλαμβάνουν:

- Γραφική διεπαφή χρήστη (GUI): Το Raspberry Pi OS παρέχει μια διαισθητική γραφική διεπαφή χρήστη, που επιτρέπει στους χρήστες να πλοηγούνται στις ρυθμίσεις συστήματος, να έχουν πρόσβαση σε εφαρμογές και να διαμορφώνουν περιφερειακές συσκευές μέσω ενός φιλικού προς το χρήστη περιβάλλοντος επιφάνειας εργασίας.
- Εργαλεία διαμόρφωσης συστήματος: Το Raspberry Pi OS περιλαμβάνει ένα σύνολο εργαλείων διαμόρφωσης συστήματος, όπως το `raspi-config`, που επιτρέπει στους χρήστες να διαμορφώνουν ρυθμίσεις δικτύου, να επεκτείνουν τα συστήματα αρχείων, να ενημερώνουν πακέτα λογισμικού και να διαχειρίζονται περιφερειακές συσκευές με ευκολία.
- Περιβάλλοντα προγραμματισμού: Το Raspberry Pi OS υποστηρίζει μια ποικιλία γλωσσών προγραμματισμού και περιβαλλόντων ανάπτυξης, συμπεριλαμβανομένων των Python, Scratch και Node.js, δίνοντας τη δυνατότητα στους χρήστες να αναπτύσσουν προσαρμοσμένες εφαρμογές, σενάρια και ρουτίνες αυτοματισμού προσαρμοσμένες στις λειτουργικές απαιτήσεις του συστήματος επιτήρησης.
- Υποστήριξη πολυμέσων: Το Raspberry Pi OS προσφέρει ισχυρή υποστήριξη πολυμέσων, επιτρέποντας την αναπαραγωγή και τον χειρισμό αρχείων ήχου, βίντεο και εικόνας μέσω ενσωματωμένων προγραμμάτων αναπαραγωγής πολυμέσων και πλαισίων πολυμέσων.

Αξιοποιώντας τις δυνατότητες του Raspberry Pi OS, οι χρήστες μπορούν να αξιοποιήσουν πλήρως τις δυνατότητες της πλατφόρμας Raspberry Pi 4B για να εφαρμόσουν εξελιγμένες λειτουργίες επιτήρησης, όπως ροή βίντεο σε πραγματικό χρόνο, ανίχνευση κίνησης, καταγραφή συμβάντων και απομακρυσμένη παρακολούθηση. Η ευελιξία και η επεκτασιμότητα του Raspberry Pi OS το καθιστούν ιδανική πλατφόρμα λογισμικού για την ανάπτυξη και τη διαχείριση συστημάτων παρακολούθησης κάμερας σε διαφορετικά περιβάλλοντα, που κυμαίνονται από εφαρμογές οικιακής ασφάλειας έως εμπορικές και βιομηχανικές ρυθμίσεις.

3.4 Σχεδίαση και Υλοποίηση του συστήματος

Ο σχεδιασμός και η υλοποίηση του συστήματος παρακολούθησης κάμερας περιλαμβάνει την ύπαρξη δύο ξεχωριστών αρχείων κώδικα, το καθένα προσαρμοσμένο για να εκτελεί συγκεκριμένες λειτουργίες εντός της γενικής αρχιτεκτονικής.

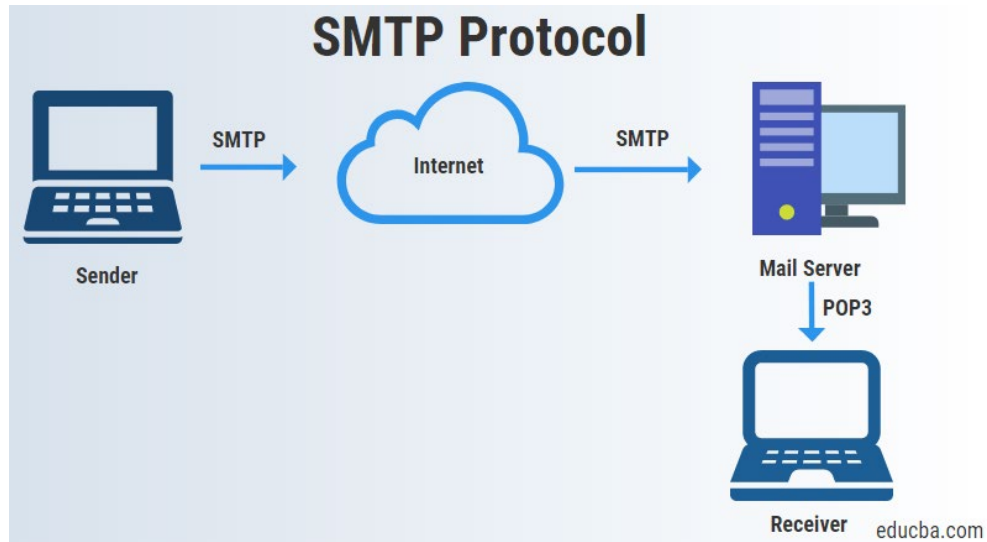
Πιο συγκεκριμένα, το πρώτο αρχείο κώδικα διέπει τον έλεγχο και τη λειτουργία του υποσυστήματος της κάμερας, διαχειριζόμενο την προετοιμασία, τη διαμόρφωση και τις λειτουργίες ροής της κάμερας web που διασυνδέεται με την πλατφόρμα Raspberry Pi 4B. Αυτό το αρχείο κώδικα είναι υπεύθυνο για τη διαχείριση των ρυθμίσεων της κάμερας, τη λήψη και την επεξεργασία καρτέ βίντεο και την εφαρμογή αλγορίθμων ανίχνευσης κίνησης για τη διάκριση ανώμαλων δραστηριοτήτων στο περιβάλλον παρακολούθησης [25], [26].

Το δεύτερο αρχείο κώδικα αφορά τη διαχείριση της διαδικασίας ειδοποίησης μέσω email, επιτρέποντας στο σύστημα να αποστέλλει αμέσως ειδοποιήσεις email σε καθορισμένους παραλήπτες κατά τον εντοπισμό προκαθορισμένων συμβάντων ή ανωμαλιών. Αυτό το αρχείο κώδικα ενσωματώνει τη λογική για τη σύνθεση email, την ανάλυση διεύθυνσης παραλήπτη και την επικοινωνία, διασφαλίζοντας την απρόσκοπτη μετάδοση των ειδοποιήσεων επιτήρησης που περιέχουν σχετικές εικόνες και πληροφορίες σχετικά με τα συμφραζόμενα.

3.5 Διαδικασία συλλογής δεδομένων

Η συλλογή δεδομένων εντός του συστήματος παρακολούθησης της κάμερας διευκολύνεται μέσω δύο βασικών τρόπων: ειδοποιήσεων που βασίζονται σε email και ενσωμάτωσης με μια πλατφόρμα Διαδικτύου των πραγμάτων (IoT). Οι τρόποι αυτοί επιτρέπουν στο σύστημα να διαδίδει δεδομένα επιτήρησης και ειδοποιήσεις στους ενδιαφερόμενους μέσω email ή καναλιών ανταλλαγής μηνυμάτων μέσω κινητού τηλεφώνου, ενισχύοντας έτσι την επίγνωση της κατάστασης και την ανταπόκριση.

Ειδικότερα, το σύστημα αξιοποιεί τις ειδοποιήσεις που βασίζονται σε ηλεκτρονικό ταχυδρομείο ως μηχανισμό συλλογής και διάδοσης δεδομένων, επιτρέποντας τη μετάδοση καταγεγραμμένων εικόνων και ειδοποιήσεων συμβάντων σε καθορισμένες διευθύνσεις email. Με την ανίχνευση κίνησης ή προκαθορισμένων συμβάντων, το σύστημα ξεκινά τη διαδικασία σύνθεσης email, ενσωματώνοντας εικόνες που έχουν ληφθεί και πληροφορίες συμφραζομένων στο σώμα του μηνύματος πριν το αποστείλει στους παραλήπτες μέσω διακομιστών SMTP [27].



Εικόνα 10 SMTP protocol

Εναλλακτικά, το σύστημα μπορεί να ενσωματωθεί με μια πλατφόρμα IoT για να διευκολύνει τη συλλογή δεδομένων και τις λειτουργίες ειδοποίησης σε πραγματικό χρόνο. Με τη διασύνδεση με υπηρεσίες IoT όπως το Blynk ή το ThingSpeak, το σύστημα μπορεί να μεταδίδει δεδομένα παρακολούθησης και ειδοποιήσεις σε κινητές συσκευές μέσω ειδοποιήσεων push ή μηνυμάτων SMS (Υπηρεσία σύντομων μηνυμάτων). Αυτή η ενοποίηση ενισχύει την ανταπόκριση και την προσβασιμότητα του συστήματος, επιτρέποντας στους ενδιαφερόμενους φορείς να λαμβάνουν έγκαιρες ειδοποιήσεις και να ανταποκρίνονται σε έκτακτα γεγονότα ανεξάρτητα από τη γεωγραφική τους θέση.

3.6 Εύρος και περιορισμοί

Η εφαρμογή του συστήματος παρακολούθησης κάμερας χρησιμοποιώντας Python στην πλατφόρμα Raspberry Pi 4B προσφέρει πολλά πλεονεκτήματα και παρουσιάζει ορισμένους περιορισμούς, οι οποίοι οριοθετούν το εύρος και τα λειτουργικά όρια της συγκεκριμένης εφαρμογής.

Πλεονεκτήματα του Raspberry Pi:

- **Κόστους-αποτελεσματικότητα:** Ο μικροϋπολογιστής Raspberry Pi παρέχει μια οικονομικά αποδοτική λύση με περιορισμένους προϋπολογισμούς, δίνοντάς τους τη δυνατότητα να αυτοματοποιούν εργασίες ή να δημιουργούν μικρές βάσεις δεδομένων και διακομιστές πολυμέσων χωρίς να επιβαρύνονται με σημαντικά έξοδα.

- Προσβασιμότητα και Ευκολία Χρήσης: Η Python, η γλώσσα προγραμματισμού που χρησιμοποιείται από το Raspberry Pi, προσφέρει ένα φιλικό προς τον χρήστη και προσβάσιμο περιβάλλον για άτομα, συμπεριλαμβανομένων των νεότερων γενεών, για να μάθουν έννοιες προγραμματισμού. Η απλότητα και η αναγνωσιμότητα της Python την καθιστούν ιδανική για αρχάριους και έμπειρους προγραμματιστές.
- Ευελιξία και πειραματισμός: Η πλατφόρμα Raspberry Pi προσφέρει άφθονο χώρο για πειραματισμό και προσαρμογή, επιτρέποντας στους χρήστες να επαναχρησιμοποιήσουν τη συσκευή για διαφορετικές εφαρμογές, απλώς ανταλλάσσοντας κάρτες SD ή τροποποιώντας τις διαμορφώσεις λογισμικού. Αυτή η προσαρμοστικότητα ενισχύει την καινοτομία και τη δημιουργικότητα στην ανάπτυξη καινοτόμων λύσεων.
- Προσαρμοστικότητα για ενσωματωμένα συστήματα: Η ικανότητα του Raspberry Pi να προβάλλει εικόνες και βίντεο υψηλής ευκρίνειας το καθιστά κατάλληλο για πρωτότυπα ενσωματωμένων συστημάτων και εφαρμογές προσαρμοστικής τεχνολογίας, επιτρέποντας την ανάπτυξη πολύπλοκων συστημάτων με ένα κλάσμα του κόστους που σχετίζεται με τις παραδοσιακές λύσεις υλικού.
- Φιλικότητα προς το περιβάλλον: Το συμπαγές μέγεθος και ο ενεργειακά αποδοτικός σχεδιασμός του Raspberry Pi συμβάλλουν στην περιβαλλοντική του βιωσιμότητα προσφέροντας μια ηθική εναλλακτική σε μεγαλύτερους διακομιστές που καταναλώνουν σημαντική ενέργεια και απαιτούν εκτεταμένα συστήματα ψύξης.

Περιορισμοί του Raspberry Pi:

- Περιορισμοί ισχύος επεξεργασίας: Ο επεξεργαστής του Raspberry Pi μπορεί να μην ταιριάζει με τις δυνατότητες απόδοσης των παραδοσιακών υπολογιστών, οδηγώντας σε περιορισμούς στις εργασίες πολλαπλών εργασιών και σε εργασίες που απαιτούν υπολογιστική ένταση. Οι σύνθετες λειτουργίες ενδέχεται να προκαλέσουν καθυστερήσεις ή αναποτελεσματικότητα στην επεξεργασία.
- Συμβατότητα λειτουργικού συστήματος: Το Raspberry Pi είναι κυρίως συμβατό με λειτουργικά συστήματα που βασίζονται σε Linux και ενδέχεται να μην υποστηρίζει άλλες πλατφόρμες, όπως τα Windows, περιορίζοντας τη συμβατότητα λογισμικού και τη διαλειτουργικότητα σε ορισμένα περιβάλλοντα.
- Πολυπλοκότητα προσαρμογής: Ενώ το Raspberry Pi διευκολύνει την προσαρμογή και τον πειραματισμό, η διαμόρφωση και η εγκατάσταση ενημερώσεων λογισμικού ή νέων εφαρμογών ενδέχεται να περιλαμβάνει

μια χρονοβόρα διαδικασία, ιδιαίτερα για χρήστες που δεν είναι εξοικειωμένοι με περιβάλλοντα που βασίζονται σε Linux.

- Περιορισμένο πεδίο εφαρμογής για αναπτύξεις μεγάλης κλίμακας: Το Raspberry Pi ενδέχεται να μην είναι κατάλληλο για μεγάλης κλίμακας αναπτύξεις ή εφαρμογές σε εταιρικό επίπεδο που απαιτούν ισχυρές δυνατότητες επεξεργασίας και εκτεταμένη υποδομή διακομιστή. Η περίπτωση χρήσης του είναι κυρίως προσαρμοσμένη σε έργα μικρής κλίμακας και σε προσπάθειες δημιουργίας πρωτοτύπων.

4. Σχεδίαση συστήματος και υλοποίηση

4.1 Ρύθμιση του Raspberry Pi και ενσωμάτωση της κάμερας

Η επιτυχής ρύθμιση του Raspberry Pi απαιτεί μια σχολαστική διαδικασία εγκατάστασης εντός του συστήματος λογισμικού, ακολουθούμενη από μια συστηματική διαμόρφωση που περιγράφεται λεπτομερώς στο Παράρτημα της παρούσας εργασίας. Αυτός ο περιεκτικός οδηγός στοχεύει να απλοποιήσει τη διαδικασία εγκατάστασης και να ενσωματώσει απρόσκοπτα το στοιχείο της κάμερας στο περιβάλλον Raspberry Pi.

Ωστόσο, η ενσωμάτωση της μονάδας κάμερας στο Raspberry Pi αντιπροσωπεύει μια βασική πτυχή της λειτουργικότητάς της. Είναι αξιοσημείωτο ότι η μονάδα κάμερας διαθέτει λειτουργία plug-and-play παρόμοια με μια τυπική κάμερα web, αποφεύγοντας την ανάγκη για περίπλοκες διαδικασίες διαμόρφωσης. Αυτή η έμφυτη ικανότητα, που αναφέρεται ως «Cam Pi», διευκολύνει την απρόσκοπτη ενσωμάτωση χωρίς να απαιτούνται εξειδικευμένες ρυθμίσεις.

4.2 Ενσωμάτωση αισθητήρων και πρωτόκολλο επικοινωνίας

Για την ενσωμάτωση αισθητήρων, το σύστημα επιλέγει αισθητήρες Παθητικής Υπέρυθρης ακτινοβολίας (PIR) που διασυνδέονται μέσω pins εισόδου γενικής χρήσης (GPIO), διευκολύνοντας την ανίχνευση κίνησης σε πραγματικό χρόνο εντός της παρακολουθούμενης περιοχής. Αξιοποιώντας τα pins GPIO του Raspberry Pi, οι αισθητήρες PIR συμβάλλουν στην ικανότητα του συστήματος να διακρίνει τις αλλαγές στην υπέρυθρη ακτινοβολία που εκπέμπεται από κινούμενα αντικείμενα, ενισχύοντας έτσι την απόκρισή του σε δυναμικά περιβαλλοντικά ερεθίσματα.

Όσον αφορά τα πρωτόκολλα επικοινωνίας, το σύστημα υιοθετεί το πρωτόκολλο MQTT (Message Queuing Telemetry Transport) για απρόσκοπτη μετάδοση δεδομένων μεταξύ της συσκευής της κάμερας και των κεντρικών διακομιστών [28]. Επιπλέον, το SMTP (Simple Mail Transfer Protocol) χρησιμοποιείται για τη διευκόλυνση της επικοινωνίας μέσω email, επιτρέποντας την άμεση αποστολή ειδοποιήσεων που περιέχουν στιγμιότυπα εικόνων που υποδηλώνουν ανιχνευμένη κίνηση. Αυτή η προσέγγιση διπλού πρωτοκόλλου εξασφαλίζει αποτελεσματική ανταλλαγή δεδομένων και έγκαιρη διάδοση κρίσιμων πληροφοριών σε καθορισμένους παραλήπτες, ενισχύοντας έτσι την αποτελεσματικότητα του συστήματος στην απόκριση σε ανιχνευόμενες ανωμαλίες.

4.3 Σύνδεση στην πλατφόρμα και ενσωμάτωση και διασύνδεση του συστήματος

Στο πλαίσιο της παρούσας εργασίας, όπου στοχεύεται η παρακολούθηση καθορισμένων περιοχών, η δημιουργία αξιόπιστης σύνδεσης με την πλατφόρμα και η απρόσκοπτη ενσωμάτωση του συστήματος είναι κρίσιμα στοιχεία για τη λειτουργικότητα και την αποτελεσματικότητά του.

Για το έργο επιτήρησης κάμερας, η πλατφόρμα Blynk χρησιμεύει ως η βέλτιστη λύση, παρέχοντας μια φιλική προς τον χρήστη διεπαφή και ισχυρή λειτουργικότητα προσαρμοσμένη για έργα IoT. Μέσω του Blynk, οι χρήστες μπορούν να αναπτύξουν κώδικες και έργα προσαρμοσμένα στις προδιαγραφές του Raspberry Pi, διευκολύνοντας τη βελτιωμένη επικοινωνία και τον έλεγχο των συνδεδεμένων συσκευών και αισθητήρων. Με το κλείσιμο της Cayenne [29], [30], η Blynk αναδεικνύεται ως μια αξιόπιστη εναλλακτική λύση, διασφαλίζοντας τη συνέχεια στις αναπτύξεις IoT.



Εικόνα 11 Cayenne platform

Η ενοποίηση συστήματος περιλαμβάνει την απρόσκοπτη ενσωμάτωση λειτουργιών όπως η επικοινωνία μέσω email και η αλληλεπίδραση με την πλατφόρμα Blynk [31]. Η ενσωμάτωση SMTP εντός της βάσης κωδικών επιτρέπει στο σύστημα να αποστέλλει

ειδοποιήσεις email που περιέχουν σχετικά στιγμιότυπα εικόνων κατά την ανίχνευση κίνησης, διασφαλίζοντας την άμεση διάδοση των κρίσιμων δεδομένων σε καθορισμένους παραλήπτες. Επιπλέον, η διασύνδεση με την πλατφόρμα Blynk επιτυγχάνεται μέσω συγκεκριμένων κωδικών που έχουν σχεδιαστεί για να διευκολύνουν την επικοινωνία και τον έλεγχο μεταξύ του Raspberry Pi και της διεπαφής Blynk. Αυτή η ενσωμάτωση επιτρέπει την απομακρυσμένη παρακολούθηση και διαχείριση του συστήματος κάμερας, βελτιώνοντας την προσβασιμότητα και τη λειτουργικότητά του.



Εικόνα 12 Blynk platform

4.4 Λειτουργικότητα συστήματος

Το σύστημα επιτήρησής μας έχει σχεδιαστεί για να ανιχνεύει και να ανταποκρίνεται σε συμβάντα κίνησης εντός του παρακολουθούμενου περιβάλλοντος διασφαλίζοντας ταυτόχρονα την άμεση επικοινωνία των σχετικών πληροφοριών στους καθορισμένους ενδιαφερόμενους φορείς. Αξιοποιώντας έναν συνδυασμό στοιχείων υλικού, αλγορίθμων λογισμικού και πρωτοκόλλων επικοινωνίας, το σύστημα λειτουργεί απρόσκοπτα για να εκπληρώσει τους στόχους του [32].

Στον πυρήνα του, το σύστημα χρησιμοποιεί προηγμένους αλγόριθμους ανίχνευσης κίνησης για την παρακολούθηση αλλαγών στο περιβάλλον ενδεικτικές κίνησης ή δραστηριότητας. Με την ανίχνευση ενός συμβάντος κίνησης, ξεκινούν προκαθορισμένες ενέργειες, συμπεριλαμβανομένης της λήψης στιγμιότυπων εικόνων ή βίντεο, της ενεργοποίησης συναγερμών και της μετάδοσης ειδοποιήσεων σε πραγματικό χρόνο σε εξουσιοδοτημένο προσωπικό ή σταθμούς παρακολούθησης.

Η ενσωμάτωση με τις δυνατότητες αποστολής email επιτρέπει στο σύστημα να αποστέλλει αμέσως ειδοποιήσεις που περιέχουν εικόνες ή βίντεο κλιπ σε καθορισμένες διευθύνσεις email, διευκολύνοντας την ταχεία διάδοση κρίσιμων πληροφοριών. Η εφαρμογή κωδικών εντοπισμού και αποστολής email εντός της αρχιτεκτονικής λογισμικού του συστήματος διασφαλίζει την αυτοματοποίηση των βασικών διαδικασιών, βελτιώνοντας τη λειτουργική αποτελεσματικότητα και την επεκτασιμότητα.

Στις επόμενες υποενότητες, θα διερευνήσουμε τις λεπτομέρειες εφαρμογής του κώδικα ανίχνευσης, εστιάζοντας στους αλγόριθμους και τις τεχνικές ανίχνευσης κίνησης. Στη συνέχεια, θα εμβαθύνουμε στην υλοποίηση του κώδικα αποστολής email, τονίζοντας την ενσωμάτωση των πρωτοκόλλων SMTP και των λειτουργιών ειδοποίησης email στην αρχιτεκτονική του συστήματος.

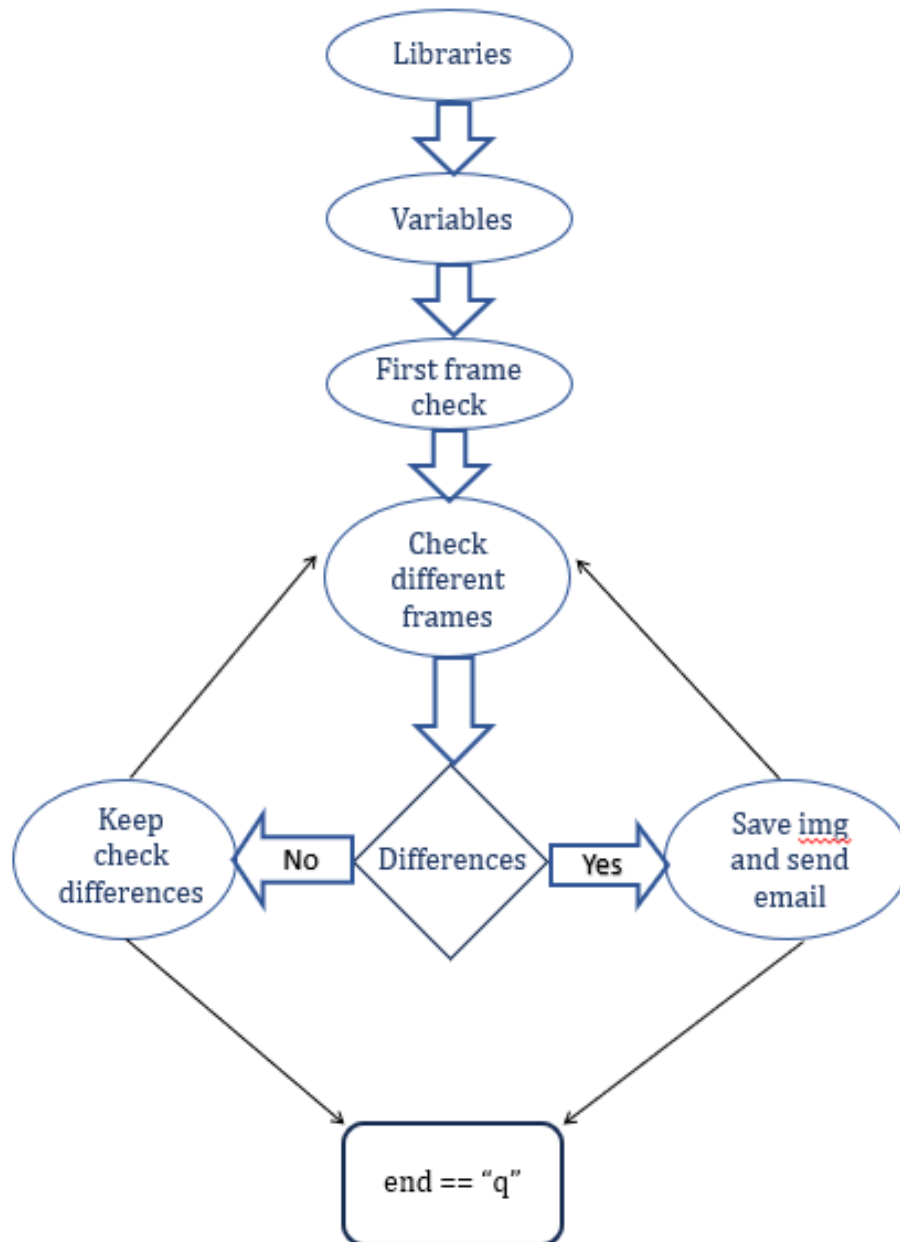
Κατανοώντας τη λειτουργικότητα και την εφαρμογή του συστήματος επιτήρησής μας, τα ενδιαφερόμενα μέρη μπορούν να αξιοποιήσουν αποτελεσματικά τις δυνατότητές του για την ενίσχυση της ασφάλειας, τον μετριασμό των κινδύνων και τη διασφάλιση της ασφάλειας των περιουσιακών στοιχείων και των πόρων σε δυναμικά περιβάλλοντα.

Παρακάτω θα περιγράψουμε τη λειτουργική διαδικασία του προγράμματός μας. Για να διασφαλίσουμε τη ορθή λειτουργία του, θα χρειαστούμε τη δημιουργία δύο αρχείων γλώσσας Python [33], με το καθένα να έχει ξεχωριστές εργασίες. Το αρχικό αρχείο θα διαδραματίσει θεμελιώδη ρόλο, εξετάζοντας ενδελεχώς εάν η κίνηση έχει διακριθεί από την καθορισμένη κάμερα. Στη συνέχεια, το δεύτερο αρχείο, κατά την ανίχνευση κίνησης, θα αποστέλλει ένα email που περιέχει την εικόνα που καταγράφει την ανιχνευόμενη κίνηση, συνοδευόμενη από σχετικές ετικέτες.

Για την υλοποίηση της εφαρμογής μας, πρωταρχική προϋπόθεση είναι η εγκατάσταση της βιβλιοθήκης OpenCV στο σύστημα Raspberry Pi 4B. Σχεδιασμένο να παρέχει ένα τυποποιημένο πλαίσιο για εφαρμογές υπολογιστικής όρασης, το OpenCV επιταχύνει τη χρήση της μηχανικής αντίληψης [34]. Εφαρμογές παρόμοιες με τα συστήματα αυτόνομης οδήγησης και αποφυγής σύγκρουσης στα αυτοκίνητα έχουν εφαρμοστεί με επιτυχία χρησιμοποιώντας αλγόριθμους παρόμοιους με αυτούς που είναι εγγενείς στο OpenCV. Επιπλέον, οι συνηθισμένες εφαρμογές της όρασης υπολογιστή περιλαμβάνουν την αναγνώριση αντικειμένων και προσώπου. Η προσπάθειά μας θα αξιοποιήσει ένα κλάσμα των λειτουργιών ανίχνευσης κίνησης του OpenCV.

Αυτή η βιβλιοθήκη διαθέτει εκτεταμένο περιεχόμενο περίπλοκων εργαλείων. Αρκεί μια ελάχιστη διαμόρφωση υλικού του Raspberry Pi 2, αν και η βέλτιστη απόδοση διακρίνεται με το Raspberry Pi 4, ειδικά κατά το χειρισμό απαιτητικών εργασιών. Επιπλέον, είναι επιτακτική ανάγκη να διασφαλισθούν τουλάχιστον 4 GB διαθέσιμου χώρου στην κάρτα SD.

Μετά την επικύρωση ότι η συσκευή μας πληροί τις προβλεπόμενες προϋποθέσεις, η διαδικασία εγκατάστασης της προαναφερθείσας βιβλιοθήκης ξεκινά χωρίς καθυστέρηση. Με την ολοκλήρωση της διαδικασίας εγκατάστασης, είμαστε έτοιμοι να ξεκινήσουμε την εφαρμογή του κώδικα μας [35]. Για την αποτελεσματική πλοήγηση αυτής της προσπάθειας, θα πρέπει να γίνει αναφορά στο συνοδευτικό διάγραμμα ροής.



Εικόνα 13 Διάγραμμα ροής

4.4.1 Υλοποίηση κώδικα ανίχνευσης

Η εφαρμογή του κώδικα ανίχνευσης κίνησης σηματοδοτεί μια κομβική φάση στην ανάπτυξη του προγράμματός μας. Όπως συνηθίζεται στις προσπάθειες προγραμματισμού, ξεκινάμε με την ενσωμάτωση βασικών βιβλιοθηκών στην αρχή του προγράμματος.

Πριν από την εισαγωγή της βιβλιοθήκης, είναι επιτακτική ανάγκη να καθοριστεί η κωδικοποίηση του πηγαίου κώδικα σε utf-8. Αυτό το προληπτικό μέτρο μετριάζει τα σφάλματα κωδικοποίησης στα αρχεία μας. Η προβλεπόμενη μορφή για τη δήλωση κωδικοποίησης, που αναγνωρίζεται από τους περισσότερους συντάκτες, συνεπάγεται τη συμπερίληψη της ακόλουθης εντολής στις αρχικές γραμμές του κώδικα:

```
#!/usr/bin/python
```

```
# -*- encoding: <encoding name> -*-
```

Εδώ, το <encoding name> αντικαθίσταται από το 'utf-8', το επιλεγμένο μας πρότυπο κωδικοποίησης για την υλοποίηση του προγράμματος. Εναλλακτικά, μπορούν να χρησιμοποιηθούν άλλες κωδικοποιήσεις όπως "iso-8859-15" ή "Latin-1".

Στη συνέχεια, προχωράμε στην εισαγωγή των απαραίτητων βιβλιοθηκών που είναι απαραίτητες για τη λειτουργικότητα του προγράμματος. Στην ανάλυσή μας, εξετάζουμε εξονυχιστικά το αρχείο **detect_motion.py**, το οποίο έχει ως αποστολή την ανίχνευση κίνησης μέσω μιας κάμερας web ή μιας κάμερας κινητού τηλεφώνου που χρησιμοποιεί μια διεύθυνση IP. Αν και η απεικόνισή μας αφορά κυρίως μια κάμερα web, υπάρχουν προβλέψεις για τη χρήση κάμερας κινητού τηλεφώνου.

Η εισαγωγή της βιβλιοθήκης OpenCV εκτελείται μέσω της εντολής εισαγωγής cv2, διευκολύνοντας την απρόσκοπτη ενσωμάτωση των λειτουργιών OpenCV στο πρόγραμμά μας. Επιπλέον, η χρήση της βιβλιοθήκης ημερομηνίας επιτρέπει την απόκτηση ακριβών χρονικών σημάνσεων που αντιστοιχούν στην εμφάνιση γεγονότων κίνησης. Επιπλέον, ακολουθεί η προετοιμασία τριών μεταβλητών:

Μια μεταβλητή που προορίζεται για την αποθήκευση καρέ κάμερας, που αρχικά ορίστηκε σε Κανένα(*none*).

Δύο μεταβλητές Boolean, δηλαδή *SendEmails* και *SaveImage*, αρχικοποιήθηκαν σε *True*. Το πρώτο διέπει τις ειδοποιήσεις μέσω email, ενώ το δεύτερο υπαγορεύει τη λειτουργία αποθήκευσης εικόνας που σχετίζεται με την ανίχνευση κίνησης.

Ένας έλεγχος υπό όρους που βασίζεται στη μεταβλητή *SendEmails* κατευθύνει τη ροή του προγράμματος προς την επίκληση του δεύτερου αρχείου, *s_email.py*, και την ενσωμάτωση της βιβλιοθήκης χρόνου. Η βιβλιοθήκη χρόνου χρησιμεύει για να εισάγει μια καθυστέρηση στη διαδικασία μετάδοσης email, διασφαλίζοντας ότι τα email

αποστέλλονται κατά διαστήματα, αποτρέποντας έτσι τον κατακλυσμό πολλών email. Αυτή η λειτουργία εκτελείται μέσω της εισαγωγής της μονάδας χρόνου και της προετοιμασίας μιας μεταβλητής μέτρησης χρόνου που υποδηλώνεται με το `timeemailsnt`.

Σε σενάρια όπου μια κάμερα web δεν είναι διαθέσιμη, η προσφυγή σε κάμερα κινητού τηλεφώνου είναι εφικτή. Σε αυτό το πλαίσιο, η εκχώρηση μιας μεταβλητής `IpCam=False` είναι κατάλληλη, που υποδηλώνει την απουσία διαμόρφωσης κάμερας IP. Η λογική πίσω από την εκχώρηση `False` είναι διπλή: πρώτον, να απαιτείται η παροχή της διεύθυνσης IP της κάμερας του κινητού τηλεφώνου για λόγους συνδεσιμότητας και δεύτερον, να δημιουργηθεί η μεταβλητή για αξιολόγηση υπό όρους.

Μέσα σε μια ρήτρα `'if'`, εισάγουμε την εντολή `import urllib.request`, διευκολύνοντας την ανάκτηση εικόνων που δημοσιεύονται από την εφαρμογή IP Webcam στο τοπικό μας δίκτυο. Το `urllib.request` παρέχει λειτουργίες και κλάσεις απαραίτητες για την ανάγνωση σελίδων ή αντικειμένων από το Διαδίκτυο, περιλαμβάνοντας λειτουργικότητα για πολύπλοκα περιβάλλοντα όπως βασικός και ψηφιακός έλεγχος ταυτότητας, ανακατευθύνσεις, cookies και άλλα.

Στη συνέχεια, η ενσωμάτωση της `numpy` βιβλιοθήκης είναι εγγυημένη. Φημισμένο για την παροχή πολυδιάστατων αντικειμένων συστοιχιών και για μια πληθώρα ρουτινών για γρήγορες λειτουργίες σε συστοιχίες, το `Numpy` εξέχει στην Επιστήμη των Δεδομένων και τη μηχανική μάθηση. Η χρήση του διευκολύνει μεταξύ άλλων αποτελεσματικές μαθηματικές, λογικές και στατιστικές πράξεις.

Η εντολή `url=http://X.X.X.X:8080/shot.jpg`, όπου το 'X' αντιπροσωπεύει τον αριθμό IP της WebCamera μας και το '8080' υποδηλώνει τον αριθμό θύρας, μια κοινή θύρα πρωτοκόλλου TCP.

Στην επόμενη ρήτρα `'else'`, η εκτέλεση προχωρά ανάλογα με τη μεταβλητή `IpCam` που ισοδυναμεί με `False`. Εδώ, ένα αντικείμενο βίντεο δημιουργείται για να αντιστοιχεί στην είσοδο βίντεο μας. Η τιμή '0' στο `VideoCapture` υποδηλώνει την επιλογή της πρώτης κάμερας που είναι συνδεδεμένη στη συσκευή μας, ενσωματωμένη στην εντολή `video = cv2.VideoCapture(0)`.

Στη συνέχεια, το πρόγραμμά μας έχει ρυθμιστεί ώστε να επεξεργάζεται μεμονωμένα καρέ του βίντεο σε έναν άπειρο βρόχο, που συμβολίζεται με ένα `True`. Μέσα σε αυτόν τον βρόχο, ο πρωταρχικός στόχος είναι να διακρίνονται οι διαφορές μεταξύ διαδοχικών καρέ, διευκολύνοντας την ανίχνευση και την ανάλυση κίνησης. Αυτή η επαναληπτική διαδικασία είναι απαραίτητη για την παρακολούθηση και τον εντοπισμό γεγονότων κίνησης σε πραγματικό χρόνο στο περιβάλλον επιτήρησης.

Σε περίπτωση που χρησιμοποιείται η εφαρμογή IP Webcam, η ανάκτηση της διεύθυνσης εικόνας μέσω urllib διευκολύνεται, όπως δηλώθηκε προηγουμένως στη μεταβλητή url. Η επόμενη εντολή `frame = urllib.request.urlopen(url)` επιτρέπει την ανάκτηση της εικόνας από την καθορισμένη διεύθυνση URL, μετατρέποντάς την σε μια αναγνώσιμη μορφή.

Η ενσωμάτωση του numpy είναι καθοριστική για τη μετατροπή της λαμβανόμενης εικόνας από το κινητό σε πίνακα. Αυτός ο μετασχηματισμός εκτελείται μέσω της εντολής:

```
frame = numpy.array(bytearray(frame.read()))
```

Μετατρέποντας την εικόνα σε πίνακα, γίνεται επιδεκτική περαιτέρω επεξεργασίας και ανάλυσης. Στη συνέχεια, η συνάρτηση `imdecode` χρησιμοποιείται για την αποκωδικοποίηση της έγχρωμης εικόνας στα συνιστώμενα έγχρωμα κανάλια BGR, καθιστώντας την συμβατή με εντολές OpenCV:

```
frame = cv2.imdecode(frame, cv2.IMREAD_UNCHANGED)
```

Η συνάρτηση `cv2.imdecode()`, εγγενής στη βιβλιοθήκη OpenCV της Python, διευκολύνει την ερμηνεία των δεδομένων εικόνας από ένα buffer και τη μετατροπή τους σε μορφή εικόνας. Αυτή η λειτουργία είναι ιδιαίτερα χρήσιμη για την αποτελεσματική φόρτωση εικόνων από το διαδίκτυο.

Η σύνταξη της συνάρτησης είναι η εξής:

```
cv2.imdecode(buf, flags)
```

όπου:

- `buf`: Αντιπροσωπεύει τα δεδομένα εικόνας που λαμβάνονται σε byte
- `flags`: Καθορίζει τη μέθοδο με την οποία πρέπει να διαβάζεται η εικόνα

Η προεπιλεγμένη τιμή είναι `cv2.IMREAD_COLOR`.

Κατά την εκτέλεση, η συνάρτηση επιστρέφει έναν πίνακα εικόνων, επιτρέποντας έτσι τις επόμενες λειτουργίες επεξεργασίας εικόνας εντός του προγράμματος.

Οι λειτουργικές διαδικασίες εντός του προγράμματός μας συνεχίζονται με την εντολή `video.read()`, επιτρέποντας την εξαγωγή ενός καρέ από την τροφοδοσία βίντεο της κάμερας. Η μεταβλητή `ret` επιστρέφει μια Boolean τιμή `True` εάν το πλαίσιο διαβάζεται με επιτυχία, υποδηλώνοντας κανονική λειτουργία και `False` εάν προκύψουν ανωμαλίες κατά την ανάκτηση πλαισίου. Η εντολή `ret, frame = video.read()` ενσωματώνει αυτή τη λειτουργία.

Στη συνέχεια, το μεταβλητό κείμενο αρχικοποιείται για να αποθηκεύσει το μήνυμα που θα εμφανιστεί στο βίντεο κατά την ανίχνευση κίνησης. Αυτή η αρχικοποίηση συμβολίζεται ως εξής: `text = ""`.

Η μέθοδος `cvtColor()` διευκολύνει τη μετατροπή χρωματικού χώρου εντός του προγράμματος. Το OpenCV προσφέρει μια πληθώρα μεθόδων μετατροπής χρωματικού χώρου, που ξεπερνούν τις 150 στον αριθμό. Εδώ, χρησιμοποιούμε το `cv2.COLOR_BGR2GRAY` ως όρισμα, μετατρέποντας την έγχρωμη εικόνα που λαμβάνεται από την κάμερα σε κλίμακα του γκρι:

```
gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
```

Για να βελτιωθεί η ανιχνευσιμότητα των αλλαγών, η εικόνα θαμπώνει χρησιμοποιώντας το `cv2.GaussianBlur()`. Το μέγεθος του πυρήνα σε pixel καθορίζεται ως το δεύτερο όρισμα, με μεγαλύτερους πυρήνες να έχουν ως αποτέλεσμα μεγαλύτερη θαμπάδα εικόνας. Η σύνταξη έχει ως εξής:

```
gray = cv2.GaussianBlur(gray, (61, 61), 0)
```

Κατά την αρχική επανάληψη του βρόχου `while`, το πρώτο πλαίσιο αποθηκεύεται στη μεταβλητή `first_frame` για επόμενες συγκρίσεις:

```
if first_frame is None:
```

```
    first_frame = gray
```

Η δήλωση συνέχειας διευκολύνει τη μετάβαση στην επόμενη επανάληψη του βρόχου `while`. Με κάθε νέο `frame`, γίνεται σύγκριση με το αρχικό `frame` χρησιμοποιώντας τη μέθοδο `cv2.absdiff()` και η διαφορά που προκύπτει αποθηκεύεται στη μεταβλητή `diff_frame`:

```
diff_frame = cv2.absdiff(first_frame, gray)
```

Στη συνέχεια, η μέθοδος `cv2.threshold()` χρησιμοποιείται για την αξιολόγηση της διαφοράς κάθε pixel στο τρέχον `frame` σε σχέση με το αρχικό `frame`. Εάν η διαφορά υπερβαίνει την τιμή κατωφλίου των 75, όπως καθορίζεται στο `cv2.threshold()`, εμφανίζεται το λευκό χρώμα (255) στα αντίστοιχα pixel, τα οποία στη συνέχεια αποθηκεύονται στη μεταβλητή `thresh_frame`. Μπορεί να απαιτούνται προσαρμογές στην τιμή κατωφλίου για τη βελτιστοποίηση της απόδοσης ανίχνευσης κίνησης:

```
ret, thresh_frame = cv2.threshold(diff_frame, 75, 255, cv2.THRESH_BINARY)
```

Είναι επιτακτική ανάγκη να αναγνωρίσουμε ότι τα χρώματα κωδικοποιούνται χρησιμοποιώντας κωδικοποίηση RGB, με τα 75 και 255 να χρησιμεύουν ως αριθμητικές αναπαραστάσεις εντός του χρωματικού φάσματος.

Μετά τη διαδικασία διαστολής, χρησιμοποιούμε μια εντολή για να επεκτείνουμε το αντικείμενό μας, δηλαδή την εικόνα, για να ενισχύσουμε τη ιδιαιτερότητά του. Οι θεμελιώδεις αρχές της διαστολής στο OpenCV της Python συνεπάγονται την ικανότητά του να:

- Αυξάνει την περιοχή του αντικειμένου
- Επισημαίνει τα χαρακτηριστικά μέσα στην εικόνα

Η συνάρτηση dilate στο OpenCV της Python λειτουργεί ως εξής:

- Ένας πυρήνας, που αντιπροσωπεύεται από έναν πίνακα περιττού μεγέθους (π.χ., 3x3, 5x5, 7x7), επικαλύπτεται στην εικόνα.
- Ένα στοιχείο pixel στην αρχική εικόνα ορίζεται σε "1" εάν τουλάχιστον ένα pixel κάτω από τον πυρήνα πληροί τα κριτήρια κατωφλίου.
- Αυτή η διαδικασία ενισχύει τη λευκή περιοχή μέσα στην εικόνα ή μεγεθύνει το μέγεθος του αντικειμένου του προσκηνίου.
- Με τη cv2.dilate(), αυξάνουμε το μέγεθος του αντικειμένου που προσδιορίζεται από τη cv2.threshold(), ενισχύοντας έτσι την εξέχουσα θέση του. Η εντολή που χρησιμοποιείται για αυτή τη λειτουργία είναι η εξής:

thresh_frame = cv2.dilate(thresh_frame, None, iterations=2)

Στη συνέχεια, προχωράμε στην οριοθέτηση του περιγράμματος του αντικειμένου που περιέχεται στη μεταβλητή thresh_frame. Αυτό το περίγραμμα χρησιμεύει για να τονίσει την ανίχνευση κίνησης μέσα στην εικόνα. Το OpenCV προσφέρει τη λειτουργία findContours(), η οποία διευκολύνει την εξαγωγή περιγραμμάτων από εικόνες. Είναι πιο αποτελεσματικό όταν εφαρμόζεται σε δυαδικές εικόνες, απαιτώντας προηγούμενη εφαρμογή τεχνικών κατωφλίου ή άκρων Sobel. Τα περιγράμματα αντιπροσωπεύουν τη γραμμή που συνδέει όλα τα σημεία κατά μήκος του ορίου μιας εικόνας που μοιράζονται το ίδιο επίπεδο έντασης. Αποδεικνύονται ανεκτίμητα στην ανάλυση σχήματος, στον προσδιορισμό του μεγέθους του αντικειμένου και στην ανίχνευση αντικειμένων.

Η συνάρτηση cv2.findContours() προσδιορίζει περιγράμματα αντικειμένων εντός του thresh_frame, αποθηκεύοντας τα αποτελέσματα στη λίστα περιγραμμάτων. Η σύνταξη που χρησιμοποιείται για αυτή τη λειτουργία είναι η εξής:

contours, _ = cv2.findContours(thresh_frame, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)

Στη συνέχεια, προχωράμε στην εξακρίβωση της έκτασης του εμβαδού του αντικειμένου εντός του ορίου. Αυτή η αξιολόγηση είναι ζωτικής σημασίας, καθώς παρέχει πληροφορίες για τη χωρική κατοχή του αντικειμένου εντός των συνόρων, ανάλογα με την

ανάλυση της κάμερας. Στην επίδειξή μας, ορίσαμε το όριο στα 10.000 pixel για να δημιουργήσουμε μια σημαντική περιοχή για ανάλυση. Για να προσδιορίσουμε τον αριθμό pixel εντός των ορίων μας, χρησιμοποιούμε την εντολή `cv2.contourArea()`, με τη λίστα περιγραμμάτων που αποθηκεύσαμε προηγουμένως να χρησιμεύει ως μεταβλητή.

Ακολουθούν ορισμένες βασικές πτυχές σχετικά με την εντολή `cv2.contourArea()`:

Αυτή η εντολή επιτρέπει την εξαγωγή ιδιοτήτων αντικειμένων που χρησιμοποιούνται συνήθως, όπως στερεότητα, ισοδύναμη διάμετρος, εικόνα μάσκας, μέση ένταση κ.λπ.

1. Aspect Ratio

Ο λόγος διαστάσεων υποδηλώνει την αναλογία του πλάτους προς το ύψος του δεξιού περιγράμματος του αντικειμένου, όπως φαίνεται στην παρακάτω εξίσωση [36]:

$$\textit{Aspect Ratio} = \frac{\textit{Width}}{\textit{Height}}$$

Αυτή η αναλογία χρησιμεύει ως κεντρική μέτρηση στον χαρακτηρισμό και την ανάλυση αντικειμένων. Παρέχει πληροφορίες σχετικά με την επιμήκυνση ή τη συμπίεση του αντικειμένου κατά μήκος των οριζόντιων και κάθετων αξόνων του, διευκολύνοντας έτσι την ολοκληρωμένη αξιολόγηση του αντικειμένου.

2. Έκταση

Στη συνέχεια, διερευνούμε την έννοια της έκτασης, μια θεμελιώδη ιδιότητα που περικλείει τον λόγο του εμβαδού του περιγράμματος προς το εμβαδόν του οριοθετημένου ορθογωνίου. Όπως φαίνεται στην παρακάτω εξίσωση, η έκταση χρησιμεύει ως κρίσιμη μέτρηση στον χαρακτηρισμό αντικειμένων και στη χωρική ανάλυση [37].

$$\textit{Extent} = \frac{\textit{Object Area}}{\textit{Bounding Rectangle Area}}$$

Ο υπολογισμός της έκτασης παρέχει πολύτιμες πληροφορίες σχετικά με τον βαθμό κάλυψης ή πλήρωσης του οριοθετημένου ορθογωνίου από το περίγραμμα. Μια τιμή μεγάλης έκτασης σημαίνει ότι το περίγραμμα καταλαμβάνει ένα σημαντικό τμήμα του οριοθετημένου ορθογωνίου, ενδεικτικό μιας καλά καθορισμένης και ουσιαστικής παρουσίας αντικειμένου. Αντίθετα, μια τιμή χαμηλής έκτασης υποδηλώνει ότι το περίγραμμα καταλαμβάνει ένα σχετικά μικρότερο τμήμα του οριοθετημένου ορθογωνίου, υπονοώντας είτε μερική παρουσία αντικειμένου είτε ένα λιγότερο καθορισμένο όριο αντικειμένου.

3. Στερεότητα

Μια άλλη κρίσιμη ιδιότητα του περιγράμματος είναι η στερεότητα, η οποία αντιπροσωπεύει την αναλογία του εμβαδού του περιγράμματος προς την περιοχή του κυρτού περιβλήματος, όπως φαίνεται στην παρακάτω εξίσωση [37].

$$\text{Solidity} = \frac{\text{Contour Area}}{\text{Convex Hull Area}}$$

Η στερεότητα χρησιμεύει ως βασική μέτρηση στον χαρακτηρισμό και την ανάλυση αντικειμένων, προσφέροντας πληροφορίες για τη συμπαγή και κυρτότητα του σχήματος του αντικειμένου. Μια υψηλότερη τιμή στερεότητας υποδηλώνει ότι το περίγραμμα ακολουθεί πιστά το σχήμα του αντικειμένου, εμφανίζοντας ελάχιστες ανωμαλίες ή κοιλότητες. Αντίθετα, μια χαμηλότερη τιμή στερεότητας υποδηλώνει την παρουσία κοιλότητων ή εσοχών μέσα στο σχήμα του αντικειμένου, υποδηλώνοντας μια λιγότερο συμπαγή και πιο ακανόνιστη δομή.

4. Ισοδύναμη διάμετρος

Η ισοδύναμη διάμετρος αντιπροσωπεύει τη διάμετρο του κύκλου του οποίου η περιοχή ταιριάζει με αυτή του περιγράμματος, όπως φαίνεται στην παρακάτω εξίσωση [37].

$$\text{Equivalent Diameter} = \sqrt{\frac{4 \times \text{Contour Area}}{\pi}}$$

Αυτή η μέτρηση χρησιμεύει ως θεμελιώδης παράμετρος στον χαρακτηρισμό αντικειμένων και τη χωρική ανάλυση, παρέχοντας πληροφορίες για το μέγεθος και την κλίμακα του αντικειμένου μέσα στην εικόνα. Με την ποσοτικοποίηση της ισοδύναμης διαμέτρου, οι αναλυτές μπορούν να καθορίσουν ένα τυποποιημένο μέτρο για το μέγεθος του αντικειμένου, διευκολύνοντας τις συγκριτικές αξιολογήσεις και την ανάλυση διαστάσεων σε διάφορα σενάρια απεικόνισης.

Ο υπολογισμός της ισοδύναμης διαμέτρου επιτρέπει τη μετατροπή ακανόνιστων σχημάτων αντικειμένων σε μια τυποποιημένη κυκλική αναπαράσταση, απλοποιώντας τον χαρακτηρισμό αντικειμένων και τις εργασίες σύγκρισης. Μια μεγαλύτερη ισοδύναμη διάμετρος υποδηλώνει μεγαλύτερο μέγεθος αντικειμένου, ενώ μια μικρότερη ισοδύναμη διάμετρος υποδηλώνει μικρότερο μέγεθος αντικειμένου, ανεξάρτητα από το ακανόνιστο σχήμα ή την πολυπλοκότητα του αντικειμένου.

Η ενσωμάτωση της ισοδύναμης διαμέτρου στις ροές εργασίας επεξεργασίας εικόνας ενισχύει την ακρίβεια και την αξιοπιστία των εργασιών εκτίμησης μεγέθους αντικειμένου και χωρικής ανάλυσης. Επιπλέον, χρησιμεύει ως θεμελιώδης παράμετρος στους

αλγόριθμους εξαγωγής χαρακτηριστικών και αναγνώρισης προτύπων, επιτρέποντας ισχυρή και ολοκληρωμένη ανάλυση αντικειμένων σε ψηφιακές εικόνες.

5. Προσανατολισμός

Ο προσανατολισμός ενός αντικειμένου αναφέρεται στη γωνία στην οποία είναι τοποθετημένο. Αυτή η γωνία μπορεί να προσδιοριστεί μαζί με τα μήκη του κύριου και του δευτερεύοντος άξονα χρησιμοποιώντας τη μέθοδο `cv2.fitEllipse(cnt)`. Χρησιμοποιώντας αυτή τη μέθοδο, εξάγονται οι ακόλουθες μεταβλητές: (x,y) που αντιπροσωπεύει το κέντρο της έλλειψης, (MA,ma) που υποδεικνύει τα μήκη του κύριου και του δευτερεύοντος άξονα αντίστοιχα, και η γωνία που υποδηλώνει τη γωνία προσανατολισμού του αντικειμένου [37].

6. Μάσκα και σημεία pixel

Σε ορισμένα σενάρια, μπορεί να είναι απαραίτητο να ληφθούν όλα τα σημεία που περιλαμβάνουν ένα δεδομένο αντικείμενο. Αυτό μπορεί να επιτευχθεί με την ακόλουθη μέθοδο [37]:

```
mask = np.zeros(imgray.shape, np.uint8)
cv2.drawContours(mask, [cnt], 0, 255, -1)
pixelpoints = np.transpose(np.nonzero(mask))
```

Εδώ, παρουσιάζονται δύο διακριτές μέθοδοι: η μία αξιοποιεί τις συναρτήσεις NumPy, ενώ η άλλη χρησιμοποιεί μια συνάρτηση OpenCV (η τελευταία γραμμή που σχολιάστηκε) για να επιτύχει το ίδιο αποτέλεσμα. Παρόλο που και οι δύο μέθοδοι αποδίδουν συγκρίσιμα αποτελέσματα, υπάρχει μια λεπτή διαφορά: Το NumPy παρέχει συντεταγμένες σε μορφή (γραμμή, στήλη), ενώ το OpenCV παρέχει συντεταγμένες σε μορφή (x,y) . Ως εκ τούτου, οι έξοδοι ενδέχεται να διαφέρουν ελαφρώς. Αξίζει να σημειωθεί ότι γραμμή = x και στήλη = y .

7. Μέγιστη τιμή, ελάχιστη τιμή και οι θέσεις τους

Αυτές οι παράμετροι μπορούν να προσδιοριστούν χρησιμοποιώντας μια εικόνα μάσκας [37]:

```
min_val, max_val, min_loc, max_loc = cv2.minMaxLoc(imgray, mask=mask)
```

8. Μέσο χρώμα ή μέση ένταση

Το μέσο χρώμα ή η μέση ένταση ενός αντικειμένου μπορεί να υπολογιστεί χρησιμοποιώντας την ακόλουθη μέθοδο [37]:

```
mean_val = cv2.mean(im, mask=mask)
```

9. Ακραία σημεία

Τα ακραία σημεία αντιπροσωπεύουν τα ανώτερα, κατώτερα, δεξιά και αριστερά σημεία του αντικειμένου. Μπορούν να αναγνωριστούν χρησιμοποιώντας τις ακόλουθες εντολές [37]:

```
leftmost = tuple(cnt[cnt[:, :, 0].argmin()][0])
```

```
rightmost = tuple(cnt[cnt[:, :, 0].argmax()][0])
```

```
topmost = tuple(cnt[cnt[:, :, 1].argmin()][0])
```

```
bottommost = tuple(cnt[cnt[:, :, 1].argmax()][0])
```

Για να σχεδιασθεί ένα οριοθετημένο ορθογώνιο γύρω από το αντικείμενο και να εμφανισθεί ενημερωτικό κείμενο στο τρέχον frame, χρησιμοποιούνται οι μέθοδοι `cv2.rectangle()` και `cv2.putText()`. Αυτές οι μέθοδοι επιτρέπουν την προσθήκη οπτικών σχολιασμών και πληροφοριών κειμένου για τη βελτίωση της ανίχνευσης και ανάλυσης αντικειμένων εντός του πλαισίου. Μετά την ολοκλήρωση της επεξεργασίας του καρέ, η μέθοδος `cv2.imshow()` χρησιμοποιείται για την εμφάνιση της εικόνας που προκύπτει.

Μετά τη ρύθμιση πολλών παραθύρων, το καθένα εξυπηρετεί διαφορετικούς σκοπούς, προχωράμε στην εμφάνιση των επεξεργασμένων εικόνων μέσα στα καθορισμένα παράθυρα. Η συνάρτηση `cv2.imshow()` διευκολύνει την παρουσίαση εικόνων μέσα σε καθορισμένα παράθυρα, προσφέροντας οπτική απεικόνιση διαφόρων σταδίων επεξεργασίας εικόνας και ανίχνευσης αντικειμένων.

Οι παράμετροι της συνάρτησης `cv2.imshow()` είναι οι εξής:

window_name: Μια συμβολοσειρά που υποδηλώνει το όνομα του παραθύρου στο οποίο θα εμφανίζεται η εικόνα.

image: Η εικόνα που θα εμφανιστεί.

Εδώ είναι τα παράθυρα και οι αντίστοιχες εικόνες τους:

1. Grayscale Window

Εμφανίζει την εικόνα σε αποχρώσεις του γκρι, ακολουθώντας τη διαδικασία μετατροπής:

```
cv2.imshow("Grayscale", gray)
```

2. Difference Window

Εμφανίζει τα pixel της γκρι εικόνας που διαφέρουν από το αρχικό πλαίσιο:

```
cv2.imshow("Difference", diff_frame)
```

3. Threshold Window

Παρουσιάζει μια εικόνα μόνο με δύο χρώματα (ασπρόμαυρο), όπου τα λευκά pixel υποδηλώνουν διαφορά από το αρχικό πλαίσιο που υπερβαίνει την τιμή κατωφλίου που έχει οριστεί στο `cv2.threshold()`:

```
cv2.imshow("Threshold", thresh_frame)
```

4. Color Window

Εμφανίζει το τρέχον έγχρωμο πλαίσιο μαζί με τα περιγράμματα των αντικειμένων που εντοπίστηκαν:

```
cv2.imshow("Color", frame)
```

Μεταξύ αυτών των παραθύρων, η έγχρωμη οθόνη πλαισίου είναι η κύρια, που δείχνει την τρέχουσα κατάσταση του πλαισίου μαζί με περιγράμματα αντικειμένων. Τα προηγούμενα παράθυρα χρησιμεύουν για τη διαμόρφωση της εφαρμογής και την προσαρμογή στις προδιαγραφές και τις απαιτήσεις της κάμερας.

Στη συνέχεια, η ανίχνευση κίνησης επαληθεύεται εξετάζοντας την αρχικοποιημένη μεταβλητή `SaveImage` και προσδιορίζοντας εάν έχει εντοπιστεί κάποια κίνηση. Αυτή η αξιολόγηση πραγματοποιείται χρησιμοποιώντας τη συνάρτηση `len()`, η οποία υπολογίζει τον αριθμό των χαρακτήρων σε μια μεταβλητή συμβολοσειράς. Το αποτέλεσμα αυτής της αξιολόγησης χρησιμεύει ως καθοριστικός παράγοντας για την ενεργοποίηση περαιτέρω ενεργειών με βάση την ανιχνευόμενη κίνηση.

Αφού αξιολογήσουμε τις συνθήκες που θέτει η συνάρτηση `len()` και επαληθεύσουμε την εγκυρότητα της μεταβλητής `SaveImage`, προχωράμε στα επόμενα βήματα της εφαρμογής μας. Ο πρωταρχικός στόχος σε αυτό το στάδιο είναι να αποθηκεύσουμε τοπικά τη λήψη της εικόνας στη συσκευή μας Raspberry Pi. Αυτή η διαδικασία περιλαμβάνει τον ορισμό μιας μεταβλητής ονόματος αρχείου με βάση την τρέχουσα ημερομηνία και ώρα και

τη χρήση της συνάρτησης `cv2.imwrite()` για την αποθήκευση της εικόνας. Η σύνταξη της συνάρτησης `cv2.imwrite()` είναι η εξής:

```
cv2.imwrite(filename, image)
```

όπου οι παράμετροι της εντολής:

filename: Μια συμβολοσειρά που αντιπροσωπεύει το όνομα αρχείου. Πρέπει να περιλαμβάνει μια μορφή εικόνας όπως `.jpg`, `.png`, κ.λπ.

image: Η εικόνα που θα αποθηκευτεί.

Η συνάρτηση επιστρέφει `True` εάν η εικόνα αποθηκευτεί με επιτυχία.

Μετά τη διαδικασία αποθήκευσης εικόνας, εφαρμόζουμε έναν μηχανισμό ελέγχου για την ενεργοποίηση της λειτουργίας ειδοποίησης μέσω email εντός του καθορισμένου χρονικού διαστήματος των 30 δευτερολέπτων. Αυτός ο μηχανισμός ελέγχου χρησιμοποιεί τη λειτουργία `time()` για να προγραμματίζει με ακρίβεια τις ειδοποιήσεις μέσω email, διασφαλίζοντας έγκαιρες ειδοποιήσεις χωρίς να πλημμυρίζει τα εισερχόμενα του παραλήπτη με περιττά email.

Κατά την αποστολή της ειδοποίησης email, η μεταβλητή `timeemailsnt` ενημερώνεται με την τρέχουσα χρονική σήμανση, επιτρέποντας την ακριβή παρακολούθηση των χρόνων αποστολής email για τον επόμενο προγραμματισμό.

Καθώς πλησιάζουμε στο τέλος του πρώτου μέρους του κώδικά μας, ενσωματώνουμε μια συνάρτηση για την ανίχνευση εισόδου πληκτρολογίου χρησιμοποιώντας `cv2.waitKey()`. Η ρύθμιση της τιμής σε 1 διασφαλίζει τη λειτουργία ζωντανού βίντεο, επιτρέποντας τη συνεχή επεξεργασία και ανάλυση βίντεο. Η επιλογή μεταξύ της ρύθμισης του `waitKey()` σε 0 ή 1 καθορίζει τη συμπεριφορά της οθόνης του παραθύρου: Το 0 διατηρεί το παράθυρο ανοιχτό μέχρι να πατηθεί ένα πλήκτρο, ενώ το 1 εμφανίζει το πλαίσιο για τουλάχιστον 1 ms πριν κλείσει αυτόματα.

Τέλος, ορίζουμε την συνθήκη τερματισμού για τον βρόχο `while`, απελευθερώνουμε τους πόρους της κάμερας web χρησιμοποιώντας το `video.release()` και κλείνουμε όλα τα ανοιχτά παράθυρα χρησιμοποιώντας το `cv2.destroyAllWindows()`. Αυτό διασφαλίζει την ομαλή ολοκλήρωση της αίτησής μας και τη σωστή αποδέσμευση των πόρων του συστήματος.

4.4.2 Υλοποίηση κώδικα αποστολής email

Για τη συνάρτηση της αποστολής email θα πρέπει να φτιάξουμε ένα καινούριο αρχείο της python. Θα έχει το όνομα της συνάρτησης που θα καλέσουμε μέσα από το βασικό αρχείο της ανίχνευσης κίνησης.

Για τη δημιουργία του συγκεκριμένου αρχείου θα προσθέσουμε κάτι γνώριμο από την αναφορά μας στο προηγούμενο υπό-κεφάλαιο (2,1) με την παρακάτω εντολή

```
# -*- coding: <encoding name> -*-
```

Στη συνέχεια θα ακολουθήσουμε την ίδια λογική με τη δήλωση των βιβλιοθηκών και την αρχικοποίηση των απαραίτητων μεταβλητών μας. Στο συγκεκριμένο αρχείο τα πράγματα είναι κάπως πιο απλά. Θα εισάγουμε την βιβλιοθήκη os η οποία μας δίνει πρόσβαση στα αρχεία μας. Η os στην Python παρέχει λειτουργίες για την αλληλεπίδραση με το λειτουργικό σύστημα. Το λειτουργικό σύστημα υπάγεται στις τυπικές βοηθητικές μονάδες της Python. Αυτή η ενότητα παρέχει έναν φορητό τρόπο χρήσης λειτουργιών που εξαρτώνται από το λειτουργικό σύστημα. Οι λειτουργικές μονάδες *os* και *os.path* περιλαμβάνουν πολλές λειτουργίες για αλληλεπίδραση με το σύστημα αρχείων. Λειτουργίες λειτουργικής μονάδας Python OS [38]. Εδώ θα συζητήσουμε μερικές σημαντικές λειτουργίες της ενότητας Python os:

- Χειρισμός του τρέχοντος καταλόγου εργασίας
- Δημιουργία καταλόγου
- Καταχώριση αρχείων και καταλόγων με Python
- Διαγραφή καταλόγου ή αρχείων με χρήση Python

Χειρισμός του τρέχοντος καταλόγου εργασίας

Θεωρούμε το Current Working Directory (CWD) ως φάκελλο, όπου λειτουργεί η Python. Κάθε φορά που τα αρχεία καλούνται μόνο με το όνομά τους, η Python υποθέτει ότι ξεκινά από το CWD, πράγμα που σημαίνει ότι η αναφορά μόνο με όνομα θα είναι επιτυχής μόνο εάν το αρχείο βρίσκεται στο CWD της Python. Σημείωση: Ο φάκελος όπου εκτελείται το σενάριο Python είναι γνωστός ως Τρέχων Κατάλογος.

Στη συνέχεια θα κάνουμε χρήση κάποιων εργαλείων που είναι απαραίτητα για την μορφοποίηση του email μας. Στην ουσία θα χρησιμοποιήσουμε κάποιες βιβλιοθήκες για να μπορέσουμε να προσθέσουμε την εικόνα που θα μας αποσταλεί στο email που θα ορίσουμε καθώς και ένα κείμενο μαζί με το θέμα του email. Αυτό που θα χρειαστούμε είναι να

εισάγουμε το MIMEText, το MIMEImage και το MIMEMultipart . Οι γραμμές κώδικα που θα χρησιμοποιήσουμε είναι οι ακόλουθες:

```
from email.mime.text import MIMEText
```

```
from email.mime.image import MIMEImage
```

```
from email.mime.multipart import MIMEMultipart
```

Τελειώνοντας με τις αρχικοποιήσεις έχουμε μόνο να δηλώσουμε κάποιες μεταβλητές που αφορούν τα στοιχεία του email που θα συνδεθεί(αποστολέας) και το email που θα λάβει(παραλήπτης) καθώς και το μονοπάτι(path) για τον εντοπισμό του αρχείου του οποίου θα στείλουμε.

Οι μεταβλητές αυτές είναι οι AddressF, διεύθυνση αποστολέα και το password. Εδώ οφείλουμε να επιστήσουμε την προσοχή διότι δεν θέλουμε το 2FA του email μας να είναι ενεργοποιημένο. Η μεταβλητή addressTo η διεύθυνση δηλαδή του παραλήπτη καθώς και η μεταβλητή path η διαδρομή στην οποία βρίσκεται το αρχείο που θα επισυνάψουμε.

Μετά την δήλωση των μεταβλητών και των βιβλιοθηκών μας θα προχωρήσουμε στη δημιουργία της συνάρτησης μας η οποία θα στείλει το email ειδοποίησης μαζί με την εικόνα όπως την έχουμε διαμορφώσει. Η συνάρτηση αυτή θα κληθεί μαζί με ένα όρισμα, μια μεταβλητή η οποία θα είναι το όνομα του αρχείου που θέλουμε να στείλουμε. Με την παρακάτω εντολή θα καλέσουμε τη συνάρτησή μας μέσα από το πρώτο αρχείο του κώδικα που έχουμε φτιάξει.

```
def send_email(FN):
```

Παρακάτω χρησιμοποιούμε μία μεταβλητή στην οποία αποθηκεύουμε το πλήρες όνομα του αρχείου μας. Η μεταβλητή αυτή είναι τύπου string και προσθέτουμε τη μεταβλητή path+FN που είναι το όρισμά μας. Έπειτα διαβάζουμε το αρχείο μας με την χρήση της βιβλιοθήκης os που αναφερθήκαμε παραπάνω. Η εντολή που πρέπει να γράψουμε είναι η img_data=open(FIn,'rb').read(). Το rb(read binary) σημαίνει ότι ανοίγουμε το αρχείο μας σε δυαδική μορφή (binary mode) δηλαδή μηδέν(0) και ένα(1) και με το .read() το διαβάζουμε σε αυτή τη μέθοδο.

Στη συνέχεια ορίζουμε ένα αντικείμενο για να δομήσουμε το μήνυμά μας με τη χρήση των εργαλείων που αναφέραμε πιο πάνω. Με λίγα λόγια θα ορίσουμε το αντικείμενο να έχει τη δομή του email μας που θα αποτελείται από πολλά «κομμάτια». Επομένως θα δώσουμε την εντολή msg = MIMEMultipart(). Έπειτα θα δώσουμε θέμα στο email μας με την εντολή msg['Subject']='Ανιχνεύθηκε κίνηση'.

Ως επόμενες γραμμές κώδικα θα πρέπει να δηλώσουμε τη διεύθυνση του αποστολέα και του παραλήπτη ως εξής:

Msg['From']= adressF δηλαδή τη διεύθυνση του αποστολέα

Msg['To']=addressTo όπου είναι η διεύθυνση του παραλήπτη.

Ως επόμενο βήμα θα ορίσουμε το κείμενο του μηνύματος μας και το ενσωματώνουμε στο email μας με την εντολή `attach()`. Η εντολή για την ενσωμάτωση του κειμένου μας θα γίνει με τον εξής τρόπο,

`Msg.attach(text)` όπου `text` μία μεταβλητή στην οποία βάζουμε το κείμενο με το οποίο θέλουμε να στείλουμε απ' το σώμα του κειμένου στο email μας.

Με αντίστοιχο τρόπο θα ενσωματώσουμε στο μήνυμά μας και την εικόνα όμως πρώτα πρέπει να χρησιμοποιήσουμε την υποκλάση της `MIMENonMultipart` την `MIMEImage` η οποία χρησιμοποιείται για τη δημιουργία μηνύματος αντικειμένων βασικού τύπου `image._imagedata` που περιέχουν τα Bytes για τη γραμμή των δεδομένων της εικόνας. Εάν αυτός ο τύπος δεδομένων μπορεί να ανιχνευθεί (`jpeg, png, gif, tiff, rgb, pbm, pgm, ppm, rast, xbm, bmp, webp` και `exr attempted`), τότε ο υποτύπος θα συμπεριληφθεί αυτόματα στην επικεφαλίδα `Content-Type`. Διαφορετικά, μπορείτε να καθορίσετε ρητά τον υποτύπο της εικόνας μέσω του όρου `_subtype`. Εάν ο δευτερεύων τύπος δεν μπόρεσε να μαντέψει και δεν δόθηκε το `_subtype`, τότε δημιουργείται `TypeError`.

Εντολή που θα χρησιμοποιήσουμε είναι πιο κάτω

`image = MIMEImage(img_data, name=os.path.basename(FN))`

όπου `img_data` είναι τα δεδομένα που διαβάσαμε από αρχείο με τη χρήση της `open`. Τώρα δεν μας μένει παρά να την κάνουμε `attach` όπως κάναμε με τον ίδιο τρόπο πιο πριν το κείμενο δηλαδή σαν επόμενη γραμμή στον κώδικά μας θα γράψουμε `msg.attach(image)`.

Τέλος θα χρησιμοποιήσουμε την `try except`, δηλαδή ένα χειρισμό εξαιρέσεων. Όταν συμβαίνει ένα σφάλμα ή μια εξαίρεση, όπως την αποκαλούμε, η Python θα σταματήσει κανονικά και θα δημιουργήσει ένα μήνυμα σφάλματος.

Αυτές οι εξαιρέσεις μπορούν να αντιμετωπιστούν με τη χρήση της εντολής `try`. Με άλλα λόγια δηλαδή, εάν κάτι δεν πάει καλά με τις εντολές που θα χρησιμοποιήσουμε μέσα στην `try` για να στείλουμε το email μας θα λάβουμε απ το `except` ένα μήνυμα σφάλματος.

Μέσα στην `try` θα δοκιμάσουμε να συνδεθούμε με τον εξυπηρετητή `smtp` του `gmail`. Η εντολή που θα χρησιμοποιήσουμε είναι `srv=smtplib.SMTP('smtp.gmail.com',587)`

Στην κλάση της Python `smtp` που χρησιμοποιούμε δηλώνουμε το `hostname` που στην περίπτωση μας είναι ο `host` του `gmail` και την θύρα(`port`) `587` που θα χρησιμοποιήσει για να συνδεθεί. Έπειτα θα χρησιμοποιήσουμε την εντολή `srv.ehlo()`. `Ehlo()` είναι μια μέθοδος που μας χρησιμεύει να ταυτοποιήσουμε το όνομα του κεντρικού υπολογιστή.

Στη συνέχεια θα χρησιμοποιήσουμε μία άλλη μέθοδο την `starttls()` με τον ίδιο τρόπο που γράψαμε την εντολή της `ehlo()`. Η `starttls()` θέτει τη σύνδεση με τον διακομιστή SMTP σε λειτουργία TLS. Έπειτα ακολουθεί η μέθοδος `login(user, password)` με αυτή τη μέθοδο θα συνδεθούμε σε έναν διακομιστή SMTP που απαιτεί έλεγχο ταυτότητας. Η εντολή που θα χρησιμοποιήσουμε είναι η `srv.login(addressF, password)` όπου `address` είναι η διεύθυνση του αποστολέα και όπου `password` ο κωδικός που δημιουργούμε μέσω του gmail για συσκευές όπως το raspberry με χρήση G-Mail and less secure sign-in technology. Αυτό αναφέρεται στο κεντρικό φόρουμ της `raspberrypi.com`

Ως επόμενο βήμα θα πρέπει να χρησιμοποιήσουμε τη μέθοδο `sendmail()` η οποία δέχεται τρεις παραμέτρους η πρώτη είναι ο αποστολέα, η δεύτερη ο παραλήπτης και τέλος το μήνυμα, όπως το διαμορφώσαμε στο αντικείμενο `msg` πιο πάνω. Αυτό επιτυγχάνεται με την παρακάτω εντολή.

`srv.sendmail(addressF, addressTo, msg.as_string())`

Όταν το μήνυμα μας σταλεί κλείνουμε την σύνδεση μας με τον εξυπηρετητή `smtp` κάνοντας τη χρήση της μεθόδου `quit()` και δίνοντας την εντολή `srv.quit()`³³. Εάν θέλουμε για την καλύτερη ενημέρωσή μας μετά την εντολή `quit` να κάνουμε χρήση της κλάσης `print()` ώστε να τυπώσουμε ένα μήνυμα ότι το email μας έχει αποσταλεί. Στην συνέχεια μέσα στην `except` θα βάλουμε μία εντολή της κλάσης `print()` ότι κάτι δεν πήγε καλά με την αποστολή του email μας. Σε αυτό το σημείο φτάσαμε στο τέλος και του δεύτερου αρχείου με τον κώδικα που έπρεπε να υλοποιήσουμε ώστε να δημιουργήσουμε την εφαρμογή ανίχνευσης κίνησης μέσω κάμερας και αποστολή της ανιχνευμένης εικόνας σε email.

Αφού ολοκληρώσουμε τον κώδικα μας και σιγουρευτούμε ότι το πρόγραμμά μας δουλεύει σωστά, μπορούμε να εκτελέσουμε σε τερματικό(`terminal`) τις παρακάτω εντολές

`sudo crontab -e`

`@reboot YOUR_SHELL_COMMAND &`

ώστε να ξεκινάει να εκτελείται το πρόγραμμά μας με το ξεκίνημα του raspberry pi μας.

Όπου `YOUR_SHELL_COMMAND` βάζουμε το πλήρες `path` του αρχείου που θέλουμε να εκτελείται κατά την επανεκκίνηση του raspberry μας καθώς και την εντολή που θα εκτελεί το πρόγραμμα. Για παράδειγμα στην περίπτωσή μας η εντολή θα είναι η ακόλουθη `@reboot python3 /home/pi/project/detect_motion.py &`

Ως μία μικρή αναφορά θα μπορούσαμε να πούμε ότι εάν στο raspberry pi μας προσθέσουμε ένα ανιχνευτή κίνησης PIR ή ένα καλώδιο που να ενεργοποιεί ένα συναγερμό μπορούμε να αποτρέψουμε κάποιον να μην εισέλθει στον απαγορευμένο χώρο τον οποίο

ελέγχουμε με την κάμερα. Επιπλέον με τη χρήση των διαφόρων πλατφορμών IoT για παράδειγμα η blynk, μπορούμε να δημιουργήσουμε ένα συμβάν (trigger) ώστε να μας έρθει ειδοποίηση μέσω της εκάστοτε πλατφόρμας(πχ blynk). Στην παρούσα εργασία στέλνουμε εμείς το email αλλά μπορούμε να προσθέσουμε στον κώδικα μας ένα σύνολο εντολών που να ενεργοποιεί μια θύρα εξόδου από το Raspberry μας σε κάποιον ακροδέκτη ενός συνδεδεμένου ηχείου ή συναγερμού τον οποίο θα έχουμε επιλέξει εμείς. Αυτή η διαδικασία γίνεται μέσω του GPIO του Raspberry pi ενεργοποιώντας μία θύρα για παράδειγμα την 17. Κάτι δηλαδή σαν ενεργοποίηση ενός interrupt με σκοπό την αποτροπή της εισόδου ενός ατόμου στον επιβλέποντα χώρο.

Η συγκεκριμένη διαδικασία γίνεται μέσω του πρωτοκόλλου MQTT(Message Queue Telemetry Transport). Αυτό το πρωτόκολλο είναι ένα πολύ ελαφρύ πρωτόκολλο που έχει σχεδιαστεί για τη μετάδοση δεδομένων τηλεμετρίας μέσω μηνυμάτων και είναι κατάλληλο για δίκτυα με χαμηλό εύρος. Με τη χρήση του συγκεκριμένου πρωτοκόλλου όταν ανιχνευθεί η διαφορά των frame στην εικόνα θα στείλουμε και την εντολή μέσω του MQTT να αλλάξει η τιμή στο pin(X) του GPIO του Raspberry PI μας. Όπου X ο αριθμός του pin που θέλουμε να ενεργοποιήσουμε πχ 17. Εκτός του επιπλέον κώδικα που πρέπει να γράψουμε πρέπει να δημιουργήσουμε και ένα λογαριασμό στο blynk.cloud και ένα νέο template με τη συσκευή μας, ώστε να μας δώσει έναν κωδικό για τη συσκευή που θα βλέπει online η πλατφόρμα.

5. Συλλογή δεδομένων και ανάλυση

5.1 Διαδικασία συλλογής δεδομένων

Στο πλαίσιο του έργου επιτήρησης κάμερας, οι διαδικασίες συλλογής δεδομένων είναι θεμελιώδεις για την απόκτηση σχετικών πληροφοριών σχετικά με εντοπισμένα συμβάντα κίνησης και ενημερώσεις κατάστασης συστήματος. Η ενσωμάτωση διάφορων μηχανισμών συλλογής δεδομένων, συμπεριλαμβανομένης της κάμερας PIR, του email και του Blynk, εξασφαλίζει ολοκληρωμένες δυνατότητες παρακολούθησης και ανάλυσης.

- Μέσω κάμερας PIR

Ο κύριος μηχανισμός συλλογής δεδομένων περιστρέφεται γύρω από τη χρήση της μονάδας κάμερας Παθητικής Υπέρυθρης ακτινοβολίας (PIR) που είναι ενσωματωμένη στην υποδομή επιτήρησης. Η κάμερα PIR λειτουργεί ανιχνεύοντας αλλαγές στην υπέρυθρη ακτινοβολία που εκπέμπεται από αντικείμενα εντός του οπτικού της πεδίου, επιτρέποντας έτσι την αναγνώριση γεγονότων κίνησης. Κατά την ανίχνευση κίνησης, η κάμερα τραβάει στιγμιότυπα εικόνων, παρέχοντας οπτική απόδειξη της δραστηριότητας που ανιχνεύτηκε.

Αυτά τα στιγμιότυπα εικόνων χρησιμεύουν ως κύρια σημεία δεδομένων για μετέπειτα σκοπούς ανάλυσης και αρχειοθέτησης [39].

- Μέσω ηλεκτρονικού ταχυδρομείου

Το email χρησιμεύει ως βοηθητικό κανάλι συλλογής δεδομένων, διευκολύνοντας τη μετάδοση κρίσιμων πληροφοριών και στιγμιότυπων εικόνων σε καθορισμένους παραλήπτες. Κατά την ανίχνευση συμβάντων κίνησης, το σύστημα αποστέλλει αμέσως ειδοποιήσεις email που περιέχουν τα στιγμιότυπα που τραβήχτηκαν, υποδηλώνοντας την ασυμφωνία μεταξύ των προηγούμενων και των επόμενων καρτέ. Αυτή η στιγμιαία διάδοση πληροφοριών δίνει τη δυνατότητα στους ενδιαφερόμενους να μένουν ενημερωμένοι για τις ανωμαλίες που ανιχνεύονται σε πραγματικό χρόνο, διευκολύνοντας τους μηχανισμούς άμεσης απόκρισης και τις διαδικασίες λήψης αποφάσεων [40].

- Μέσω Blynk

Η ενσωμάτωση της πλατφόρμας Blynk ενισχύει τις δυνατότητες συλλογής δεδομένων παρέχοντας μια κεντρική διεπαφή για την παρακολούθηση και τον έλεγχο του συστήματος επιτήρησης από απόσταση. Μέσω της εφαρμογής Blynk, οι χρήστες μπορούν να έχουν πρόσβαση σε ροές δεδομένων σε πραγματικό χρόνο και να λαμβάνουν ειδοποιήσεις σχετικά με συμβάντα κίνησης που έχουν εντοπιστεί. Επιπλέον, το Blynk επιτρέπει στους χρήστες να αλληλεπιδρούν με το σύστημα επιτήρησης, επιτρέποντας δυναμικές προσαρμογές και αλλαγές διαμόρφωσης ανάλογα με τις ανάγκες. Αυτή η δυνατότητα αμφίδρομης επικοινωνίας δίνει τη δυνατότητα στους ενδιαφερόμενους να ασχοληθούν ενεργά με την υποδομή επιτήρησης, ενισχύοντας την επίγνωση της κατάστασης και τη λειτουργική αποτελεσματικότητα [41].

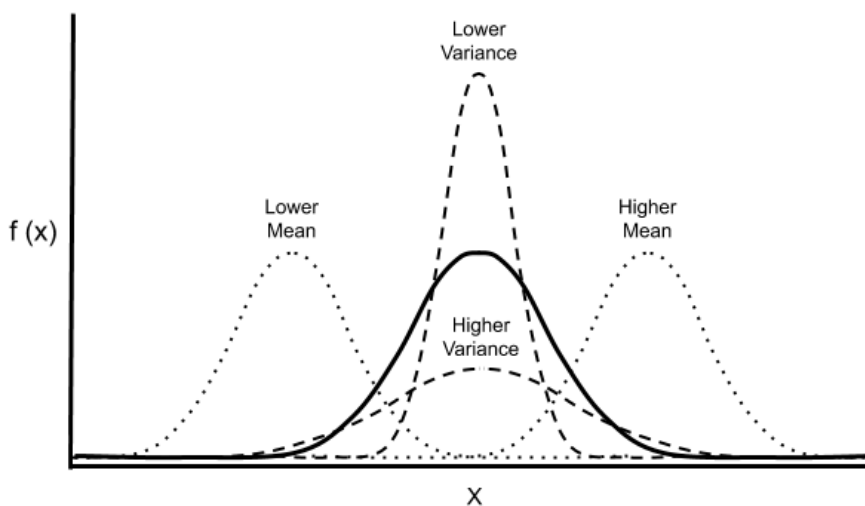
5.2 Τεχνικές ανάλυσης των δεδομένων

Η αξιοποίηση προηγμένων αναλυτικών μεθοδολογιών ενισχύει την ικανότητά μας να διακρίνουμε μοτίβα, ανωμαλίες και τάσεις στα δεδομένα επιτήρησης, επιτρέποντας τη λήψη τεκμηριωμένων αποφάσεων και τους μηχανισμούς προληπτικής απόκρισης.

Ανάλυση και Επεξεργασία Δεδομένων μέσω Τεχνικών Gaussian

Οι τεχνικές Gauss αντιπροσωπεύουν μια εξελιγμένη προσέγγιση στην ανάλυση και επεξεργασία δεδομένων, ιδιαίτερα κατάλληλη για την εξαγωγή σχετικών πληροφοριών από θορυβώδη ή πολύπλοκα σύνολα δεδομένων. Οι διεργασίες Gauss, που έχουν τις ρίζες τους στην πιθανολογική μοντελοποίηση, επιτρέπουν τον χαρακτηρισμό της αβεβαιότητας και την εκτίμηση των υποκείμενων προτύπων μέσα στα δεδομένα. Με τη μοντελοποίηση της

κατανομής των παρατηρούμενων σημείων δεδομένων χρησιμοποιώντας κατανομές Gaussian, μπορούμε να συμπεράνουμε πολύτιμες πληροφορίες για τη συμπεριφορά του περιβάλλοντος επιτήρησης [42], [43].



Εικόνα 14 Gaussian analysis

Στο πλαίσιο μας, οι τεχνικές Gauss μπορούν να εφαρμοστούν για την ανάλυση και την επεξεργασία δεδομένων εικόνων που καταγράφονται από τις κάμερες παρακολούθησης. Με τη μοντελοποίηση των εντάσεων των εικονοστοιχείων εντός των λαμβανόμενων εικόνων ως κατανομές Gauss, μπορούμε να αναγνωρίσουμε περιοχές ενδιαφέροντος που αντιστοιχούν σε συμβάντα κίνησης που ανιχνεύθηκαν. Επιπλέον, τεχνικές φιλτραρίσματος Gauss μπορούν να χρησιμοποιηθούν για τη βελτίωση της καθαρότητας της εικόνας και τη μείωση του θορύβου, βελτιώνοντας την ακρίβεια της επακόλουθης ανάλυσης και ερμηνείας.

Ανιχνευτής κίνησης

Ο ανιχνευτής κίνησης αντιπροσωπεύει ένα βασικό συστατικό του πλαισίου ανάλυσης δεδομένων μας, επιτρέποντας την ανίχνευση και τον χαρακτηρισμό γεγονότων κίνησης σε πραγματικό χρόνο μέσα στο περιβάλλον επιτήρησης. Αξιοποιώντας αλγόριθμους ανίχνευσης κίνησης, όπως αφαίρεση φόντου ή ανάλυση οπτικής ροής, ο ανιχνευτής κίνησης προσδιορίζει χωρικές και χρονικές αλλαγές στη ροή βίντεο που καταγράφηκε, ενδεικτικές κίνησης ή δραστηριότητας [44].

Κατά την ανίχνευση συμβάντων κίνησης, ο ανιχνευτής κίνησης ενεργοποιεί τη λήψη στιγμιότυπων εικόνων, διευκολύνοντας την επακόλουθη ανάλυση και την αρχειοθέτηση των σχετικών δεδομένων. Επιπλέον, ο ανιχνευτής κίνησης χρησιμεύει ως πρόδρομος για πιο προηγμένες αναλύσεις, επιτρέποντας την ταξινόμηση και την κατηγοριοποίηση των ανιχνευόμενων γεγονότων με βάση προκαθορισμένα κριτήρια. Ενσωματώνοντας τον ανιχνευτή κίνησης στη γραμμή ανάλυσης δεδομένων μας, μπορούμε να παρακολουθούμε και να αναλύουμε αποτελεσματικά τα δεδομένα επιτήρησης, διακρίνοντας σχετικά μοτίβα και ανωμαλίες με ακρίβεια και αποτελεσματικότητα [45], [23].

5.3 Παρουσίαση των αποτελεσμάτων

Τα συμπεράσματα που εξήχθησαν σε αυτή τη μελέτη προέκυψαν μέσω συστηματικής ανάλυσης δεδομένων και παρατηρήσεων που συλλέχθηκαν χρησιμοποιώντας την πλατφόρμα Cayenne. Αξίζει να σημειωθεί ότι όλες οι πτυχές της έρευνας, συμπεριλαμβανομένης της συλλογής δεδομένων, της ανάλυσης και των συμπερασμάτων, διεξήχθησαν αποκλειστικά στο περιβάλλον Cayenne, το οποίο χρησίμευσε ως το πρωταρχικό πλαίσιο για την παρακολούθηση και την αξιολόγηση της απόδοσης του συστήματος επιτήρησης.



Εικόνα 15 Cayenne platform

1. Συλλογή δεδομένων

Οι διαδικασίες συλλογής δεδομένων περιελάμβαναν τη συνεχή παρακολούθηση και καταγραφή των δραστηριοτήτων του συστήματος επιτήρησης, συμπεριλαμβανομένης της ανίχνευσης κίνησης, των ειδοποιήσεων μέσω email και της ζωντανής ροής, μέσω του διαδικτυακού πίνακα ελέγχου της Cayenne. Αισθητήρες και συσκευές που συνδέονται με τις πλακέτες Raspberry Pi ή Arduino διευκόλυναν την απόκτηση δεδομένων σε πραγματικό χρόνο σχετικά με συμβάντα ασφαλείας και αποκρίσεις συστήματος.

2. Ανάλυση και Ερμηνεία

Χρησιμοποιώντας τα προσαρμόσιμα widget που παρέχονται από την πλατφόρμα Cayenne, χρησιμοποιήθηκαν διάφορες τεχνικές οπτικοποίησης δεδομένων για την ανάλυση

τάσεων, προτύπων και ανωμαλιών στα δεδομένα επιτήρησης. Το Cayenne Online Dashboard χρησίμευσε ως κεντρικός κόμβος για την οπτικοποίηση δεδομένων της συσκευής, τον καθορισμό κανόνων και τον προγραμματισμό συμβάντων, επιτρέποντας ολοκληρωμένη ανάλυση της συμπεριφοράς και της απόδοσης του συστήματος.

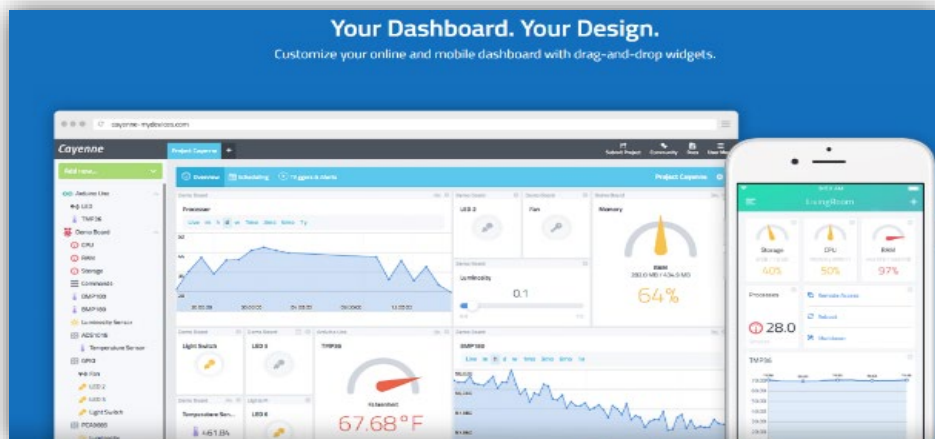
3. Συμπεράσματα

Με βάση τις γνώσεις που αποκτήθηκαν από την ανάλυση δεδομένων, εξήχθησαν συμπεράσματα σχετικά με την αποτελεσματικότητα, την ανταπόκριση και τη λειτουργική δυναμική του συστήματος επιτήρησης. Βασικοί δείκτες απόδοσης, όπως η ακρίβεια ανίχνευσης κίνησης, ο λανθάνων χρόνος ειδοποίησης μέσω email και η καθυστέρηση ζωντανής ροής αξιολογήθηκαν για την αξιολόγηση της συνολικής αξιοπιστίας και καταλληλότητας του συστήματος για εφαρμογές παρακολούθησης ασφάλειας.

4. Αναγνώριση για το κλείσιμο της πλατφόρμας

Είναι σημαντικό να αναγνωρίσουμε ότι η μελέτη διεξήχθη πριν από το κλείσιμο της πλατφόρμας Cayenne. Παρά τη διακοπή λειτουργίας της πλατφόρμας, οι γνώσεις και τα συμπεράσματα που προκύπτουν από την ανάλυση δεδομένων παραμένουν χρήσιμα και πολύτιμα για την ενημέρωση των μελλοντικών προσπαθειών έρευνας και ανάπτυξης στον τομέα των τεχνολογιών επιτήρησης.

Η Cayenne αντιπροσωπεύει μια πρωτοποριακή προσέγγιση στην ανάπτυξη έργων Internet of Things (IoT), προσφέροντας μια φιλική προς τον χρήστη διεπαφή για τη δημιουργία προτύπων και την κοινή χρήση έργων συνδεδεμένων συσκευών. Επίσης, η Cayenne εξουσιοδοτεί τους προγραμματιστές, τους σχεδιαστές και τους μηχανικούς να σχεδιάζουν, να δημιουργούν και να αναπτύσσουν λύσεις IoT με ευκολία.



Εικόνα 16 Cayenne Dashboard

Κύρια στοιχεία της πλατφόρμας:

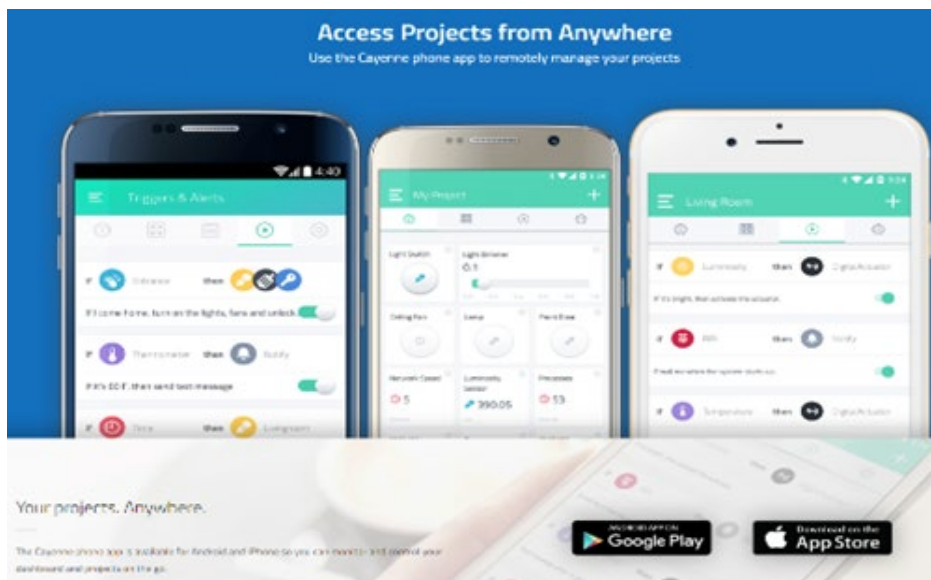
Το Cayenne Online Dashboard χρησιμεύει ως κεντρικός κόμβος για τη διαχείριση έργων IoT, παρέχοντας προσαρμόσιμα widget για οπτικοποίηση δεδομένων, ρύθμιση κανόνων, προγραμματισμό συμβάντων και πολλά άλλα. Οι χρήστες μπορούν να συνδέσουν απρόσκοπτα αισθητήρες, ενεργοποιητές και άλλα εξαρτήματα υλικού σε υπολογιστές μιας πλακέτας, μικροελεγκτές και άλλες υποστηριζόμενες συσκευές.

Απαιτήσεις υλικού και προγράμματος περιήγησης:

Για να αξιοποιήσουν τις πλήρεις δυνατότητες της Cayenne, οι χρήστες απαιτούν συμβατό υλικό, όπως συσκευές Raspberry Pi, Arduino ή LoRa συνδεδεμένες στο Διαδίκτυο ή πύλη. Επιπλέον, η Cayenne είναι βελτιστοποιημένη για να λειτουργεί σε δημοφιλή προγράμματα περιήγησης ιστού, με σχέδια για υποστήριξη Progressive Web App (PWA) σε πλατφόρμες iOS και Android στο εγγύς μέλλον.

Απομακρυσμένη παρακολούθηση & έλεγχος

Τα widgets του πίνακα ελέγχου διευκολύνουν τη διαχείριση έργου μέσω της διαισθητικής αλληλεπίδρασης με τον χρήστη. Οι χρήστες μπορούν να ενεργοποιήσουν ή να απενεργοποιήσουν γρήγορα συσκευές, όπως φώτα, χρησιμοποιώντας widget κουμπιών. Επιπλέον, οι ρυθμίσεις του θερμοστάτη για τον έλεγχο της θερμοκρασίας απλοποιούνται μέσω γραφικών στοιχείων κουμπιών. Επιπλέον, οι χρήστες μπορούν να διαμορφώνουν την ένταση του φωτός χρησιμοποιώντας γραφικά ρυθμιστικά, μεταξύ άλλων διαθέσιμων λειτουργιών.



Εικόνα 17 Απομακρυσμένη παρακολούθηση και έλεγχος

Περαιτέρω χαρακτηριστικά και δυνατότητες της πλατφόρμας Cayenne διευκρινίζονται αναλυτικά στο Παράρτημα Δ.

Αποτελέσματα πειράματος

Ο πρωταρχικός στόχος αυτού του πειράματος ήταν η αξιολόγηση της αποτελεσματικότητας του συστήματος σε διάφορα σενάρια χρήσης, τα οποία δοκιμάστηκαν συστηματικά. Συγκεκριμένα, το πείραμα επικεντρώθηκε στον έλεγχο των δυνατοτήτων ανίχνευσης κίνησης του συστήματος κάτω από διαφορετικές συνθήκες φωτισμού εντός της περιοχής παρατήρησης.

Τα σενάρια περιλάμβαναν μια ολοκληρωμένη σειρά συνθηκών φωτισμού για να μετρηθεί με ακρίβεια η απόκριση του συστήματος. Το πείραμα αξιολόγησε μεθοδικά την απόδοση του συστήματος σε ένα φάσμα επιπέδων φωτεινότητας, όπως απεικονίζεται στον ακόλουθο πίνακα:

ΑΠΟΤΕΛΕΣΜΑ ΜΕ ΦΩΤΕΙΝΟΤΗΤΑ	ΜΕΓΕΘΟΣ ΑΡΧΕΙΟΥ	ΑΝΙΧΝΕΥΣΗ ΚΙΝΗΣΗΣ
ΠΑΡΑ ΠΟΛΥ ΥΨΗΛΗ	30-50B	ΟΧΙ
ΥΨΗΛΗ	70-80KB	ΝΑΙ
ΛΙΓΟΤΕΡΟ ΥΨΗΛΗ	100-120KB	ΝΑΙ

ΜΕΤΡΙΑ	98KB	ΝΑΙ
ΕΛΑΧΙΣΤΗ	91 KB	ΝΑΙ
ΚΑΘΟΛΟΥ	ΔΕΝ ΥΠΑΡΧΕΙ ΑΡΧΕΙΟ	ΟΧΙ

Κάθε σενάριο παρέιχε ξεχωριστές πληροφορίες για τη συμπεριφορά και την απόκριση του συστήματος σχετικά με την ανίχνευση κίνησης. Μεταβάλλοντας συστηματικά τα επίπεδα φωτεινότητας, το πείραμα είχε ως στόχο να διευκρινίσει πώς η απόδοση του συστήματος συσχετίζεται με διαφορετικές συνθήκες φωτισμού.

Συμπερασματικά, η ανάλυση αποκαλύπτει σημαντικές συσχετίσεις μεταξύ της φωτεινότητας της εικόνας και της απόδοσης ανίχνευσης κίνησης. Παρατηρήθηκε ότι καθώς η φωτεινότητα της εικόνας εντείνεται, το μέγεθος του αρχείου μειώνεται. Ωστόσο, το σύστημα αποτυγχάνει να ανιχνεύσει την κίνηση καθώς η εικόνα πλησιάζει τον κορεσμό. Αντίθετα, σε πιο σκοτεινές συνθήκες, το σύστημα αντιμετωπίζει προκλήσεις στην ανίχνευση κίνησης και τη λήψη εικόνας.

Υπό κανονικές συνθήκες φωτισμού, οι εικόνες ευνοούν την ανίχνευση κίνησης, συνήθως μεγέθους 120 KB κατά μέσο όρο με ανάλυση 640X480. Για την αντιμετώπιση των προκλήσεων που δημιουργούνται από περιβάλλοντα χαμηλού φωτισμού, οι κάμερες ασφαλείας συχνά ενσωματώνουν τεχνολογία υπέρυθρων. Επιπλέον, σημειώθηκε ότι οι εικόνες υψηλότερης ανάλυσης προσφέρουν ανώτερη ποιότητα. Ωστόσο, απαιτούν μεγαλύτερη κατανομή αποθηκευτικού χώρου, που κυμαίνεται από kilobyte έως megabyte

σε σύγκριση με αντίστοιχα χαμηλότερης ανάλυσης.

Enter Settings



Digital Motion Sensor
Motion detector

[View Datasheet](#)

Please fix the following errors:

- Please turn on the device

Widget Name

Digital Motion Sensor

Select Device



Raspberry Pi



Connectivity

Integrated GPIO



Channel



Choose Widget



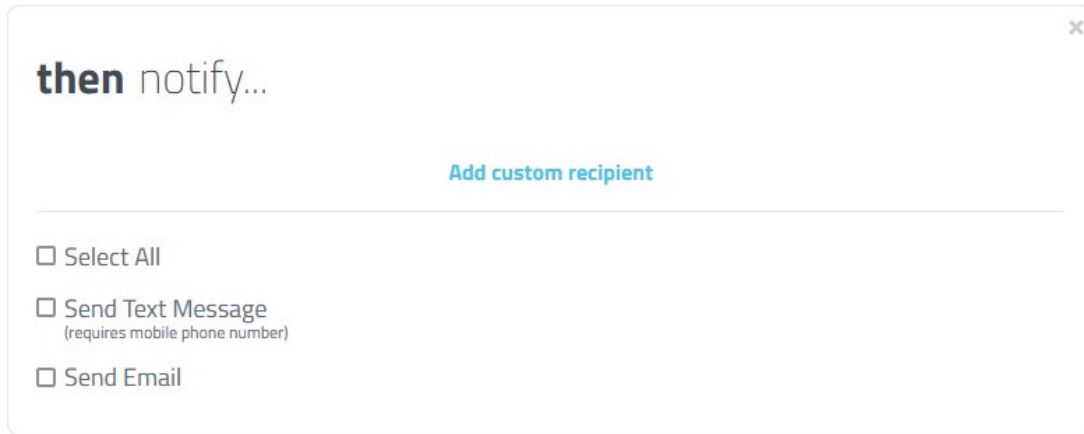
0/1 2 state



Invert Logic

Add Sensor

Εικόνα 18 Προσθήκη motion sensor μέσω cayenne



Εικόνα 19 Επιλογές για Alert

6. Συμπεράσματα και μελλοντικές αλλαγές

6.1 Σύνοψη των ευρημάτων

Βάσει τα ευρημάτων και παρατηρήσεων που προέκυψαν από τη μελέτη εξάγονται βασικά συμπεράσματα σχετικά με την αποτελεσματικότητα και τη λειτουργικότητα του υπό διερεύνηση συστήματος επιτήρησης:

- Αποτελεσματική συμπίεση αρχείων

Η ανάλυση δείχνει ότι τα αρχεία που δημιουργούνται από το σύστημα επιτήρησης, ακόμη και αυτά υψηλής ανάλυσης, διατηρούν ένα συμπαγές μέγεθος, με εγγραφές έως και 100 KB. Αυτή η αποτελεσματικότητα στη διαχείριση μεγέθους αρχείου εξασφαλίζει ελάχιστες απαιτήσεις αποθήκευσης, ιδιαίτερα όταν μεταδίδεται μέσω email.

- Ειδοποιήσεις ηλεκτρονικού ταχυδρομείου σε πραγματικό χρόνο

Το σύστημα επιδεικνύει ταχύτητα στην αποστολή και λήψη ειδοποιήσεων μέσω email, διευκολύνοντας την ταχεία διάδοση κρίσιμων πληροφοριών που σχετίζονται με εντοπισμένες ανωμαλίες ή παραβιάσεις ασφαλείας.

- Ελάχιστη καθυστέρηση στη ζωντανή ροή

Παρά τους εγγενείς περιορισμούς της τεχνολογίας ζωντανής ροής, το σύστημα επιτήρησης παρουσιάζει αξιόπαινη απόδοση με αμελητέα καθυστέρηση περίπου 10 δευτερολέπτων από το βίντεο ζωντανής ροής σε πραγματικό χρόνο, διασφαλίζοντας δυνατότητες έγκαιρης παρακολούθησης και απόκρισης.

- Προκλήσεις επεκτασιμότητας

Η μελέτη υπογραμμίζει την ανάγκη προσεκτικής εξέτασης σχετικά με την επεκτασιμότητα του συστήματος επιτήρησης, ιδιαίτερα σε σενάρια όπου απαιτείται η παρακολούθηση πολλαπλών σημείων. Ο περιορισμένος αριθμός καμερών μπορεί να δημιουργήσει προκλήσεις για την επίτευξη ολοκληρωμένης κάλυψης επιτήρησης σε μεγαλύτερες περιοχές ή πολλαπλούς ιστότοπους.

- Βελτιστοποιημένη συλλογή δεδομένων:

Με έμφαση στη διατήρηση του χώρου αποθήκευσης και στον εξορθολογισμό της διαχείρισης δεδομένων, το σύστημα δίνει προτεραιότητα στη συλλογή αρχείων κίνησης μικρού μεγέθους στο δίσκο Raspberry Pi. Αυτή η προσέγγιση, η οποία καταγράφει κυρίως εικόνες και όχι βίντεο, ενισχύει τη λειτουργική αποτελεσματικότητα και διευκολύνει τις επακόλουθες διαδικασίες ανάλυσης και αρχειοθέτησης.

- Απομακρυσμένη εποπτεία μέσω RVNC

Η ενσωμάτωση του RVNC (Remote Virtual Network Computing) επιτρέπει την αποτελεσματική επίβλεψη και διαχείριση των λειτουργιών του συστήματος, επιτρέποντας λειτουργίες απομακρυσμένης πρόσβασης και ελέγχου που συμβάλλουν στην βελτιωμένη επίβλεψη και συντήρηση του συστήματος.

6.2 Προτάσεις για μελλοντικές αλλαγές

Με βάση τα ευρήματα της μελέτης, οι μελλοντικές βελτιώσεις για το σύστημα επιτήρησης περιλαμβάνουν τη βελτιστοποίηση αλγορίθμων συμπίεσης για τη μείωση των μεγεθών αρχείων, την ενσωμάτωση συστημάτων ειδοποιήσεων σε πραγματικό χρόνο για έγκαιρες ειδοποιήσεις, την ελαχιστοποίηση του λανθάνοντος χρόνου ζωντανής ροής, την εξερεύνηση επεκτάσιμων επιλογών ανάπτυξης, την ενσωμάτωση ανάλυσης βίντεο και AI για έξυπνη παρακολούθηση.

Οι εν λόγω αλλαγές στοχεύουν στην ενίσχυση της αποτελεσματικότητας του συστήματος, της ανταπόκρισης, της επεκτασιμότητας και της εμπειρίας χρήστη, διασφαλίζοντας αποτελεσματική παρακολούθηση ασφάλειας και ανίχνευση απειλών σε διαφορετικά περιβάλλοντα.

BIBΛΙΟΓΡΑΦΙΚΕΣ ΑΝΑΦΟΡΕΣ

1. Obaidat, M.; Obeidat, S.; Holst, J.; Al Hayajneh, A.; Brown, J. A Comprehensive and Systematic Survey on the Internet of Things: Security and Privacy Challenges, Security Frameworks, Enabling Technologies, Threats, Vulnerabilities and Countermeasures. *Computers* **2020**, *9*, 44, doi:10.3390/computers9020044.
2. Mowla, Md.N.; Mowla, N.; Shah, A.F.M.S.; Rabie, K.M.; Shongwe, T. Internet of Things and Wireless Sensor Networks for Smart Agriculture Applications: A Survey. *IEEE Access* **2023**, *11*, 145813–145852, doi:10.1109/ACCESS.2023.3346299.
3. Khan, M.A.; Ahmad, I.; Nordin, A.N.; Ahmed, A.E.-S.; Mewada, H.; Daradkeh, Y.I.; Rasheed, S.; Eldin, E.T.; Shafiq, M. Smart Android Based Home Automation System Using Internet of Things (IoT). *Sustainability* **2022**, *14*, 10717, doi:10.3390/su141710717.
4. Shah, S.K.; Tariq, Z.; Lee, Y. Audio IoT Analytics for Home Automation Safety. In Proceedings of the 2018 IEEE International Conference on Big Data (Big Data); December 2018; pp. 5181–5186.
5. He, T.; Krishnamurthy, S.; Stankovic, J.; Abdelzaher, T.; Luo, L.; Stoleru, R.; Yan, T.; Gu, L.; Hui, J.; Krogh, B. *Energy-Efficient Surveillance System Using Wireless Sensor Networks*; 2004;
6. Mellado, D.; Rosado, D.G. An Overview of Current Information Systems Security Challenges and Innovations J.UCS Special Issue.
7. 10 Most Important Qualities of a Good Security System 2018.
8. Core Characteristics and Advantages of Smart Home Security Systems | Eastern Peak Available online: <https://easternpeak.com/blog/building-smart-home-security-system/> (accessed on 5 February 2024).
9. Elharrouss, O.; Almaadeed, N.; Al-Maadeed, S. A Review of Video Surveillance Systems. *J. Vis. Commun. Image Represent.* **2021**, *77*, 103116, doi:10.1016/j.jvcir.2021.103116.
10. Porikli, F.; Brémond, F.; Dockstader, S.; Ferryman, J.; Hoogs, A.; Lovell, B.; Pankanti, S.; Rinner, B.; Tu, P.; Venetianer, P. Video Surveillance: Past, Present, and Now the Future. *IEEE Signal Process. Mag.* **2013**, *30*, 190–198, doi:10.1109/MSP.2013.2241312.
11. Rajeswari; Kumar, N.V.; Suresh, K.M.; Kumar, N.S.; Sravani, K.G. IoT-Based Smart Home Security Alert System for Continuous Supervision. In *Machine Learning Techniques for VLSI Chip Design*; 2024; pp. 51–63 ISBN 978-1-119-91049-7.

12. Yunus, N.H.M.; Zamri, M.I.M.; Yusof, N. Development of a Motion Activated Security Cam for Monitoring Applications. *SpringerBriefs Appl. Sci. Technol.* **2024**, Part F2025, 1–8, doi:10.1007/978-3-031-47727-0_1.
13. Parmar, D.; Mehta, B. Face Recognition Methods & Applications. *Int. J. Comput. Technol. Appl.* **2014**, 4, 84–86.
14. Boesch, G. Image Recognition: The Basics and Use Cases (2024 Guide) Available online: <https://viso.ai/computer-vision/image-recognition/> (accessed on 5 February 2024).
15. Face Recognition: 7 Techniques and a Quick OpenCV Tutorial Available online: <https://datagen.tech/guides/face-recognition/7-techniques-and-a-quick-opencv-tutorial/> (accessed on 5 February 2024).
16. Ltd, R.P. Raspberry Pi Available online: <https://www.raspberrypi.com/> (accessed on 5 February 2024).
17. ARM Cortex-A72. *Wikipedia* 2023.
18. Camera Serial Interface. *Wikipedia* 2023.
19. FrontPage - Raspbian Available online: <https://www.raspbian.org/> (accessed on 5 February 2024).
20. AnkerWork PowerConf C200 Webcam Available online: <https://us.ankerwork.com/products/a3369> (accessed on 5 February 2024).
21. Ltd, R.P. Buy a Raspberry Pi 4 Model B Available online: <https://www.raspberrypi.com/products/raspberry-pi-4-model-b/> (accessed on 5 February 2024).
22. GPIO Pins | Physical Computing with Python | Python | Coding Projects for Kids and Teens Available online: <https://projects.raspberrypi.org/en/projects/physical-computing/1> (accessed on 5 February 2024).
23. PIR Motion Detector Sensor Module Available online: <https://rathyelectronics.com/product/pir-sensor-module/> (accessed on 5 February 2024).
24. Ltd, R.P. Raspberry Pi OS Available online: <https://www.raspberrypi.com/software/> (accessed on 5 February 2024).
25. Al-Rawi, M.; Abdulhamid, M.; Sheshai, S. Design of Security System Based on Raspberry-PI. *Sci. Bull. Electr. Eng. Fac.* **2019**, 19, 56–61, doi:10.1515/sbeef-2019-0022.

26. Iwan, T.; Fahrezy, H.; Merliasari, R. *The Design and the Implementation of Security System Office Door Using Raspberry Pi Face Detection*; 2020;
27. Simple Mail Transfer Protocol. *Wikipedia* 2024.
28. MQTT. *Wikipedia* 2023.
29. Cayenne Available online: <https://www.thethingsindustries.com/docs/integrations/cloud-integrations/cayenne/> (accessed on 5 February 2024).
30. Why Cayenne? · Apache Cayenne Available online: <https://cayenne.apache.org/why-cayenne.html> (accessed on 5 February 2024).
31. Blynk: A Low-Code IoT Software Platform for Businesses and Developers Available online: <https://blynk.io> (accessed on 5 February 2024).
32. Patil, N.; Ambatkar, S.; Kakde, S. IoT Based Smart Surveillance Security System Using Raspberry Pi. *2017 Int. Conf. Commun. Signal Process. ICCSP 2017*, 0344–0348, doi:10.1109/ICCSP.2017.8286374.
33. Python (Programming Language). *Wikipedia* 2024.
34. Home Available online: <https://opencv.org/> (accessed on 5 February 2024).
35. Soni, P.; Verma, S.; Pandey, S. IoT Based Reconnaissance Framework for Utilizing Different Thresholding Algorithms for Detecting Motion by Using Raspberry Pi.; Kumar, A., Paprzycki, M., Gunjan, V.K., Eds.; Springer Singapore: Singapore, 2020; Vol. 601, pp. 231–238.
36. Aspect Ratio. *Wikipedia* 2024.
37. OpenCV: Contour Properties Available online: https://docs.opencv.org/4.x/da/dc1/tutorial_js_contour_properties.html (accessed on 5 February 2024).
38. Os — Miscellaneous Operating System Interfaces Available online: <https://docs.python.org/3/library/os.html> (accessed on 5 February 2024).
39. Wearn, O.; Glover-Kapfer, P. *Camera-Trapping for Conservation: A Guide to Best-Practices*; 2017;
40. Instructions for Encrypted Data Collection via Email Available online: <https://it.tufts.edu/book/export/html/1568> (accessed on 5 February 2024).

41. Send Data From Hardware To Blynk Available online: <https://docs.blynk.io/en/getting-started/how-to-display-any-sensor-data-in-blynk-app> (accessed on 5 February 2024).
42. Implementing Gaussian Smoothing In Data Analysis Available online: <https://fastercapital.com/keyword/implementing-gaussian-smoothing-in-data-analysis.html> (accessed on 5 February 2024).
43. Gaussian Process. *Wikipedia* 2024.
44. Motion Detector. *Wikipedia* 2024.
45. AG, T. Presence and Motion Detectors Available online: <https://www.theben.de/en/presence-and-motion-detectors-442-c/> (accessed on 5 February 2024).
46. Raspberry Pi Documentation - Raspberry Pi OS Available online: <https://www.raspberrypi.com/documentation/computers/os.html> (accessed on 7 February 2024).
47. Raspberry Pi Compute Module 4.
48. How HC-SR501 PIR Sensor Works & How To Interface It With Arduino Available online: <https://lastminuteengineers.com/pir-sensor-arduino-tutorial/> (accessed on 7 February 2024).
49. PIR Sensor - Human Body Detecting Module Pyroelectric HC-SR501 For Arduino Available online: <https://www.robotpark.com/PIR-Sensor-Human-Body-Detecting-Pyroelectric-HC-SR501> (accessed on 7 February 2024).

Παράρτημα Α

Προετοιμασία υλικού:

Πριν από την έναρξη των εργασιών προγραμματισμού για το Raspberry Pi, το προκαταρκτικό βήμα περιλαμβάνει την εγκατάσταση του λειτουργικού του συστήματος. Αυτή η διαδικασία διευκολύνεται μέσω της χρήσης ενός εργαλείου λογισμικού που είναι γνωστό ως Raspberry Pi Imager. Συγκεκριμένα, υπάρχουν διάφορες εκδόσεις αυτού του εργαλείου. Στο πλαίσιο αυτής της υλοποίησης, η εγκατάσταση εκτελέστηκε με χρήση Raspberry Pi Imager έκδοση 1.6.1.

Για να ξεκινήσει η διαδικασία εγκατάστασης, μπορείτε να προμηθευτείτε το συγκεκριμένο πρόγραμμα από τον επίσημο ιστότοπο Raspberry, ο οποίος είναι προσβάσιμος μέσω του παρακάτω συνδέσμου: <https://www.raspberrypi.com/software/>. Στη συνέχεια, οι χρήστες μπορούν να προμηθευτούν το απαραίτητο αρχείο που αντιστοιχεί στο λειτουργικό σύστημα της επιλογής τους (π.χ. Windows, macOS, Ubuntu).

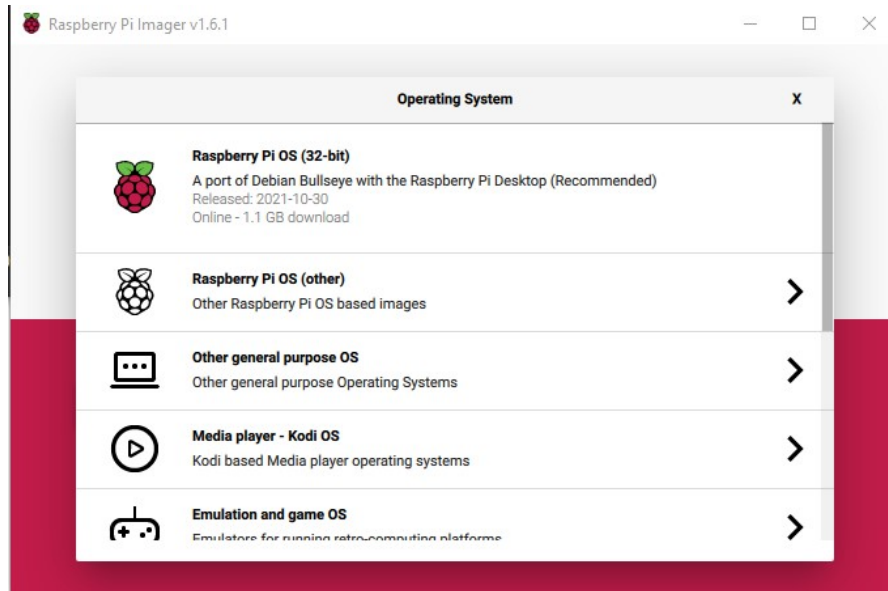
Κατά τη λήψη και εγκατάσταση του λογισμικού, η εκτέλεσή του αποκαλύπτει μια γραφική διεπαφή χρήστη (GUI) που παρουσιάζει διάφορες επιλογές και λειτουργίες. Αυτή η διεπαφή, που απεικονίζεται στην Εικόνα 20 χρησιμεύει ως η λειτουργική πύλη για την έναρξη της διαδικασίας εγκατάστασης του λειτουργικού συστήματος του Raspberry Pi.



Εικόνα 20 Προετοιμασία κάρτας

Η διεπαφή παρουσιάζει στους χρήστες δύο ευδιάκριτες επιλογές. Η αρχική επιλογή οριοθετεί το λειτουργικό σύστημα που θα εγκατασταθεί στην κάρτα μνήμης. Αυτό περιλαμβάνει επιλογές όπως το λειτουργικό σύστημα Ubuntu ή ξεχωριστές εκδόσεις του λειτουργικού συστήματος Raspberry Pi, συμπεριλαμβανομένων των επαναλήψεων

παλαιού τύπου. Όπως φαίνεται στην Εικόνα 21 παρακάτω, η διεπαφή του μενού υλοποιείται με την επιλογή του κουμπιού "CHOOSE OS" στην επιλογή Operating System.



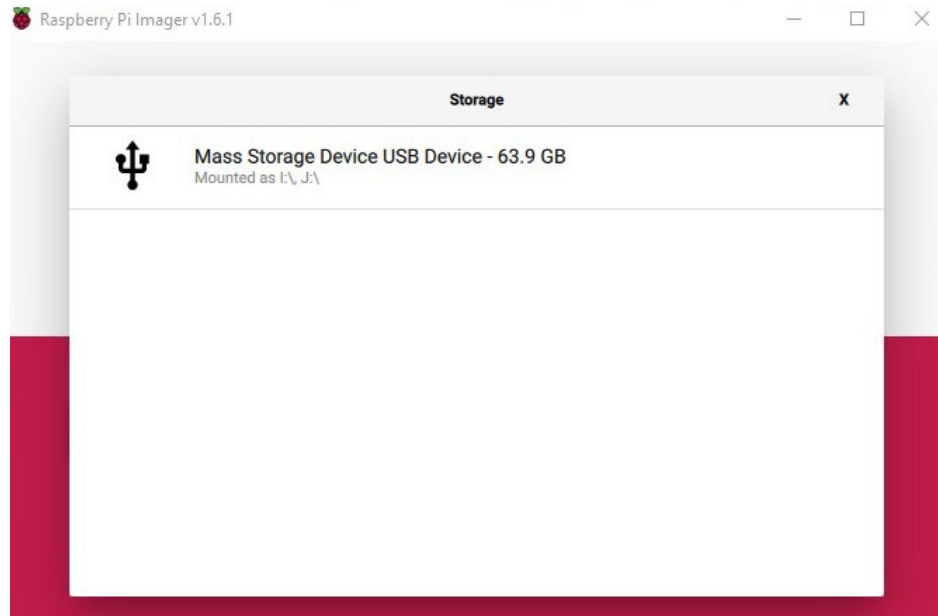
Εικόνα 21 Επιλογή λογισμικού

Στην προαναφερθείσα διεπαφή μενού παρέχονται στους χρήστες διάφορες επιλογές, όπως η εγκατάσταση των προτιμώμενων λειτουργικών συστημάτων, η διαγραφή δεδομένων στην κάρτα μνήμης ή η εγκατάσταση ληφθέντων αρχείων λειτουργικού συστήματος, όπως οι παλαιότερες εκδόσεις του Raspberry Pi OS. Για την αρχή, συνιστάται η τήρηση της προτεινόμενης επιλογής. Εγγυάται η αναφορά ότι η διαδικασία εγκατάστασης μπορεί να έχει σημαντική διάρκεια λόγω της απαιτούμενης λήψης του επιλεγμένου αρχείου, το οποίο υποδεικνύεται μαζί με το μέγεθος του αρχείου του, που συνήθως υποδηλώνεται σε megabyte (MB) ή gigabyte (GB). Στην τρέχουσα περίπτωση, το επιλεγμένο αρχείο έχει μέγεθος 1,1 GB και κυκλοφόρησε στις 30/10/2021.

Αξίζει να σημειωθεί ότι ο κωδικός που χρησιμοποιείται σε αυτή τη μελέτη λειτουργεί με την έκδοση με ημερομηνία 30/09/2019. Αξίζει την προσοχή ότι μπορεί να προκύψουν πιθανές αποκλίσεις κατά την εκτέλεση του κώδικα σε νεότερες εκδόσεις, που αποδίδονται σε απαραίτητες ενημερώσεις σε ορισμένες βιβλιοθήκες.

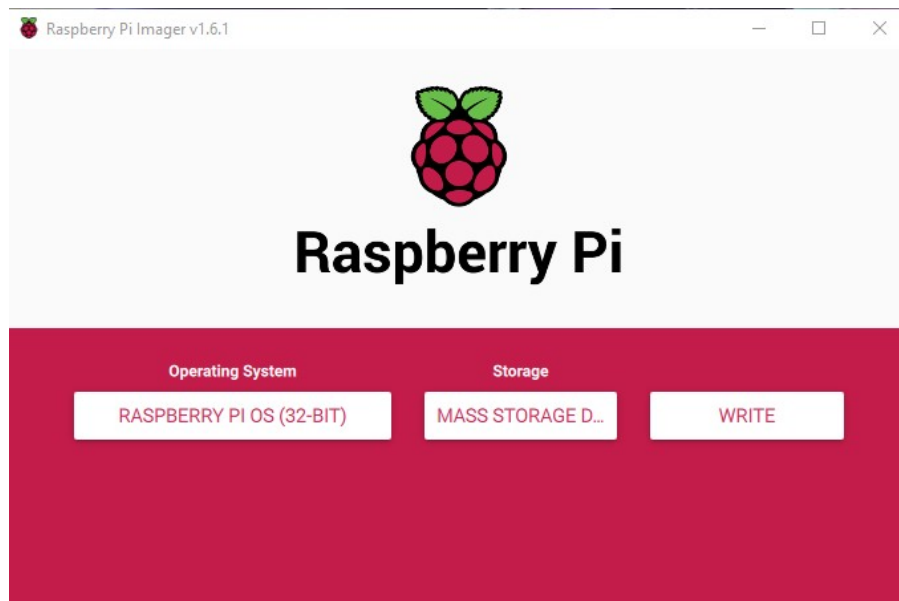
Προχωρώντας στη διαδικασία εγκατάστασης, μετά την επιλογή του επιθυμητού λειτουργικού συστήματος, οι χρήστες προχωρούν στο επόμενο βήμα κάνοντας κλικ στην επιλογή "CHOOSE STORAGE". Αυτή η ενέργεια προτρέπει την εμφάνιση της διαθέσιμης κάρτας μνήμης και της σχετικής χωρητικότητας αποθήκευσης, που υποδηλώνεται σε GB. Η επιλογή του μεγέθους της κάρτας μνήμης εξαρτάται την επιλογή του χρήστη. Αξίζει να σημειωθεί ότι συγκεκριμένες εκδόσεις λογισμικού, μπορεί να έχουν ελάχιστο όριο μνήμης

ως απαιτούμενο. Δεδομένης της χρήσης ενός Raspberry Pi 4B, χρησιμοποιείται μια κάρτα μνήμης 64 GB, όπως φαίνεται στην παρακάτω εικόνα.



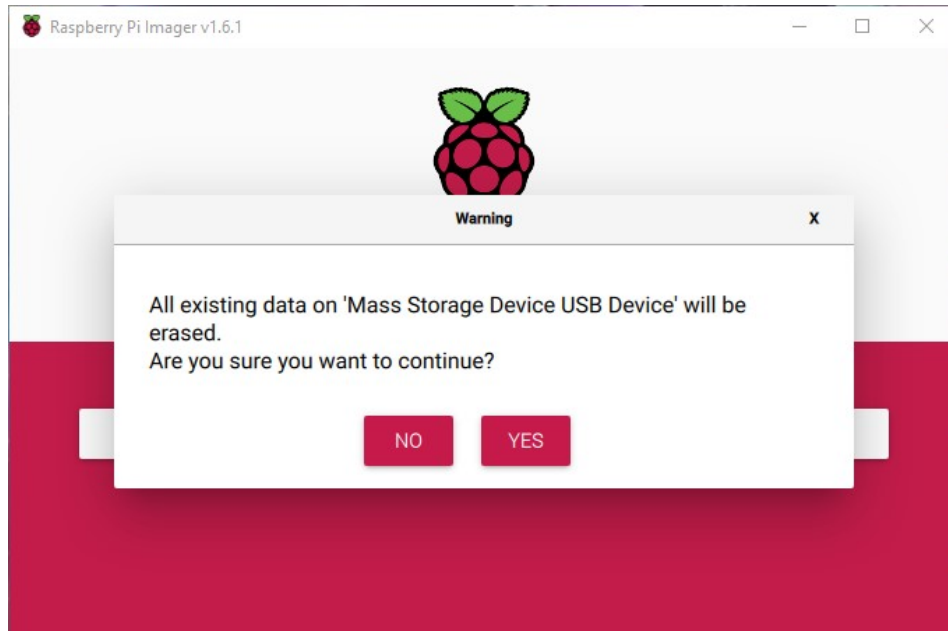
Εικόνα 22 Επιλογή κάρτας

Αφού ολοκληρωθεί και η επιλογή της κάρτας μνήμης, ενεργοποιείται η επιλογή εγγραφής.



Εικόνα 23 Εκκίνηση διαμόρφωσης

Επιλέγοντας το WRITE, το σύστημα προειδοποιεί το χρήστη αν είναι σίγουρος για την επιλογή του. Καθώς, αφού επιλεγεί η λειτουργία αυτή γίνεται διαμόρφωση της κάρτας μνήμης στο σωστό τύπο και χάνονται όλα τα αρχεία που εμπεριέχονται σε αυτή.



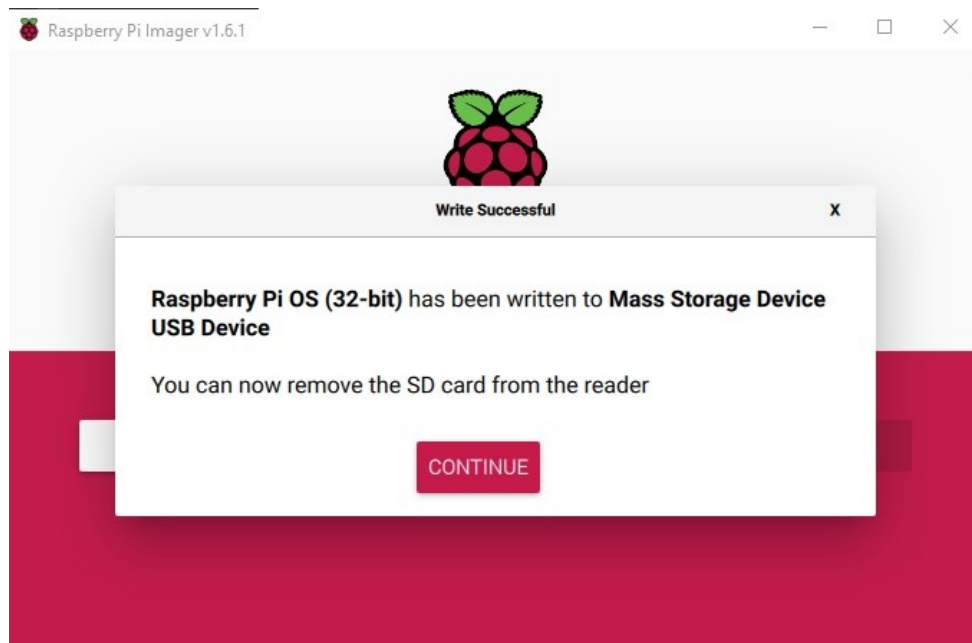
Εικόνα 24 Προειδοποιητικό μήνυμα διαμόρφωσης

Αφού εκκινηθεί η διαδικασία της διαμόρφωσης και εγγραφής του λογισμικού, ο χρήστης ενημερώνεται για την πρόοδο της μέσω μια μπάρας που εμφανίζεται στο κέντρο του παραθύρου.

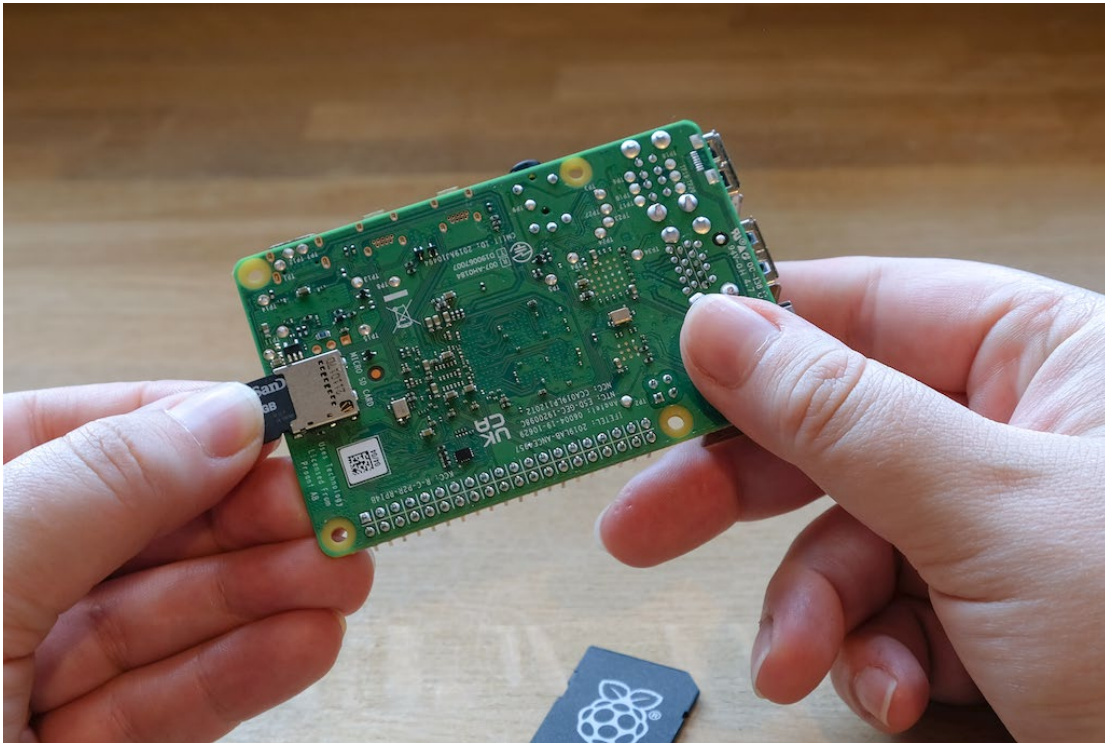


Εικόνα 25 Πρόοδος διαμόρφωσης

Τέλος, αφού ολοκληρωθεί η εγκατάσταση του λογισμικού, εμφανίζεται ένα μήνυμα επιτυχίας. Ο χρήστης ενημερώνεται επίσης ότι είναι δυνατή, η αφαίρεση της κάρτας μνήμης από τον υπολογιστή και μπορεί να προχωρήσει στην τοποθέτηση της στο Raspberry Pi.



Εικόνα 26 Μήνυμα επιτυχημένης διαμόρφωσης



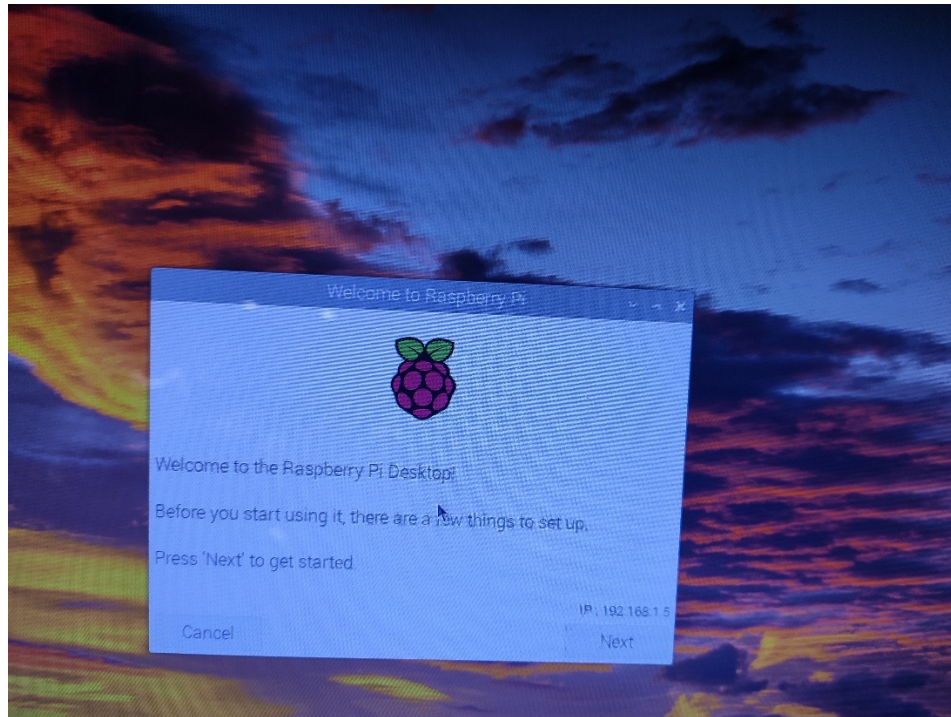
Εικόνα 27 Τοποθέτηση κάρτας στο Raspberry Pi

Εγκατάσταση λογισμικού:

Μετά την προετοιμασία του υλικού, οι χρήστες προχωρούν στη διαδικασία εγκατάστασης λογισμικού, η οποία περιλαμβάνει τη διαμόρφωση του λειτουργικού συστήματος του Raspberry Pi και την εγκατάσταση σχετικών πακέτων λογισμικού. Αυτό το βήμα συνήθως περιλαμβάνει τη λήψη της επιθυμητής εικόνας του λειτουργικού συστήματος, την αναλαμπή της σε μια κάρτα microSD και την εκκίνηση του Raspberry Pi. Στη συνέχεια, οι χρήστες μπορούν να εγκαταστήσουν πρόσθετα πακέτα λογισμικού, βιβλιοθήκες και εφαρμογές προσαρμοσμένες στις συγκεκριμένες απαιτήσεις του έργου τους.

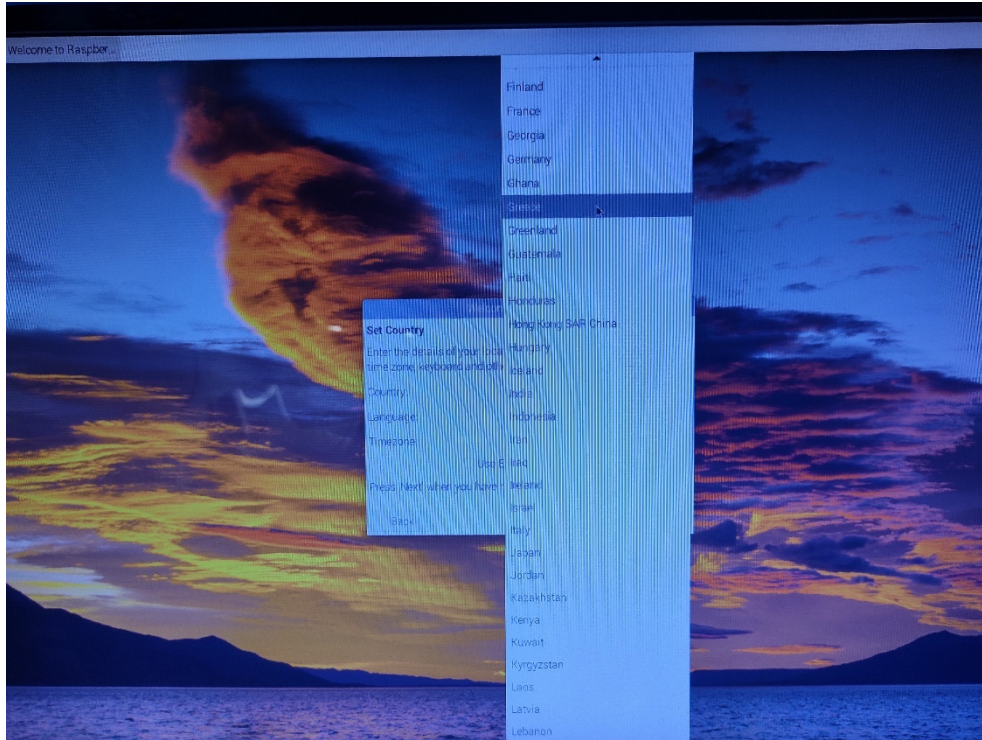
Με την ολοκλήρωση της προαναφερθείσας διαδικασίας και την επακόλουθη σύνδεση του Raspberry Pi στο τροφοδοτικό, την οθόνη, το πληκτρολόγιο και το ποντίκι η συσκευή ενεργοποιείται. Ωστόσο, απομένουν αρκετά πρόσθετα βήματα για την προετοιμασία του Pi κατά την αρχική του εκκίνηση.

Κατά την πρώτη εκκίνηση του Raspberry Pi, οι χρήστες ενδέχεται να συναντήσουν προτροπές για την εκτέλεση ενημερώσεων και τη διαμόρφωση ρυθμίσεων όπως προτιμήσεις γλώσσας, επιλογή χώρας, συνδεσιμότητα ασύρματου δικτύου και ενημερώσεις συστήματος. Συγκεκριμένα, αυτές οι διαμορφώσεις είναι δέουσας σημασίας για τη βελτιστοποίηση της λειτουργικότητας και της χρηστικότητας του Pi.



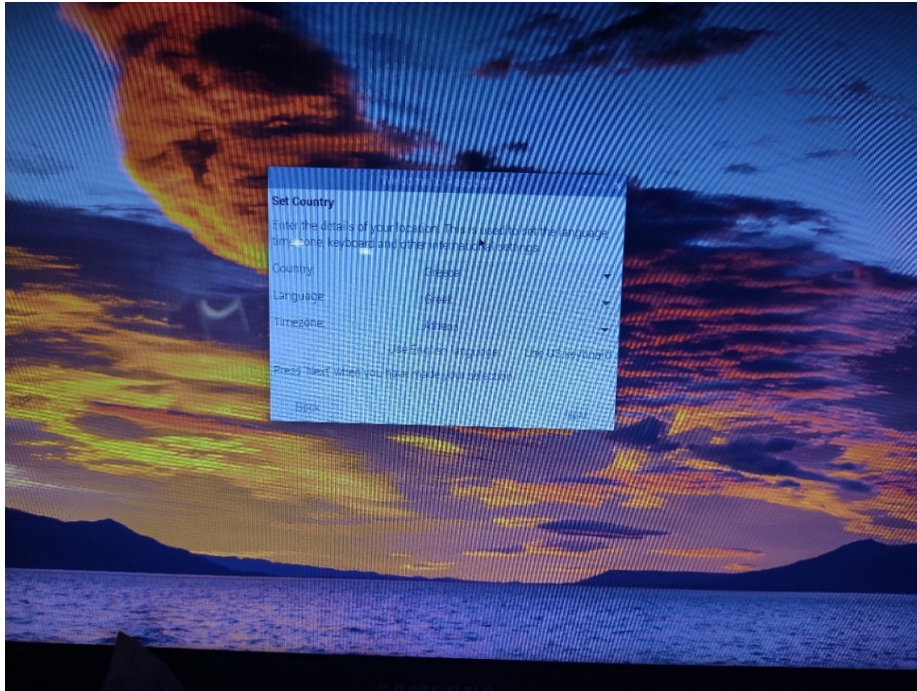
Εικόνα 28 Ρυθμίσεις εγκατάστασης- Πρώτη εκκίνηση Raspberry Pi OS

Όπως φαίνεται στην Εικόνα 28, ο οδηγός αρχικής εκκίνησης προτρέπει τους χρήστες για τις απαραίτητες διαμορφώσεις κατά την πρώτη εκκίνηση της συσκευής. Η εμφανιζόμενη διεύθυνση IP υποδηλώνει σύνδεση ενσύρματου δικτύου. Για να προχωρήσουν στον οδηγό και να διαμορφώσουν τις επόμενες ρυθμίσεις στο Raspberry Pi, οι χρήστες επιλέγουν την επιλογή "Next".



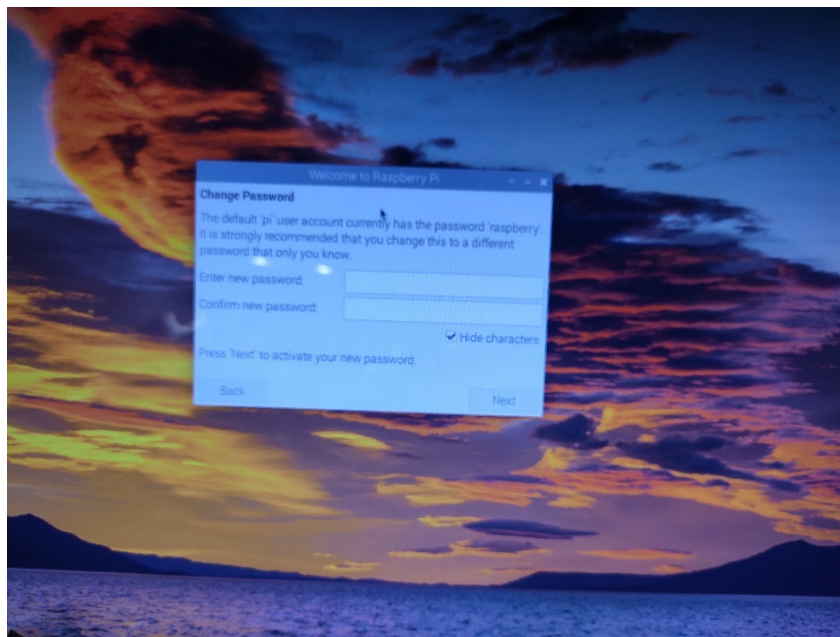
Εικόνα 29 Ρυθμίσεις εγκατάστασης- Επιλογή Χώρας

Η επιλογή "Next" δίνει στους χρήστες την ευκαιρία να καθορίσουν τις προτιμήσεις γλώσσας, τις ρυθμίσεις χώρας (π.χ. Ελλάδα) και τις προτιμήσεις ζώνης ώρας, όπως απεικονίζονται στην Εικόνα 29. Στη συνέχεια, οι χρήστες συνεχίζουν επιλέγοντας το κουμπί "Next", όπως φαίνεται στην Εικόνα 30.



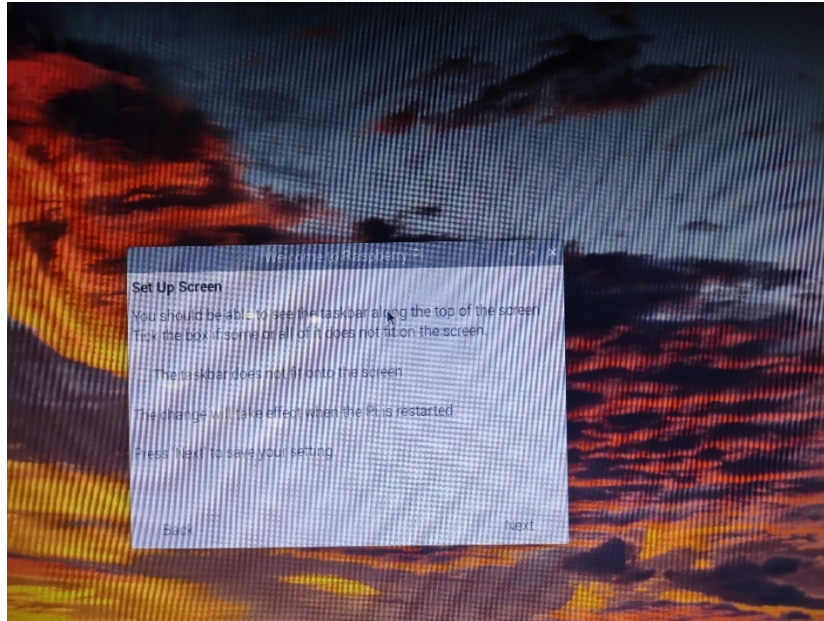
Εικόνα 30 Ρυθμίσεις εγκατάστασης- Καθορισμός Χώρας

Το επόμενο βήμα συνεπάγεται τον ορισμό ενός κωδικού πρόσβασης για τον λογαριασμό χρήστη για την προστασία της πρόσβασης στη συσκευή. Προαιρετικά, οι χρήστες μπορούν να καταργήσουν την επιλογή " Hide Characters" για να εξασφαλίσουν την ακριβή εισαγωγή του κωδικού πρόσβασης, όπως φαίνεται στην Εικόνα 31.



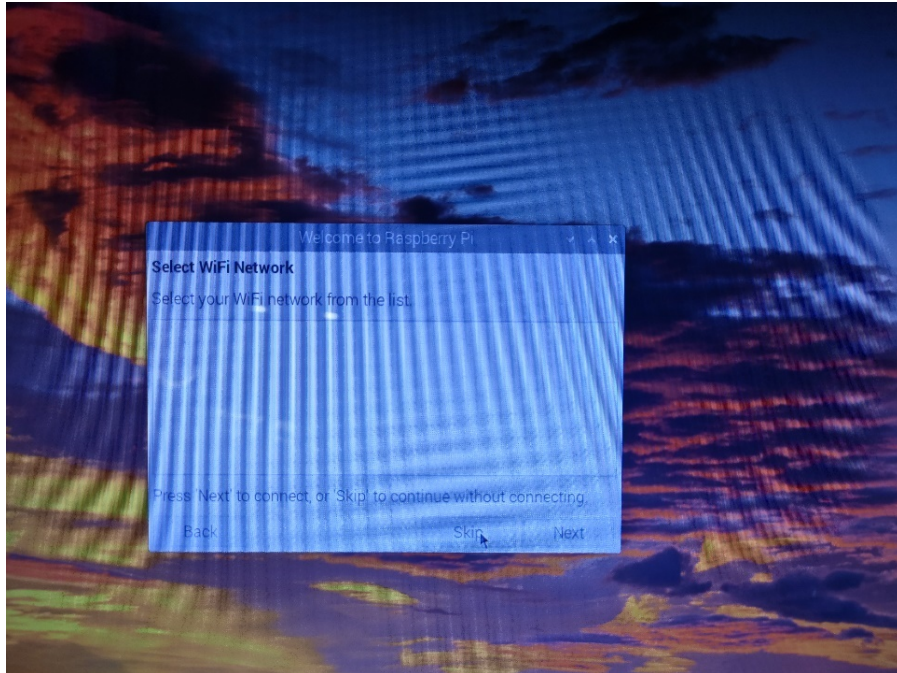
Εικόνα 31 Ρυθμίσεις εγκατάστασης- Καθορισμός Κωδικού

Μετά την προσαρμογή της γραμμής εργασιών ώστε να ταιριάζει στην οθόνη, οι χρήστες έχουν την επιλογή να προχωρήσουν επιλέγοντας το κουμπί "Next", όπως φαίνεται στην Εικόνα 32. Σε περίπτωση που η γραμμή εργασιών δεν ταιριάζει σωστά, οι χρήστες μπορούν να επιλέξουν το διάλογο πλαίσιο που παρέχεται παραπάνω.



Εικόνα 32 Ρυθμίσεις εγκατάστασης- ρυθμίσεις οθόνης

Στη συνέχεια, ζητείται από τους χρήστες να επιλέξουν την επιθυμητή μέθοδο για τη σύνδεση του Raspberry Pi σε ένα ασύρματο δίκτυο. Ωστόσο, δεδομένης της χρήσης μιας ενσύρματης σύνδεσης, οι χρήστες επιλέγουν να παρακάμψουν αυτό το βήμα επιλέγοντας το κουμπί "Skip", όπως φαίνεται στην Εικόνα 33.

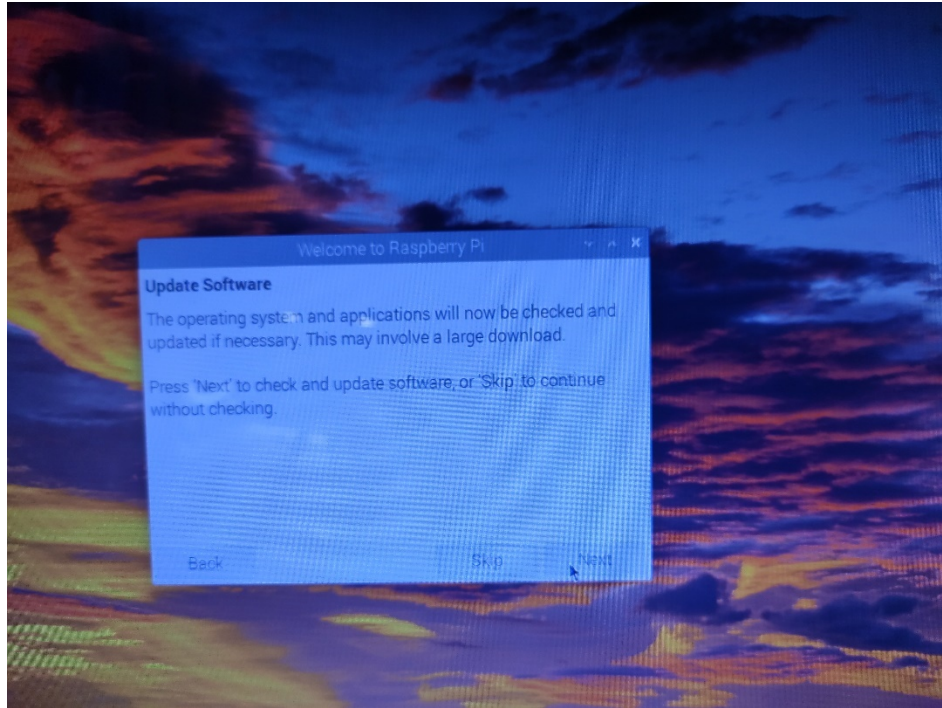


Εικόνα 33 Ρυθμίσεις εγκατάστασης- ρυθμίσεις WiFi

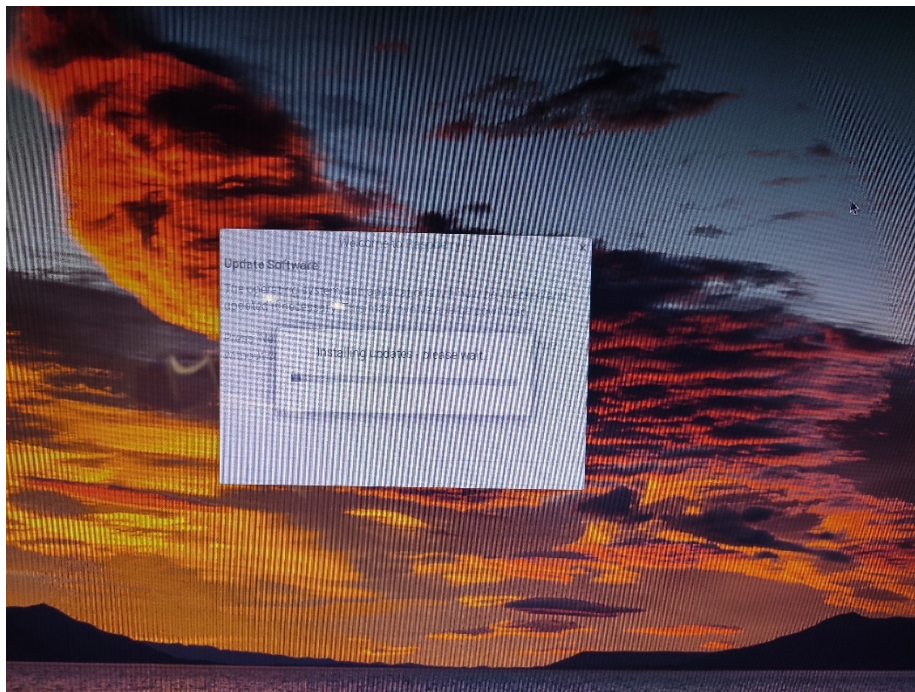
Στο επόμενο βήμα, δίνεται στους χρήστες η επιλογή να ενημερώσουν το λογισμικό στο Raspberry Pi. Παρουσιάζονται δύο επιλογές: είτε να ξεκινήσετε την ενημέρωση λογισμικού αμέσως είτε να την αναβάλετε σε μεταγενέστερο χρόνο και να εκτελέσετε την ενημέρωση χειροκίνητα μέσω εντολών τερματικού. Αυτή η επιλογή επιτρέπει στους χρήστες ευελιξία στη διαχείριση ενημερώσεων λογισμικού, όπως φαίνεται στην Εικόνα 34.

sudo apt update

sudo apt full-upgrade ή sudo apt upgrade [46]



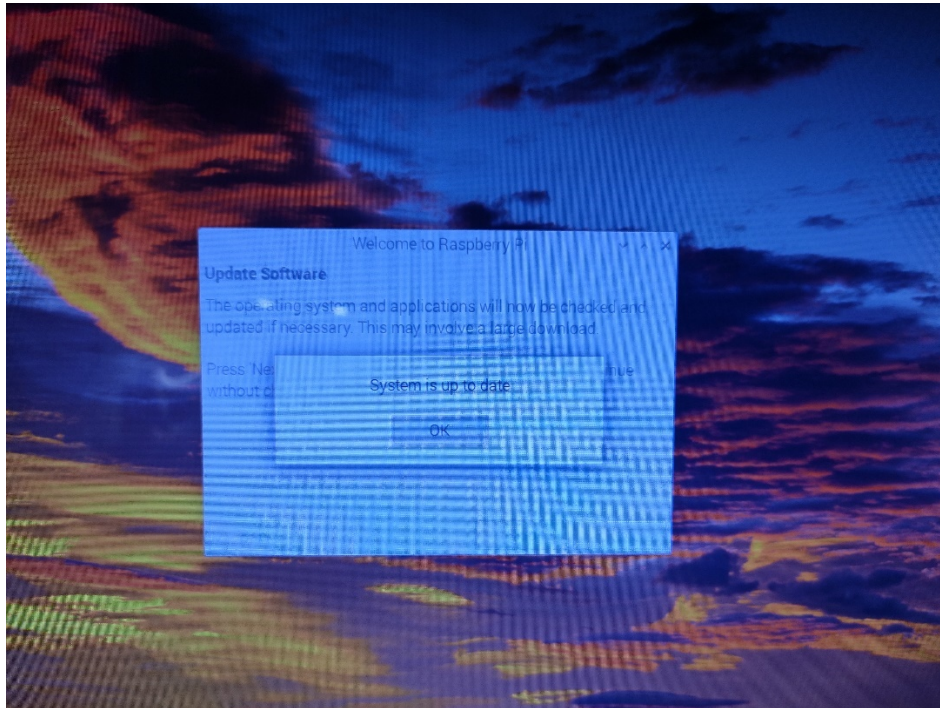
Εικόνα 34 Ρυθμίσεις εγκατάστασης- Ενημέρωση λογισμικού



Εικόνα 35 Ρυθμίσεις εγκατάστασης- Ενημέρωση λογισμικού μπάρα προόδου

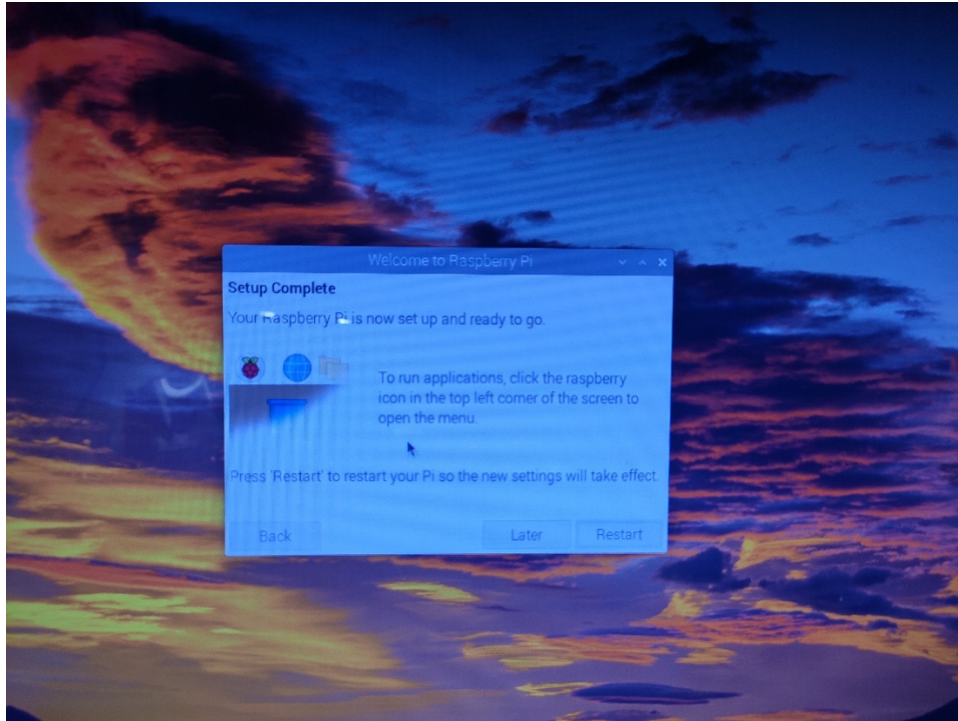
Εφ'όσον επιλέξουμε την εγκατάσταση με το κουμπί Next θα μας εμφανίσει την μπάρα προόδου εγκατάστασης.

Με το πέρας της εγκατάστασης των ενημερώσεων θα μας εμφανίσει ένα μήνυμα ότι ολοκληρώθηκε η εγκατάσταση.



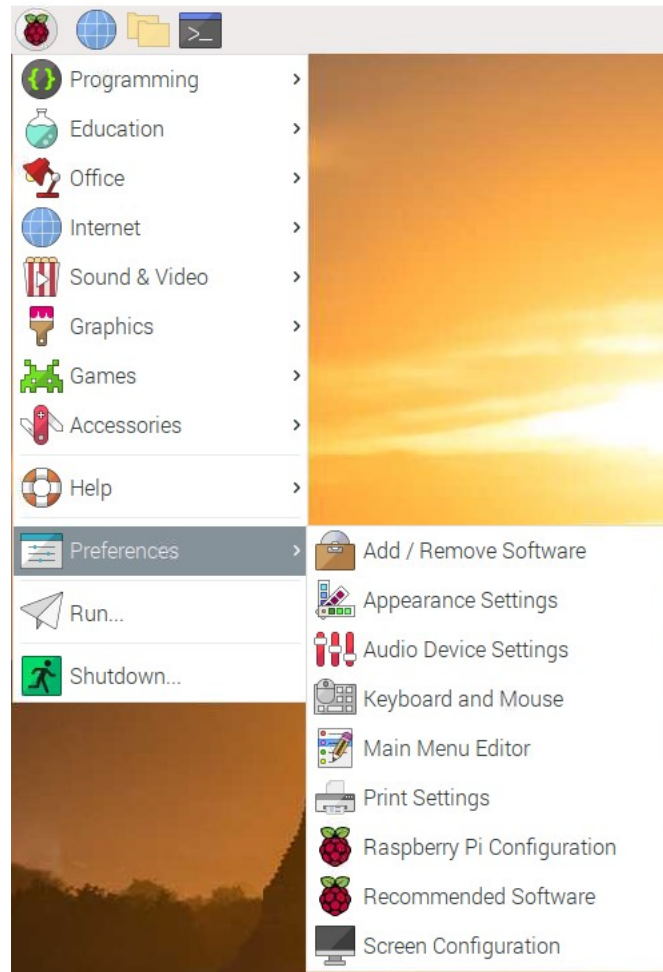
Εικόνα 36 Ρυθμίσεις εγκατάστασης- Ενημέρωση λογισμικού μήνυμα προόδου

Με το πάτημα του κουμπιού "OK", ζητείται από τους χρήστες να αποφασίσουν εάν θα επανεκκινήσουν το σύστημα μετά τις ενημερώσεις, μια συνιστώμενη πρακτική για τη διασφάλιση της εφαρμογής των αλλαγών, όπως φαίνεται στην Εικόνα εγκατάστασης 37.



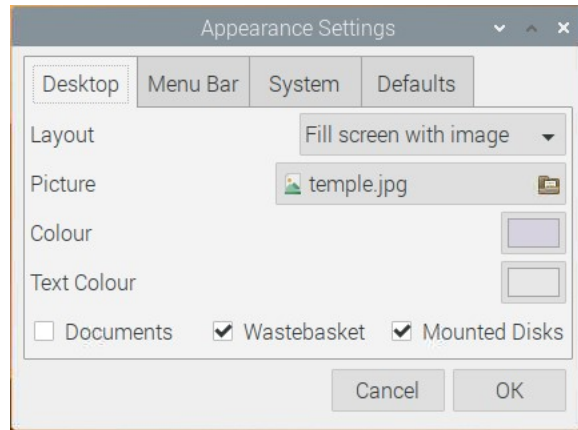
Εικόνα 37 Ρυθμίσεις εγκατάστασης- Μήνυμα ολοκλήρωσης

Η υπόλοιπη εργασία περιλαμβάνει τη διαμόρφωση των ρυθμίσεων του λειτουργικού συστήματος. Οι χρήστες πλοηγούνται στο μενού της εφαρμογής που βρίσκεται στην επάνω αριστερή γωνία και επιλέγουν "Preferences". Αυτή η ενέργεια αποκαλύπτει μια σειρά από ρυθμίσεις λειτουργικού συστήματος, διευκολύνοντας τις προσαρμογές στις επιλογές που έχουν διαμορφωθεί κατά την αρχική διαδικασία εγκατάστασης. Για βασικές ρυθμίσεις που σχετίζονται με την εμφάνιση, οι χρήστες επιλέγουν "Appearance settings".



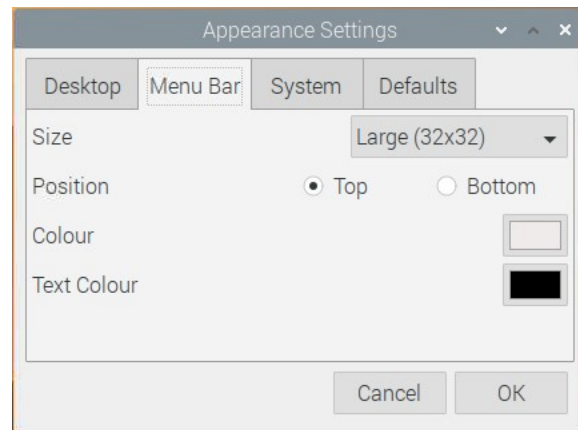
Εικόνα 38 Preferences- Appearance settings

Στο παράθυρο που ακολουθεί, παρουσιάζονται τέσσερις καρτέλες. Στην πρώτη καρτέλα με την ένδειξη "Desktop" οι χρήστες μπορούν να προσαρμόσουν την εικόνα φόντου επιλέγοντας από τις διαθέσιμες επιλογές ή προσθέτοντας μια προτιμώμενη εικόνα. Επιπλέον, η επιλογή αφαίρεσης της υπάρχουσας εικόνας και αλλαγής του χρώματος φόντου είναι διαθέσιμη στην ετικέτα "Color". Πιο κάτω στην καρτέλα, οι χρήστες συναντούν πλαίσια ελέγχου για "Documents", "Wastebasket" και "Mounted Disks", επιτρέποντας την επιλογή συντομεύσεων επιφάνειας εργασίας για αυτές τις δυνατότητες με βάση τις προτιμήσεις του χρήστη.



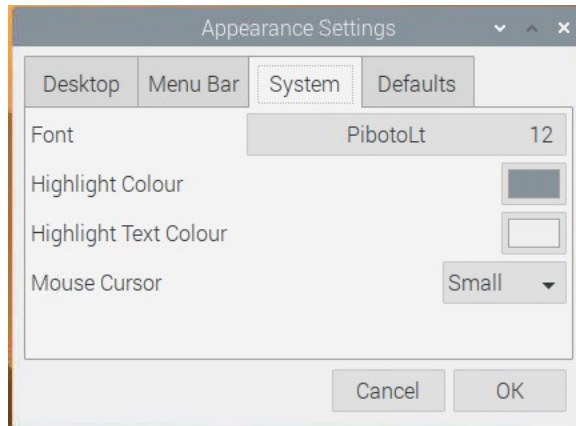
Εικόνα 39 Appearance settings- Desktop

Το Raspberry Pi OS παρουσιάζει συμβατότητα με διάφορα συστήματα αρχείων, όπως το FAT, το NTFS (που χρησιμοποιείται κυρίως στα Windows), το HFS+ (χρησιμοποιείται στο macOS) και το ext4 (που χρησιμοποιείται συνήθως από το Linux). Επιπλέον, υπάρχει η επιλογή εγκατάστασης προγραμμάτων οδήγησης για την αναγνώριση πρόσθετων συστημάτων αρχείων όπως το exFAT, διευκολύνοντας τη μεταφορά αρχείων χρησιμοποιώντας εξωτερικές μονάδες αποθήκευσης, όπως USB sticks.



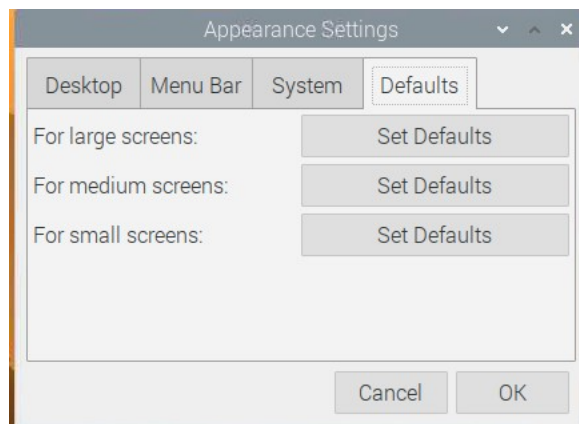
Εικόνα 40 Appearance settings - Menu bar

Η επόμενη καρτέλα (Menu Bar) αφορά την μπάρα του μενού (ή μπάρα εργασίας) μπορούμε να αλλάξουμε το χρώμα της, και παρέχει τη δυνατότητα (από την ετικέτα Position) να τοποθετηθεί κάτω μέρος της οθόνης.



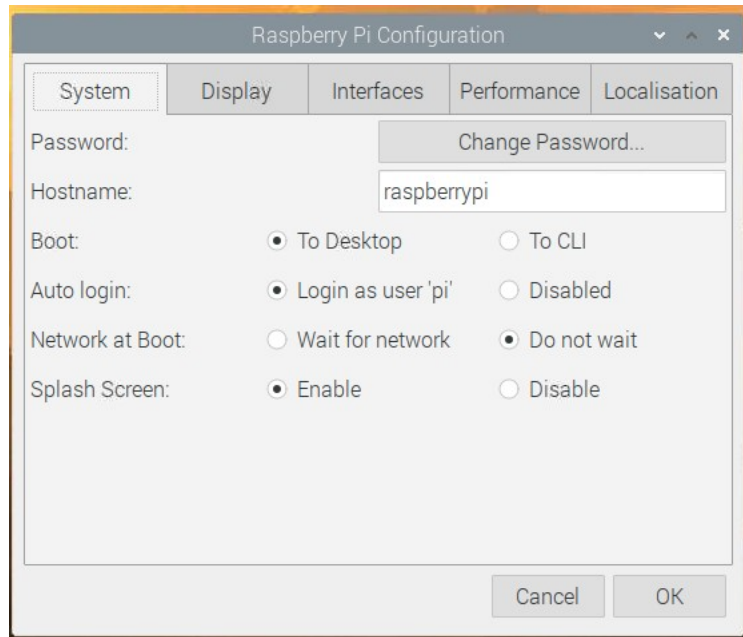
Εικόνα 41 Appearance settings - System

Στην καρτέλα System έχουμε τη δυνατότητα να κάνουμε μερικές ακόμη ρυθμίσεις εμφάνισης όπως το μέγεθος και το χρώμα της γραμματοσειράς (Font), αλλά και το μέγεθος του κέρσορα του ποντικιού μας.



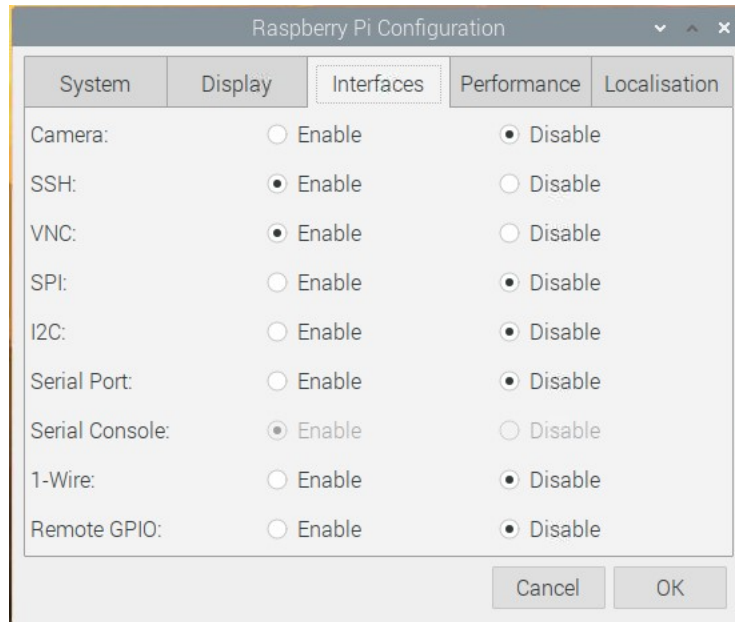
Εικόνα 42 Appearance settings - Defaults

Τώρα για το μέγεθος της εμφάνισης του περιβάλλοντος μας είναι πιο χρήσιμη η τελευταία καρτέλα (Defaults) η οποία μας παρέχει μια συνολική ρύθμιση στην εμφάνιση, ανάλογα με το μέγεθος της οθόνης που έχουμε συνδέσει στη συσκευή μας.



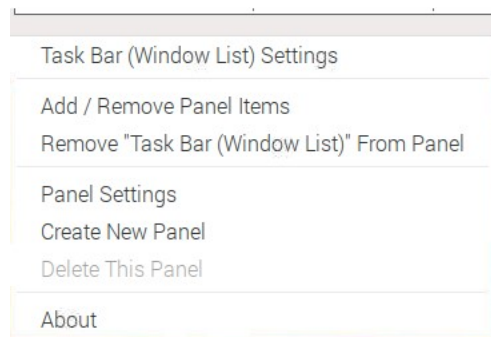
Εικόνα 43 Raspberry Pi Configuration- System

Στο υπομενού του Preferences επιλέγουμε το Raspberry Pi Configuration. Εδώ βρίσκουμε βασικές ρυθμίσεις για την λειτουργία της συσκευή μας. Στην πρώτη καρτέλα (System) μας δίνεται η δυνατότητα να αλλάξουμε το password αλλά και το όνομα της συσκευής μας. Καλό θα ήταν να αλλάχθει το όνομα εάν πρόκειται στο μέλλον να συνδέσουμε και δεύτερο Raspberry Pi. Τα ονόματα των συσκευών μας στο δίκτυο θα πρέπει να είναι μοναδικά ώστε να μπορούμε εύκολα να τα ξεχωρίζουμε. Παρακάτω στην επιλογή Boot μπορούμε να επιλέξουμε αν θα φορτώνει το γραφικό περιβάλλον αυτόματα κατά την εκκίνηση. Έχουμε δηλαδή την επιλογή όταν ανοίγουμε το Raspberry Pi να μας εμφανίζει το command line interface και όχι το γραφικό περιβάλλον. Αυτό μπορεί να είναι χρήσιμο. Αν για παράδειγμα έχουμε παλαιότερο Raspberry Pi με πολύ λίγη ram δεν θέλουμε να φορτώνουμε το γραφικό περιβάλλον παρά μόνο όταν μας είναι απαραίτητο. Με αυτόν τον τρόπο η συσκευή θα εκκινείται πιο γρήγορα. Παρακάτω στο Auto login, απενεργοποιώντας το Login as user 'pi', θα μας ζητάει κατά την εκκίνηση του λειτουργικού συστήματος το username και το password που είχαμε ορίσει πιο πριν.



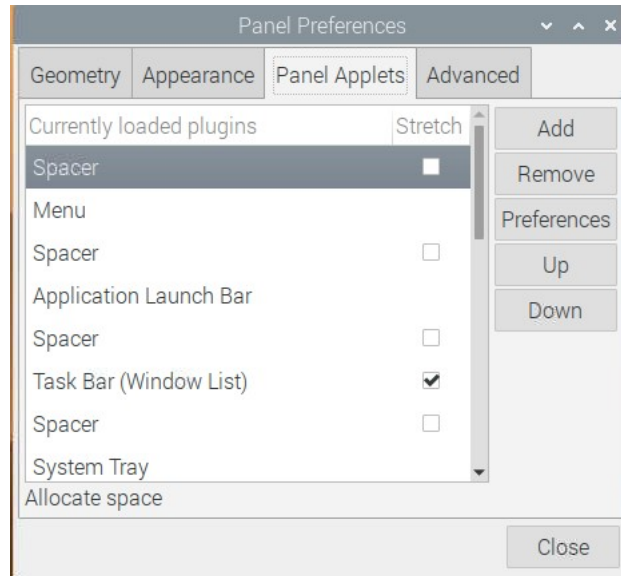
Εικόνα 44 Raspberry Pi Configuration- Interfaces

Άλλη μία σημαντική καρτέλα είναι το Interfaces. Εδώ μπορούμε να ενεργοποιούμε ή να απενεργοποιούμε κάποιες λειτουργίες της συσκευής μας. Η πρώτη λειτουργία με όνομα Camera αφορά την ειδική θύρα CSI που προορίζεται για σύνδεση συγκεκριμένου τύπου κάμερας, όπως η Raspberry Pi Camera. Δεν επηρεάζει την λειτουργία μιας απλής κάμερας που είναι συνδεδεμένη σε θύρα usb. Το SSH και το VNC θα πρέπει να είναι ενεργά για να μπορούμε να συνδεθούμε από διαφορετικό υπολογιστή. Όλες οι παρακάτω λειτουργίες αφορούν τα GPIOs της συσκευής μας. Αφού κάνουμε τις αλλαγές, πατάμε «OK» για να αποθηκευθούν.



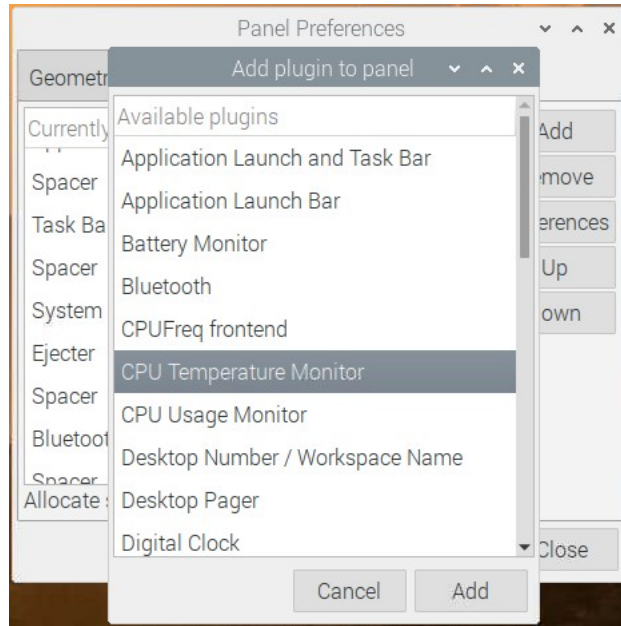
Εικόνα 45 Task Bar preferences - right click

Στη συνέχεια μπορούμε να προσθέσουμε αντικείμενα στο τέλος της μπάρας εργασιών μας. Για να το κάνουμε αυτό πατάμε δεξί κλικ πάνω στον κενό χώρο της μπάρας εργασίας και επιλέγουμε την δεύτερη επιλογή Add/Remove Panel Items.



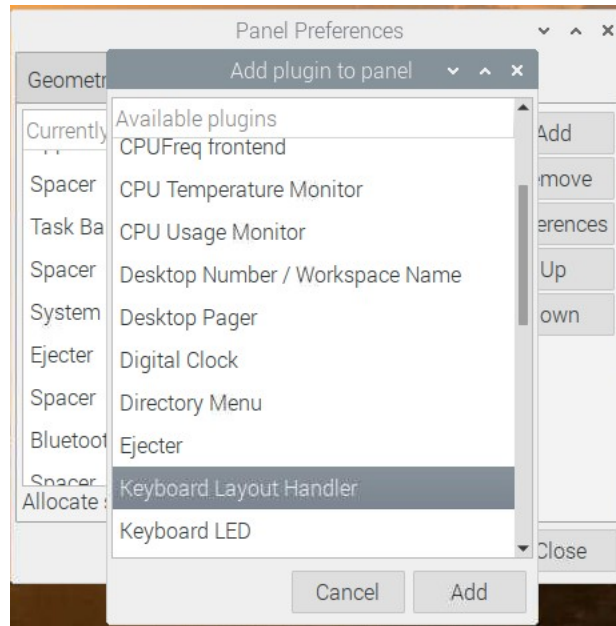
Εικόνα 46 Panel preferences – Panel applets

Μας εμφανίζεται μια λίστα με όσα αντικείμενα είναι τοποθετημένα στη μπάρα. Για παράδειγμα το Menu είναι το μενού των εφαρμογών, ενώ η Task Bar(Window List) αφορά το πλαίσιο με τις συντομεύσεις. Κάνοντας scroll down στη λίστα θα δούμε κάποια εικονίδια που βρίσκονται στο τέλος της μπάρας, όπως ο ήχος και το ρολόι. Μεταξύ των αντικειμένων θα παρατηρήσουμε ότι υπάρχει ένα περιθώριο. Αυτό μας βοηθά να είναι πιο ευανάγνωστα τα αντικείμενα. Επίσης θα διαπιστώσουμε ότι κάποια αντικείμενα δεν εμφανίζονται, για παράδειγμα το Battery που αφορά συγκεκριμένα πρόσθετα κιτ τα οποία περιλαμβάνουν μπαταρία και μετατρέπουν το Raspberry Pi σε φορητό υπολογιστή. Το εικονίδιο θα εμφανιστεί αυτόματα στην μπάρα εργασιών, μόνο αν έχουμε στη διάθεσή μας το πρόσθετο κιτ.



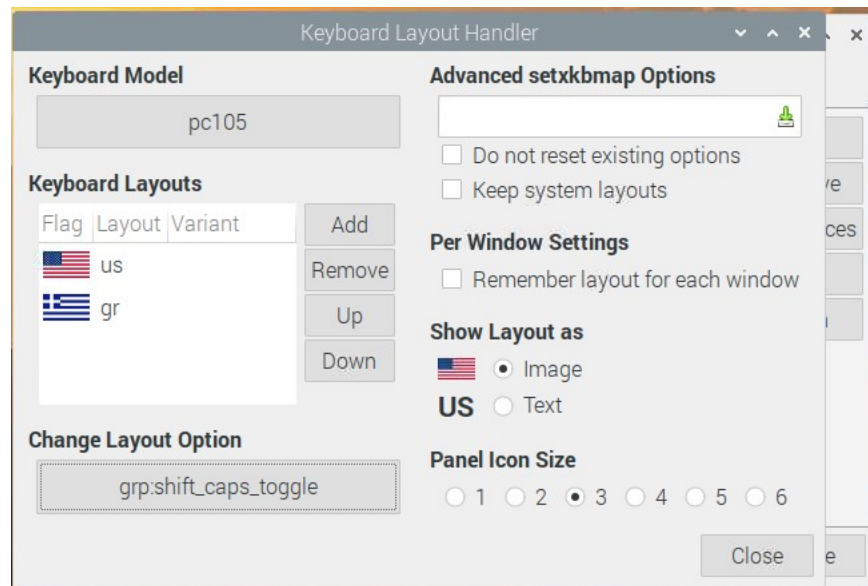
Εικόνα 47 Panel preferences – add plug in CPU TEMP

Για να προσθέσουμε κάποιο αντικείμενο πατάμε στο Add και από το παράθυρο που μας εμφανίζεται μπορούμε να επιλέξουμε ό,τι θέλουμε. Επιλέγουμε το CPU Temperature Monitor που αφορά την θερμοκρασία του επεξεργαστή. Με διπλό κλικ ή επιλέγοντας το Add που βρίσκεται στο κάτω μέρος του παραθύρου, το προσθέτουμε στο τέλος της μπάρας εργασιών, ώστε να είναι πάντοτε ορατή η θερμοκρασία του. Για την αποδοτική λειτουργία του επεξεργαστή θα πρέπει αυτή η θερμοκρασία να είναι χαμηλότερη των 80°. Έχουμε επίσης την δυνατότητα να αλλάξουμε τη θέση του με τα κουμπιά Up και Down.



Εικόνα 48 Panel preferences- add plug in Keyboard layout

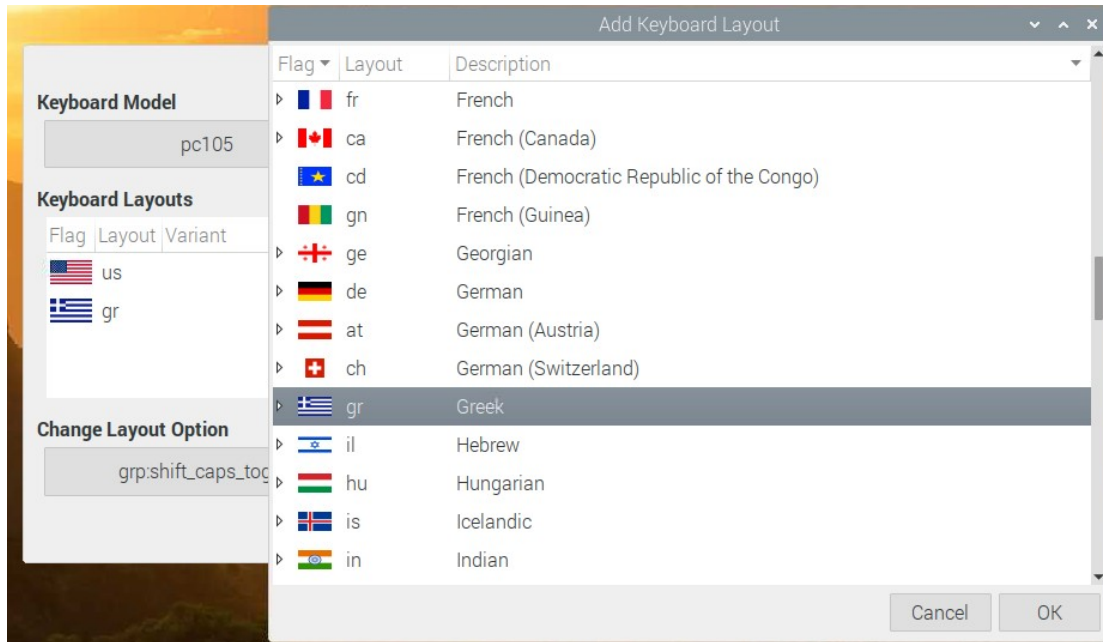
Επιπρόσθετα, αυτό που μπορούμε να προσθέσουμε στην μπάρα εργασιών είναι η βασική γλώσσα, η οποία δεν είναι η Αγγλική και αφορά στην αλλαγή γλώσσας πληκτρολογίου. Για να την προσθέσουμε επιλέγουμε το Keyboard Layout Handler. Μόλις το προσθέσουμε (Add) εμφανίζεται στο τέλος της μπάρας εργασιών μια σημαία. Τώρα μπορούμε να κλείσουμε το παράθυρο «Ιδιότητες πλαισίου».



Εικόνα 49 Keyboard layout Option

Για να συμπεριλάβουμε και την Ελληνική στις γλώσσες του πληκτρολογίου, πατάμε δεξί κλικ πάνω στη σημαία, όπου από το μενού που εμφανίζεται, επιλέγουμε το Keyboard

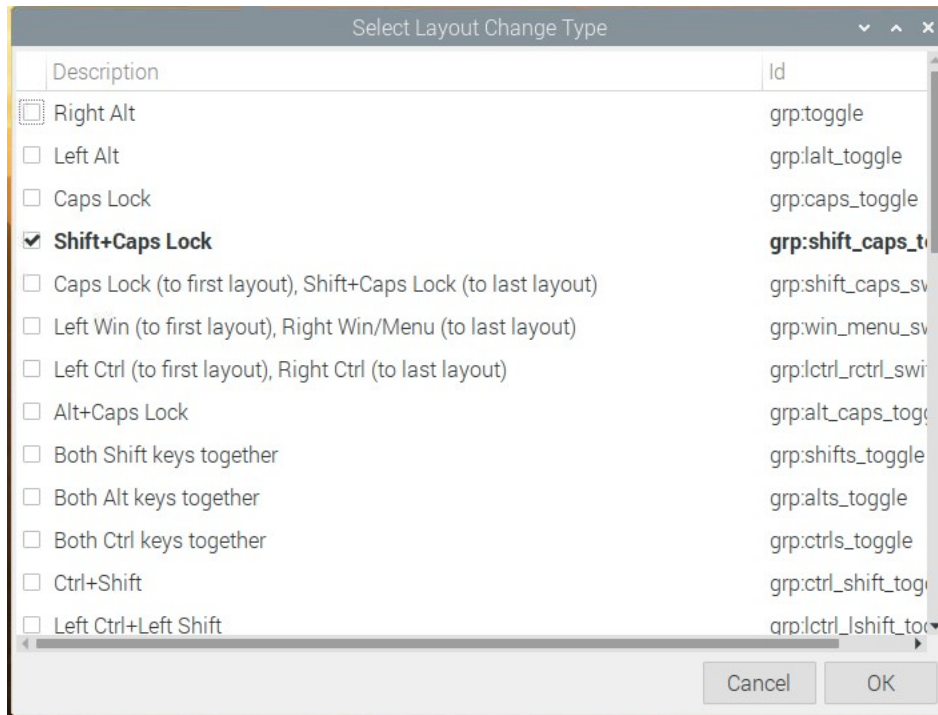
Layout Handler. Στις Keyboard Layouts βλέπουμε μόνο μια διαθέσιμη γλώσσα στη συγκεκριμένη εγκατάσταση, τα Αγγλικά Ηνωμένων Πολιτειών. Άρα θα πρέπει να προσθέσουμε και το Ελληνικό Πληκτρολόγιο. Για να το κάνουμε αυτό, το πρώτο βήμα είναι να αφαιρέσουμε την επιλογή Keep system layouts. Έπειτα μπορούμε να πατήσουμε στο Add και να επιλέξουμε την γλώσσα που επιθυμούμε.



Εικόνα 50 Keyboard layout change language

Πατώντας «OK» βλέπουμε στις Keyboard Layouts ότι έχει προστεθεί και η Ελληνική.

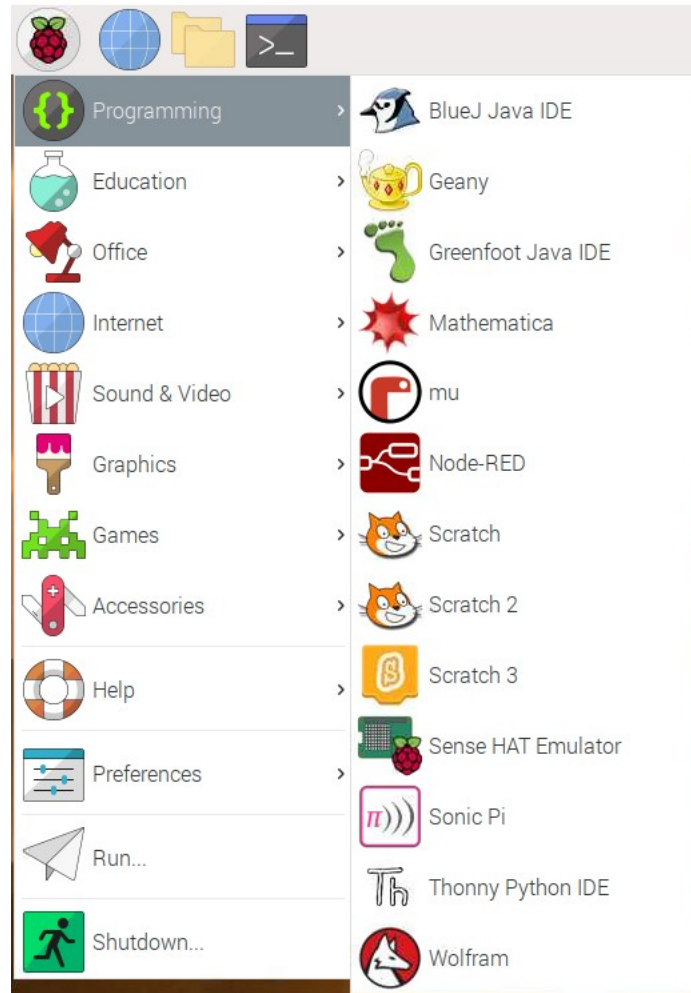
Τέλος επιλέγουμε την συντόμευση του πληκτρολογίου, ώστε να έχουμε τη δυνατότητα να αλλαγή γλώσσας. Αυτό εκτελείται με τη χρήση του μεγάλου κουμπιού που βλέπουμε κάτω από την ετικέτα Change Layout Option. Ο επιλεγμένος συνδυασμός πλήκτρων είναι το shift και caps lock και έχουμε τη δυνατότητα να αφαιρέσουμε τον επιλεγμένο συνδυασμό πλήκτρων και να προσθέσουμε ένα επιθυμητό συνδυασμό, για παράδειγμα το Ctrl+Shift .



Εικόνα 51 Keyboard layout- change layout option

Κάνουμε κλικ στο OK. Επιλέγουμε Close ώστε να αποθηκεύσουμε τις ρυθμίσεις.

Έχοντας τελειώσει με τις απαραίτητες ρυθμίσεις τώρα είμαστε έτοιμοι να ανοίξουμε το πρόγραμμα με το οποίο θα φτιάξουμε τον κώδικα για την εφαρμογή μας. Το πρόγραμμα αυτό ονομάζεται Thonny και το βρίσκουμε κάνοντας κλικ επάνω αριστερά στο εικονίδιο του raspberry pi και επιλέγοντας από τα προγράμματα (programms) Thonny Python IDE (βλ. Thonny Python)



Εικόνα 52 Programming- Thonny Python

Τώρα είμαστε έτοιμοι να ξεκινήσουμε την υλοποίηση του κώδικά μας σε 2 διαφορετικά αρχεία όπως περιγράψαμε πιο πάνω.

Παράρτημα Β

MOTION_DETECTION.PY

```
# -*- coding: utf-8 -*-

# Με την import cv2 εισάγουμε τη βιβλιοθήκη OpenCV

import cv2

# Εισάγουμε την datetime ώστε να έχουμε στη διάθεσή μας την ημερομηνία και ώρα

from datetime import datetime

# Ορίζουμε ως None τη μεταβλητή first_image που θα χρησιμοποιήσουμε αργότερα

first_frame = None

# Εδώ μπορούμε να ορίσουμε μεταβλητές ώστε να ελέγχουμε τις λειτουργίες του προγράμματός
# μας. Αν το SaveImage έχει την τιμή True, αποθηκεύονται φωτογραφίες με την εντοπισμένη
# κίνηση στον υπολογιστή μας. Αν και το SendEmails είναι True, το πρόγραμμα μάς ενημερώνει
# για την κίνηση αυτή και μέσω email όπως θα δούμε στο B4.3.3

SaveImage=True

SendEmails=True

if SendEmails:

    # Αν έχουμε επιλέξει την αποστολή emails, τότε εισάγουμε στο πρόγραμμά μας το πρόγραμμα
    # που έχουμε υλοποιήσει, δηλαδή το αρχείο sec_email.py

    import s_email

    # Εισάγουμε την time, η οποία μας επιστρέφει τον χρόνο σε δευτερόλεπτα. Θα χρειαστούμε
    # την time για να αποφύγουμε τη συνεχή αποστολή των emails

    from time import time
```

```
timeemailsnt=0
```

```
# Αλλάζουμε την τιμή στην παρακάτω μεταβλητή IpCam σε True, μόνο αν δεν έχουμε web-  
camera και επιθυμούμε να συνδέσουμε το κινητό μας τηλέφωνο μέσω της εφαρμογής IP  
Webcam
```

```
IpCam=False
```

```
# Με την if ελέγχουμε αν χρησιμοποιούμε την κάμερα του κινητού σύμφωνα με την παραπάνω  
μεταβλητή.
```

```
if IpCam:
```

```
    # Η urllib.request χρησιμοποιείται για να διαβάσουμε μια σελίδα ή κάποιο αντικείμενο από  
    το διαδίκτυο. Εδώ θα τη χρησιμοποιήσουμε για να διαβάζουμε εικόνες που δημοσιεύει η  
    εφαρμογή IP Webcam στο τοπικό μας δίκτυο.
```

```
import urllib.request
```

```
# Η numpy είναι μια μαθηματική βιβλιοθήκη που υποστηρίζει εύκολα πολυδιάστατους  
πίνακες και συναρτήσεις γραμμικής άλγεβρας.
```

```
import numpy
```

```
# Θα χρειαστεί να αλλάξετε τη διεύθυνση IP (192.168.1.198) στην παρακάτω μεταβλητή, ώστε  
να συμφωνεί με τη διεύθυνση που θα σας εμφανίσει η εφαρμογή IP Webcam κατά την εκτέλεσή  
της:
```

```
url="http://192.168.1.198:8080/shot.jpg"
```

else: # Οι παρακάτω εντολές μέσα στην else εκτελούνται αν IpCam=False Ορίζουμε το αντικείμενο video να αντιστοιχεί στην είσοδο βίντεο. Το 0 στη VideoCapture αντιστοιχεί στην πρώτη κάμερα που είναι συνδεδεμένη στη συσκευή μας:

```
video = cv2.VideoCapture(0)
```

```
# Ξεκινάμε ένα βρόχο που θα επεξεργάζεται κάθε ένα καρέ του βίντεο
```

```
while True:
```

```
    if IpCam:
```

```
        # Αν χρησιμοποιούμε την εφαρμογή IP Webcam, τότε με την urllib διαβάζουμε τη διεύθυνση της εικόνας όπως την έχουμε δηλώσει στη μεταβλητή url νωρίτερα:
```

```
        frame = urllib.request.urlopen(url)
```

```
        # Η numpy μας βοηθά να μετατρέψουμε την εικόνα που λαμβάνουμε από το κινητό σε πίνακα:
```

```
        frame = numpy.array(bytearray(frame.read()))
```

```
        # Με την imdecode αποκωδικοποιούμε την έγχρωμη εικόνα μας σε χρωματικά κανάλια BGR, ώστε να μπορέσουμε να τη χρησιμοποιήσουμε με τις εντολές της OpenCV αργότερα:
```

```
        frame = cv2.imdecode(frame,cv2.IMREAD_UNCHANGED)
```

```
    else: # Οι παρακάτω εντολές μέσα στην else εκτελούνται αν IpCam=False Με τη video.read() διαβάζουμε ένα καρέ(frame) από το βίντεο. Στη μεταβλητή ret επιστρέφει True, αν έχουμε διαβάσει το καρέ χωρίς προβλήματα, ή False αν κάτι δεν πήγε καλά κατά την ανάγνωση του καρέ.
```

```
    ret, frame = video.read()
```

Στη μεταβλητή text θα αποθηκεύσουμε αργότερα το μήνυμα που θέλουμε να εμφανίζεται στο βίντεο κατά τη διάρκεια του εντοπισμού κίνησης:

```
text = ""
```

Με την cvtColor() μετατρέπουμε το χρώμα σύμφωνα με το όρισμα που θα δώσουμε. Με το όρισμα cv2.COLOR_BGR2GRAY μετατρέπουμε την έγχρωμη εικόνα που έχουμε λάβει από την κάμερα σε αποχρώσεις του γκριζου:

```
gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
```

Με τη cv2.GaussianBlur θολώνουμε την εικόνα, ώστε να εντοπιστεί πιο εύκολα η αλλαγή (η κίνηση). Το πρώτο όρισμά της είναι η πηγή, ενώ το ζευγάρι αριθμών που ακολουθεί είναι το μέγεθος του πυρήνα σε Pixels. Όσο μεγαλύτερος ο πυρήνας, τόσο περισσότερο θα θολώνει η εικόνα. Αυτό το ζευγάρι τιμών ίσως χρειαστεί να το αλλάξει προς τα κάτω, αν η εφαρμογή δεν εντοπίζει κίνηση ή προς τα πάνω, αν εντοπίζει κίνηση σε σημεία όπου δεν υπάρχει κάποια αλλαγή στην εικόνα.

```
gray = cv2.GaussianBlur(gray, (61, 61), 0)
```

Στην πρώτη επανάληψη της while αποθηκεύουμε το πρώτο καρέ στη first_frame

if first_frame is None:

```
    first_frame = gray
```

Με την continue, συνεχίζουμε με την επόμενη επανάληψη της while

```
    continue
```

Κάθε νέο καρέ που διαβάζουμε, το συγκρίνουμε με το αρχικό με χρήση της cv2.absdiff() και τη διαφορά την αποθηκεύουμε στη diff_frame

```
diff_frame = cv2.absdiff(first_frame, gray)
```

Με την cv2.threshold() ελέγχουμε τη διαφορά κάθε εικονοστοιχείου του τρέχοντος καρέ σε σχέση με το αρχικό. Αν η διαφορά μεταξύ τους είναι πάνω από το 75, όπως έχει οριστεί στην cv2.threshold(), τότε θα εμφανίσει λευκό χρώμα(255) στα αντίστοιχα εικονοστοιχεία, τα οποία αποθηκεύονται στο thresh_frame. Το 75 μπορούμε να το μειώσουμε αν δεν μπορεί να εντοπίσει μεταβολή στην εικόνα, ή να το αυξήσουμε στην περίπτωση που βρίσκει μεταβολές στα σημεία που δεν υπάρχουν.

```
ret, thresh_frame = cv2.threshold(diff_frame, 75, 255, cv2.THRESH_BINARY)
```

Με την cv2.dilate() αυξάνουμε το μέγεθος του αντικειμένου που εντοπίστηκε από την cv2.threshold() -Το κάνουμε πιο διακριτό-

```
thresh_frame = cv2.dilate(thresh_frame, None, iterations = 2)
```

Η cv2.findContours() βρίσκει τα περιγράμματα των αντικειμένων στη thresh_frame τα οποία αποθηκεύονται στην λίστα contours.

```
contours,_ = cv2.findContours(thresh_frame,  
cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)
```

Με τη for επεξεργαζόμαστε κάθε περίγραμμα(countour) που έχει βρεθεί.

```
for contour in contours:
```



```
# Αν η περιοχή την οποία περικλείει το περίγραμμα καταλαμβάνει λιγότερα από 10000 pixels
```

```
if cv2.contourArea(contour) < 10000:
```

```
# τότε η παραπάνω if την αγνοεί και προχωράει στο επόμενο περίγραμμα που θα προσπελάσει η for.
```

```
continue
```

```
# Η boundingRect() μας επιστρέφει τις συντεταγμένες ενός ορθογωνίου που περιέχει το τρέχον πρόγραμμα. Συγκεκριμένα, μας επιστρέφει τις συντεταγμένες της πάνω δεξιάς γωνίας του (x,y), όπως και το πλάτος (w), και το ύψος (y) του ορθογωνίου.
```

```
(x, y, w, h) = cv2.boundingRect(contour)
```

```
# Η μέθοδος cv2.rectangle(εικόνα, αρχή, τέλος τετραγώνου, χρώμα*, πάχος γραμμής) στην παρακάτω γραμμή ζωγραφίζει ένα περίγραμμα γύρω από το αντικείμενο στο τρέχον καρέ:
```

```
cv2.rectangle(frame, (x, y), (x + w, y + h), (0, 255, 0), 3)
```

```
# *Το χρώμα στην OpenCV αναπαριστάται σε τριάδα μπλε πράσινο κόκκινο (Blue Green Red-BGR). Εφόσον έχουμε βρει αντικείμενα, διαμορφώνουμε ανάλογα το ενημερωτικό μήνυμα που μπορούμε να δείξουμε στο τρέχον καρέ:
```

```
text = "Movement Detected!!!"
```

```
# Με τη cv2.putText προσθέτουμε κείμενο στο τρέχον καρέ. Τα ορίσματα που δέχεται είναι cv2.putText(καρέ, κείμενο, (συντεταγμένες x, y της κάτω αριστερής γωνίας που θα ξεκινήσει το
```

κειμένο), γραμματοσειρά, μέγεθος γραμματοσειράς, χρώμα του κειμένου σε BGR, πάχος γραμματοσειράς):

```
cv2.putText(frame, text, (20, 30),  
  
cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 0, 255), 2)  
  
# Εδώ προσθέτουμε την τρέχουσα ημερομηνία και ώρα, μορφοποιημένες με την strftime()  
  
cv2.putText(frame, datetime.now().strftime("%d/%m/%Y %H:%M:%S"),  
  
(10, frame.shape[0] - 10), cv2.FONT_HERSHEY_SIMPLEX, 0.8, (0, 0, 255), 3)
```

Με την cv2.imshow(τίτλος παραθύρου, εικόνα) εμφανίζουμε τις εικόνες μας σε παράθυρο στο γραφικό περιβάλλον, μαζί με ένα τίτλο. Η παρακάτω εντολή εμφανίζει την εικόνα σε αποχρώσεις του γκριζου, όπως την είχαμε μετατρέψει νωρίτερα:

```
cv2.imshow("Grayscale", gray)
```

Η παρακάτω εντολή εμφανίζει μόνο τα εικονοστοιχεία της γκριζας εικόνας που έχουν διαφορά από το αρχικό καρέ:

```
cv2.imshow("Difference", diff_frame)
```

Παρακάτω εμφανίζουμε μια εικόνα με δύο μόνο χρώματα (μαύρο-άσπρο), όπου με άσπρο χρώμα απεικονίζουμε τα εικονοστοιχεία που είχαν διαφορά από το αρχικό καρέ, μεγαλύτερη της τιμής που είχαμε ορίσει νωρίτερα στην cv2.threshold()

```
cv2.imshow("Threshold", thresh_frame)
```

Τέλος εμφανίζουμε το τρέχον έγχρωμο καρέ, μαζί με τα τυχόν περιγράμματα των αντικειμένων που εντοπίσαμε νωρίτερα.

```
cv2.imshow("Color", frame)
```

Αυτό το έγχρωμο καρέ είναι και το μόνο απαραίτητο να εμφανίζεται. Τα προηγούμενα είναι περισσότερο χρήσιμα για να μπορέσουμε πιο εύκολα να ρυθμίσουμε την εφαρμογή, ανάλογα με τις προδιαγραφές των καμερών και τις απαιτήσεις μας. Η παρακάτω if ελέγχει αν έχει εντοπιστεί κίνηση και αν έχουμε ορίσει την τιμή TRUE στη μεταβλητή SaveImage στην αρχή του προγράμματος. Αν ισχύουν τα παραπάνω, η εικόνα αποθηκεύεται όπως έχει διαμορφωθεί νωρίτερα. Για να εξετάσουμε αν έχει εντοπιστεί κίνηση, μπορούμε να χρησιμοποιήσουμε τη μεταβλητή text η οποία σε αυτή την περίπτωση περιέχει το λεκτικό "Movement Detected!!!". Έτσι, αν το αποτέλεσμα της len(text) είναι μεγαλύτερο του μηδενός, πράγμα που σημαίνει ότι περιέχει χαρακτήρες, καταλαβαίνουμε πως έχει εντοπιστεί κίνηση.

```
if len(text)>0 and SaveImage:
```

```
# Δίνουμε στο όνομα του αρχείου την τρέχουσα ημερομηνία και ώρα:
```

```
FN="YMD"+datetime.now().strftime("%Y-%m-%d_%H:%M:%S") +".jpg"
```

```
# η cv2.imwrite() αποθηκεύει το καρέ μας(frame) στο όνομα(FN) που έχουμε ορίσει:
```

```
cv2.imwrite(FN,frame)
```

Στην if, εκτός από τη μεταβλητή SendEmails, ελέγχουμε και αν ο χρόνος που έχει περάσει από την τελευταία αποστολή των email είναι άνω των 30 δευτερολέπτων, για την αποφυγή αποστολής πολλαπλών μηνυμάτων.

```
if SendEmails and time()-timeemailsnt>30:
```

```
s_email.send_email(FN)
```

```
# Στη μεταβλητή timeemailsnt αποθηκεύουμε τη χρονική στιγμή που έφυγε το email:
```

```
timeemailsnt=time()
```

```
# Η cv2.waitKey(1) μπορεί να δεχτεί ένα χαρακτήρα από το πληκτρολόγιο του χρήστη

key = cv2.waitKey(1)

# αν ο χρήστης πληκτρολογήσει το q, τότε θα τερματιστεί ο βρόγχος while True

if key == ord('q'):

    break

# Αφού τερματίσουμε το βρόγχο με το πλήκτρο q, το πρόγραμμά μας αποδεσμεύει τη web
κάμερα.

if IpCam == False:

    video.release()

# Η destroyAllWindows() κλείνει όλα τα παράθυρα που είχαμε ανοίξει προηγουμένως με την
cv2.imshow()

cv2.destroyAllWindows()
```

Παράρτημα Γ

SEND EMAIL CODE

S_EMAIL.PY

```
# -*- coding: utf-8 -*-

# Εισάγουμε τη βιβλιοθήκη os η οποία μας δίνει πρόσβαση στα αρχεία μας:

import os

# Η smtplib αναλαμβάνει να στείλει το email μέσω του πρωτοκόλλου smtp:

import smtplib

# Προσθέτουμε μερικά εργαλεία που θα μας χρειαστούν για τη μορφοποίηση του ηλεκτρονικού
μηνύματος:

from email.mime.text import MIMEText

from email.mime.image import MIMEImage

from email.mime.multipart import MIMEMultipart

# Οι παρακάτω δύο μεταβλητές θα πρέπει να αλλάξουν σύμφωνα με τα δικά σας στοιχεία:

addressF= 'adressFROM@gmail.com'

password = 'your device Password'

addressTo='AdressTo@gmail.com'

path="/home/pi/Test/Test1/"
```

Υλοποιούμε την παρακάτω συνάρτηση την οποία θα καλέσουμε μέσα από το πρόγραμμα ανίχνευσης κίνησης που δημιουργήσαμε στο B4.3.2 Αυτή η συνάρτηση δέχεται ως όρισμα το όνομα του αρχείου που θέλουμε να στείλουμε:

```
def send_email(FN):
```

```
    # Έπειτα διαβάζουμε το όνομα του αρχείου με την open():
```

```
    FIn=path+FN
```

```
    img_data = open(FIn, 'rb').read()
```

```
    # ορίζουμε το αντικείμενο msg για να δομήσουμε το μήνυμά μας:
```

```
    msg = MIMEMultipart()
```

```
    # Ορίζουμε το θέμα του μηνύματος (Subject):
```

```
    msg['Subject'] = 'Εντοπίστηκε κίνηση'
```

```
    # Ορίζουμε τη διεύθυνση του αποστολέα και του παραλήπτη:
```

```
    msg['From'] = addressF
```

```
    msg['To'] = addressTo
```

```
    # Ορίζουμε το κείμενο του μηνυματός μας και το ενσωματώνουμε στο email μας με τη  
msg.attach():
```

```
    text = MIMEText("Κάποιος είναι στο σπίτι.")
```

```
    msg.attach(text)
```

```
    # Ενσωματώνουμε στο μήνυμά μας και την εικόνα:
```

```
image = MIMEImage(img_data, name=os.path.basename(FN))

msg.attach(image)

try:

    # Δοκιμάζουμε να συνδεθούμε με τον εξυπηρετητή smtp του gmail:

    srv = smtplib.SMTP('smtp.gmail.com',587)

    srv.ehlo()

    srv.starttls()

    srv.login(addressF , password)

    # η sendmail() δέχεται τρεις παραμέτρους: τον αποστολέα, τον παραλήπτη και το μήνυμα,
    όπως το διαμορφώσαμε στο αντικείμενο msg:

    srv.sendmail(addressF, addressTo, msg.as_string())

# Μόλις φύγει το μήνυμα, κλείνουμε τη σύνδεση με τον εξυπηρετητή smtp:

srv.quit()

print("Το μήνυμα εστάλει")

except:

# Η try-except είναι ένας μηχανισμός της python ο οποίος αρχικά δοκιμάζει να εκτελέσει τις
εντολές που βρίσκονται μέσα στην try. Αν κάποια από τις εντολές εμφανίσει σφάλμα, αντί να
τερματιστεί η εφαρμογή, εκτελεί τις εντολές που έχουμε ορίσει μέσα στην except. Αν για
παράδειγμα δεν καταφέρουμε να συνδεθούμε στον εξυπηρετητή smtp, τότε εκτελείται το
περιεχόμενο της except και μας εμφανίζει στην κονσόλα το μήνυμα “Κάτι δεν πήγε καλά”:

print("Κάτι δεν πήγε καλά")
```


Παράρτημα Δ

Πώς λειτουργεί το Cayenne

Το Cayenne είναι ο πρώτος κατασκευαστής έργων IoT drag and drop στον κόσμο που δίνει τη δυνατότητα σε προγραμματιστές, σχεδιαστές και μηχανικούς να κάνουν πρωτότυπο και να μοιράζονται τα έργα των συνδεδεμένων συσκευών τους. Το Cayenne σχεδιάστηκε για να βοηθήσει τους χρήστες να δημιουργήσουν πρωτότυπα Internet of Things και στη συνέχεια να τα φέρουν στην παραγωγή.

Το κύριο συστατικό της πλατφόρμας:

Cayenne Online Dashboard - Χρησιμοποιήστε προσαρμόσιμα γραφικά στοιχεία για να οπτικοποιήσετε δεδομένα, να ρυθμίσετε κανόνες, να προγραμματίσετε εκδηλώσεις και άλλα.

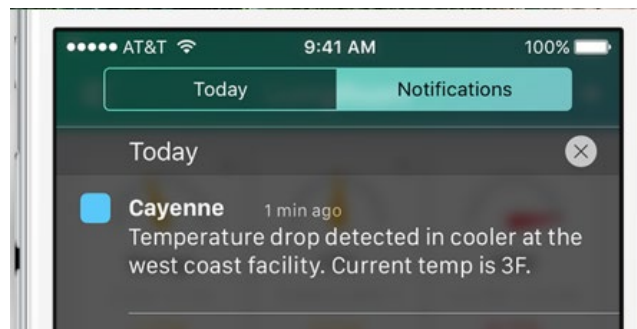
Τι χρειάζομαι;

- **Συσκευές, εξαρτήματα**
 - Μια συσκευή Raspberry Pi ή Arduino συνδεδεμένη στο Διαδίκτυο ή μια συσκευή LoRa συνδεδεμένη σε δημόσια ή ιδιωτική πύλη. Η λίστα του υλικού που λειτουργεί με το Cayenne θα συνεχίσει να αυξάνεται.
- **Πρόγραμμα περιήγησης**
 - Το Cayenne έχει σχεδιαστεί για να λειτουργεί σε δημοφιλή προγράμματα περιήγησης.* Προοδευτική εφαρμογή Ιστού (PWA) για iOS και Android σύντομα.

Ειδοποιήσεις:

Έγκαιρες ειδοποιήσεις:

Η Cayenne διαθέτει ενσωματωμένη μηχανή κανόνων για εύκολη ρύθμιση ειδοποιήσεων email και κειμένου, όταν μια συσκευή ή αισθητήρας φτάνει σε προκαθορισμένη κατάσταση.



Εικόνα 53 Cayenne platform - Notifications

Ενσωματωμένο IF / THEN

Με το Cayenne πλήρως ενσωματωμένο μηχανισμό κανόνων, το μόνο που έχετε να κάνετε είναι να μεταφέρετε και να αποθέσετε συσκευές στην ενότητα Triggers & Alerts για να δημιουργήσετε ειδοποιήσεις κατωφλίου σε έναν μόνο παραλήπτη ή ομάδα.



Εικόνα 54 Cayenne platform - Notification Settings

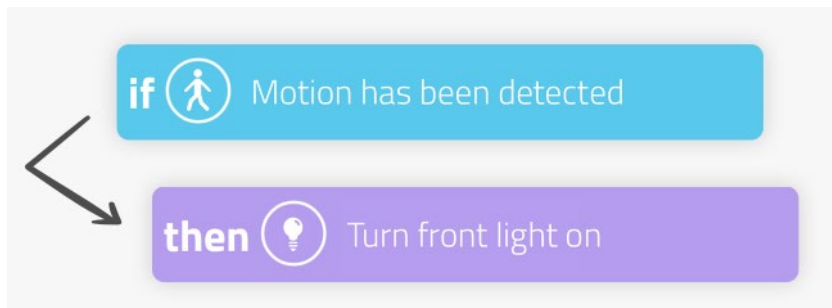
Ρύθμιση ειδοποιήσεων

Η δημιουργία ειδοποιήσεων με ενεργοποιημένα συμβάντα είναι εύκολη. Διαχειριστείτε τις ειδοποιήσεις σας ονομάζοντας πρώτα την ετικέτα σας και προσαρμόζοντας τα όρια κατωφλίου από οποιαδήποτε δεδομένα που σχετίζονται με συνδεδεμένες συσκευές.

Cayenne Triggers:

Δημιουργικός αυτοματισμός

Μόλις συνδέσετε την πρώτη σας συσκευή στο έργο, το Cayenne δίνει τη δυνατότητα στους κατασκευαστές να δημιουργήσουν αυτοματοποιημένα έργα IoT μέσα σε λίγα λεπτά.



Εικόνα 55 Cayenne platform - Triggers

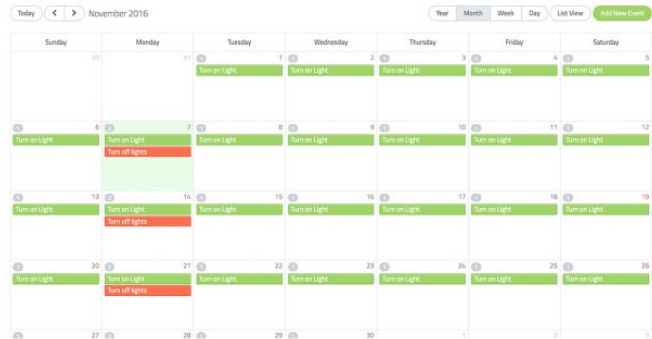
Με βάση τα δεδομένα

Τα δεδομένα της συσκευής που καταγράφονται σε πραγματικό χρόνο επιτρέπουν την εκτέλεση συμβάντων που ενεργοποιούνται. Αυτό επιτρέπει την απρόσκοπτη επικοινωνία μεταξύ συσκευών ανεξαρτήτως κατασκευαστή.

Χρονοπρογραμματισμός:

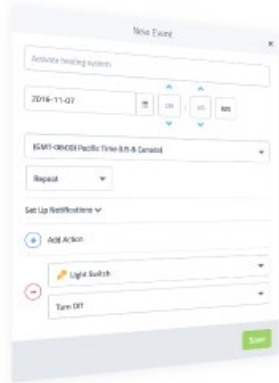
Προγραμματίστε ενέργειες:

Το υλικό που συνδέεται με Cayenne επωφελείται από μια ενσωματωμένη δυνατότητα προγραμματισμού για τον εύκολο γρήγορο προγραμματισμό εκδηλώσεων βάσει χρόνου.



ΤΟ
ΚΑΙ

Εικόνα 56 Cayenne platform - Χρονοπρογραμματισμός

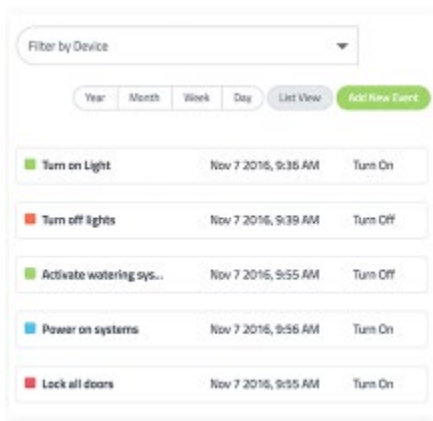


Απλή ρύθμιση

Η οθόνη εκδηλώσεων παρέχει μια απλοποιημένη προσέγγιση για τη δημιουργία χρονοδιαγραμμάτων. Ονομάστε το συμβάν, την ημερομηνία και την ώρα εμφάνισης και τη συχνότητα. Μπορείτε επίσης να ρυθμίσετε μια ειδοποίηση για μια ειδοποίηση μετά το συμβάν.

Εικόνα 57 Cayenne platform - settings

Προγραμματισμός προβολών



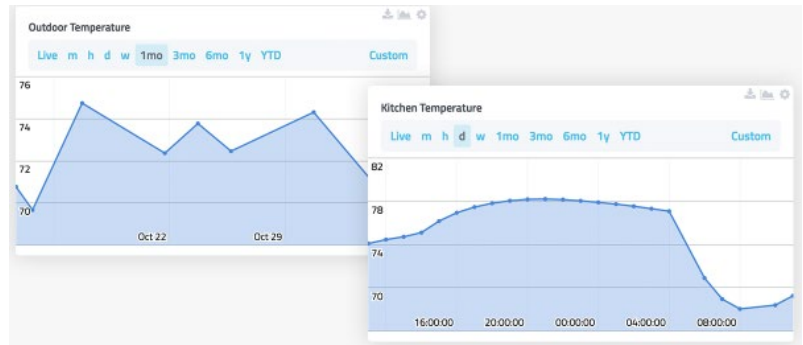
Μπορείτε εύκολα να παρακολουθείτε όλες τις εκδηλώσεις σας. Μεταβείτε απλώς στην ενότητα εκδηλώσεων, όπου μπορείτε να προβάλετε τα συμβάντα ανά έτος, μήνα, εβδομάδα, ημέρα και ως πλήρη λίστα. Μπορείτε επίσης να φιλτράρετε ανά συσκευή.

Εικόνα 58 Cayenne platform - Προγραμματισμός προβολών

Διαχείριση Δεδομένων

Ροές δεδομένων πίνακα ελέγχου

Το Cayenne εμφανίζει και αποθηκεύει δεδομένα από συσκευές και αισθητήρες παρέχοντας ορατότητα σε μοτίβα συμπεριφοράς που υποστηρίζουν βελτιώσεις σε έργα IoT.



Εικόνα 59 Cayenne platform - Ροές Δεδομένων



Εικόνα 60 Cayenne platform - Προβολή δεδομένων

Τα δεδομένα σου! Ο τρόπος σου!

Υπάρχουν πολλοί τρόποι προβολής δεδομένων στο Cayenne.

Τα έτοιμα για προγραμματισμό προσαρμοσμένα widget μπορούν να χρησιμοποιηθούν για την εμφάνιση συλλεγμένων δεδομένων με πολλούς διαφορετικούς τρόπους.

Συγκεκριμένα, η τιμή και τα γραφήματα γραφημάτων

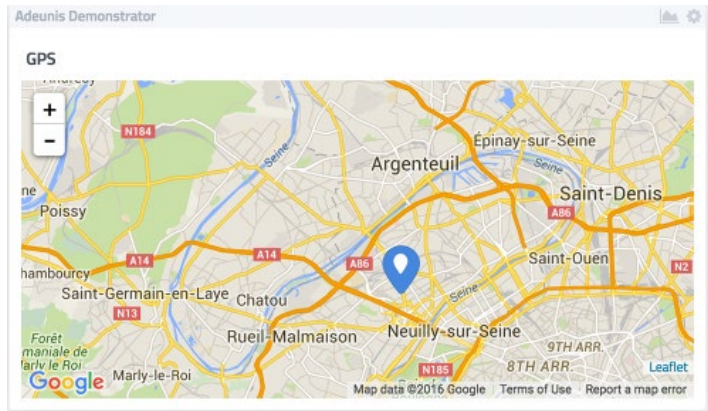
Προβολές δεδομένων:

Τα ζωντανά και τα ιστορικά δεδομένα μπορούν να φιλτραριστούν ανά λεπτό, ώρα, ημέρα, εβδομάδα, μήνα, έτος ή ένα προσαρμοσμένο εύρος ημερομηνιών. Επιπλέον λεπτομέρειες μπορούν να εμφανιστούν ανοίγοντας μια εκτεταμένη προβολή γραφήματος. Κατεβάστε εύκολα δεδομένα με μερικά κλικ.

Παρακολούθηση σε χάρτη:

Παρακολούθηση τοποθεσίας

Μπορείτε να εντοπίσετε μια χαμένη συσκευή ή να παρακολουθήσετε την κίνηση των συσκευών σας κατά τη διάρκεια της ημέρας! Μπορείτε να δείτε την προηγούμενη και την παρούσα τοποθεσία όλων των συσκευών σας που είναι συνδεδεμένες στο Cayenne.



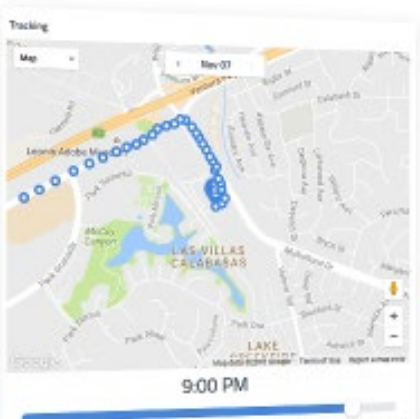
Εικόνα 61 Παρακολούθηση τοποθεσίας

Τύποι παρακολούθησης στοιχείων



Εικόνα 62 Τύποι παρακολούθησης στοιχείων

Ανάλογα με τον τύπο της προστιθέμενης συσκευής, υπάρχουν πολλές διαθέσιμες ρυθμίσεις τοποθεσίας για την ενεργοποίηση της παρακολούθησης στοιχείων. Αυτές περιλαμβάνουν συσκευές που είναι σταθερές ή κινητές και συνδεδεμένοι ή ασύρματοι αισθητήρες.



Εικόνα 63 Οπτικοποίηση στοιχείων

Οπτικοποίηση στοιχείων

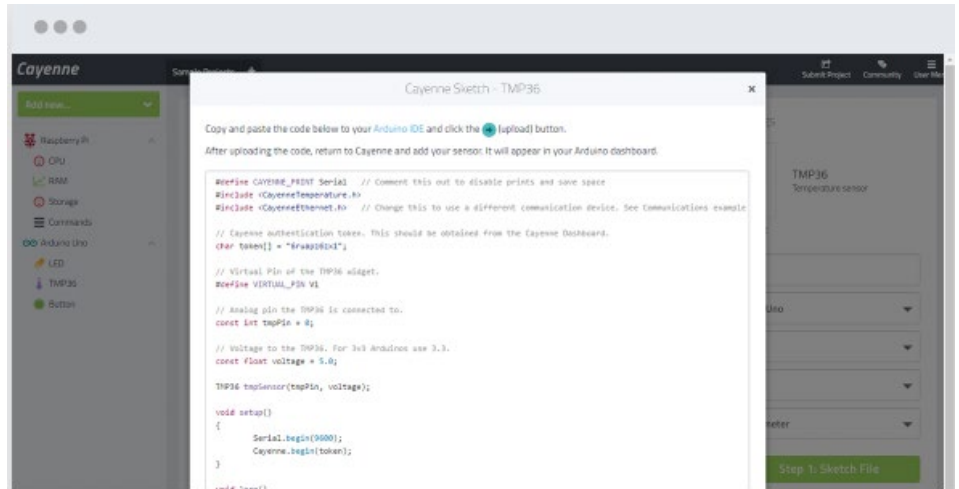
Οπτικοποιήστε τις τρέχουσες και τις προηγούμενες τοποθεσίες συσκευών με το widget Cayenne Device Map.

Παρέχει μια προβολή χάρτη του ιστορικού τοποθεσίας για μία μόνο συσκευή ή οποιονδήποτε αισθητήρα.

Προσαρμοσμένος κώδικας:

Πλήρης ευελιξία

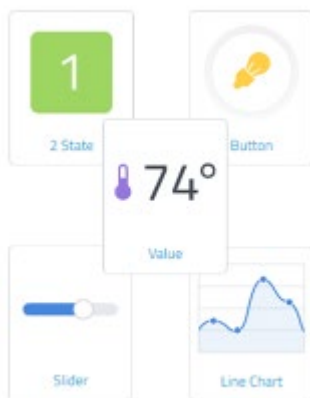
Ενσωματώνετε εύκολα τις συσκευές σας που χρησιμοποιούν προσαρμοσμένο κώδικα, την προβολή και τον έλεγχο τους χρησιμοποιώντας τον ίδιο πίνακα ελέγχου και υποστήριξη εφαρμογών για κινητά που διαθέτει το Cayenne για εγγενείς συσκευές.



Εικόνα 64 Ενσωμάτωση συσκευών στην πλατφόρμα Cayenne

Sketch Files

Η σύνταξη προσαρμοσμένου κώδικα για συσκευές συνδεδεμένες σε Arduino πραγματοποιείται χρησιμοποιώντας αρχεία σκίτσων. Γράφοντας τον κωδικό σας στο αρχείο σκίτσων, έχετε τον πλήρη έλεγχο του τρόπου αποστολής των δεδομένων σας στο Cayenne.



Εικόνα 65 Προσαρμοσμένα widget

Προσαρμοσμένα widget

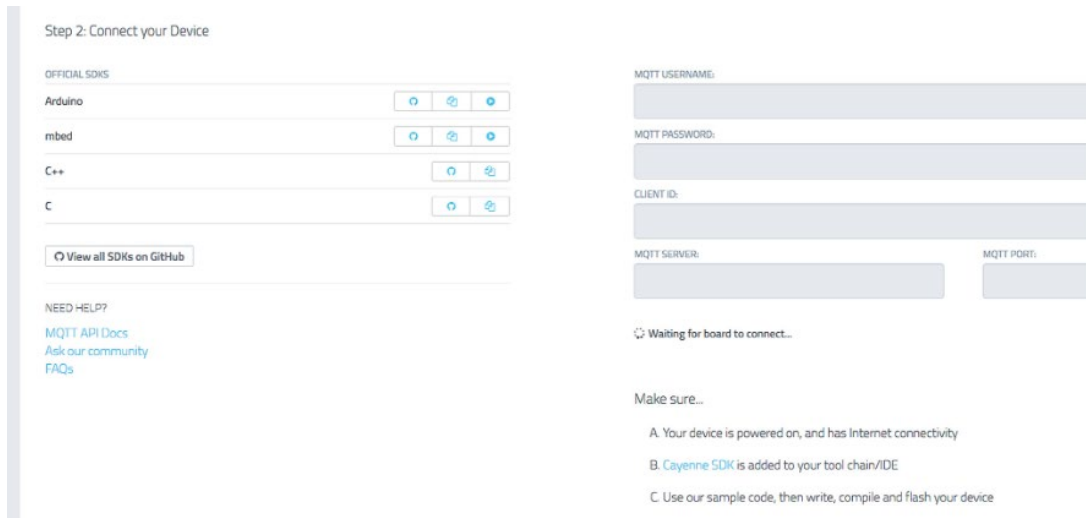
Για μεγαλύτερο έλεγχο υλικού που δεν υπάρχει στη βιβλιοθήκη συσκευών, μπορείτε να χρησιμοποιήσετε προσαρμοσμένα γραφικά στοιχεία και να προσθέσετε τον προσαρμοσμένο κώδικα που χρειάζεστε για παράδειγμα αρχεία σκίτσων.

Δείγματα Sketch Files είναι διαθέσιμα στο GitHub.

```
18 #include "CayenneMQTTClient.h"
19 #include <string.h>
20
21 void MQTTMessageArrived(MessageData* md, void* userData)
22 {
23     CayenneMQTTClient* client = (CayenneMQTTClient*)userData;
24     if (client)
25     {
26         int i;
27         int result = MQTT_FAILURE;
28         CayenneMessageData message;
29
30         result = CayenneParseTopic(&message.topic, &message.channel, &message.clientID, client->username, md->topicName);
31         if (result != CAYENNE_SUCCESS)
32             return;
33         //Null terminate the string since that is required by CayenneParsePayload. The readbuf is set to CAYENNE_MAX_MESSAGE_VALUES
34         ((char*)md->message->payload)[md->message->payloadlen] = '\0';
35         message.valueCount = CAYENNE_MAX_MESSAGE_VALUES;
36         result = CayenneParsePayload(message.values, &message.valueCount, &message.type, &message.id, message.topic, (CayenneMQTTClient*)userData);
37         if (result != CAYENNE_SUCCESS)
38             return;
39
40         result = MQTT_FAILURE;
```

Εικόνα 66 Sketch Files

MQTT API:



Εικόνα 67 MQTT API

API MQTT

Συνδέστε εύκολα οποιαδήποτε συσκευή στο Cayenne Cloud. Μόλις ενσωματωθεί η συσκευή σας, θα επωφεληθεί από όλες τις δυνατότητες IoT που είναι διαθέσιμες στο Cayenne σήμερα.

Απλοποιημένο API

Υπάρχουν πολλές διαφορετικές διαδρομές για να εισαγάγετε το υλικό σας μέσω MQTT. Τα επίσημα SDK είναι Embedded C, Embedded C++ και Arduino MQTT. Στη συνέχεια, οι συσκευές θα στείλουν δεδομένα στον πίνακα ελέγχου και θα τα εμφανίσουν με προσαρμοσμένα widget



Τύποι δεδομένων

Το Cayenne υποστηρίζει διάφορους τύπους δεδομένων, όπως, εγγύτητα, θερμοκρασία, φωτεινότητα και πίεση. Οι τύποι δεδομένων και η μονάδα μέτρησης μπορούν να διαμορφωθούν από τον πίνακα ελέγχου ή να ρυθμιστούν απευθείας κατά την αποστολή δεδομένων.

Εικόνα 68 Τύποι δεδομένων

Στην Εικόνα 69 θα αναφέρουμε κάποια χαρακτηριστικά για την συσκευή Raspberry Pi 4 που χρησιμοποιήσαμε, όπως αυτά αναφέρονται στη σελίδα των κατασκευαστών [47].



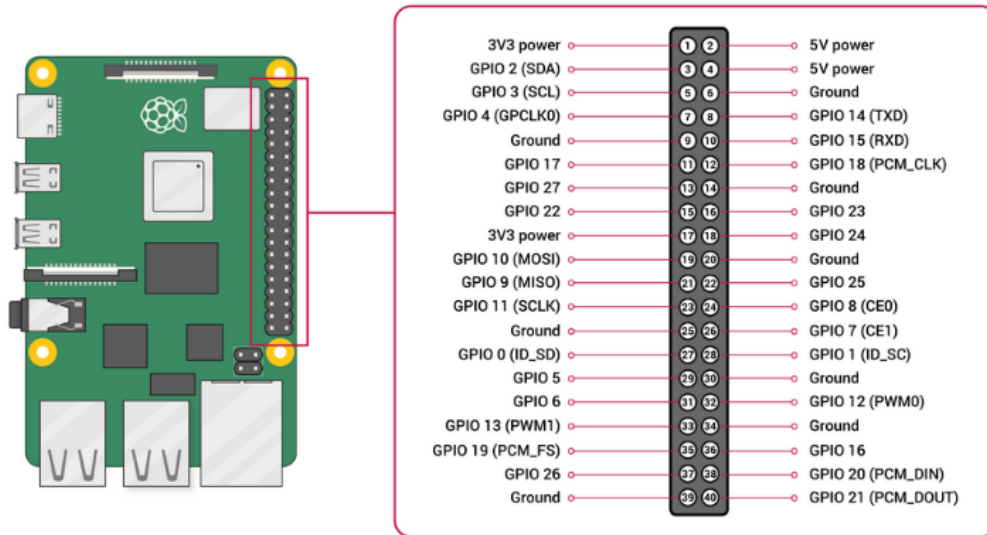
Εικόνα 69 Raspberry pi 4

Το Raspberry Pi Compute Module 4 (CM4) είναι ένα System on Module (SoM) που περιέχει επεξεργαστή, μνήμη, eMMC Flash και υποστηρικτικό κύκλωμα ισχύος. Αυτές οι μονάδες επιτρέπουν σε έναν σχεδιαστή να αξιοποιήσει το υλικό και το λογισμικό του Raspberry Pi σε δικά του προσαρμοσμένα συστήματα και τύπους. Επιπλέον, αυτές οι μονάδες διαθέτουν επιπλέον διεπαφές IO πέρα και πάνω από ό,τι είναι διαθέσιμο στις πλακέτες Raspberry Pi, ανοίγοντας περισσότερες επιλογές για τον σχεδιαστή. Τα τεχνικά χαρακτηριστικά του αναγράφονται πιο κάτω.

- Broadcom BCM2711, Quad core Cortex-A72 (ARM v8) 64-bit SoC @ 1.5GHz
- Small Footprint 55mm × 40mm × 4.7mm module
 - 4 × M2.5 mounting holes
- H.265 (HEVC) (upto 4Kp60 decode), H.264 (upto 1080p60 decode, 1080p30 encode)
- OpenGL ES 3.0 graphics
- Options for 1GB, 2GB, 4GB or 8GB LPDDR4-3200 SDRAM with ECC (see Appendix B)
- Options for 0GB (CM4Lite), 8GB, 16GB, or 32GB eMMC Flash memory (see Appendix B)
- Peak eMMC bandwidth 100MBytes/s (four times faster than previous Compute modules)
- Option (see Appendix B) for certified radio module with:
 - 2.4 GHz, 5.0 GHz IEEE 802.11 b/g/n/ac wireless
 - Bluetooth 5.0, BLE
 - On board electronic switch to select between PCB trace or external antenna
- Gigabit Ethernet PHY supporting IEEE 1588
- 1 × PCIe 1-lane Host, Gen 2 (5Gbps)
- 1 × USB 2.0 port (highspeed)
- 28 × GPIO supporting either 1.8v or 3.3v signalling and peripheral options:
 - Up to 5 × UART
 - Up to 5 × I2C
 - Up to 5 × SPI
 - 1 × SDIO interface
 - 1 × DPI (Parallel RGB Display)
 - 1 × PCM
 - Up to 2× PWM channels
 - Up to 3× GPCLK outputs
- 2 × HDMI 2.0 ports (up to 4Kp60 supported)
- MIPI DSI:
 - 1 × 2-lane MIPI DSI display port
 - 1 × 4-lane MIPI DSI display port
- MIPI CSI-2:
 - 1 × 2-lane MIPI CSI camera port
 - 1 × 4-lane MIPI CSI camera port
- 1 × SDIO 2.0 (CM4Lite)

- Single +5v PSU input.

Στην επόμενη εικόνα εμφανίζονται οι GPIO θήρες του Raspberry Pi 4.



Εικόνα 70 GPIO θήρες του Raspberry Pi 4

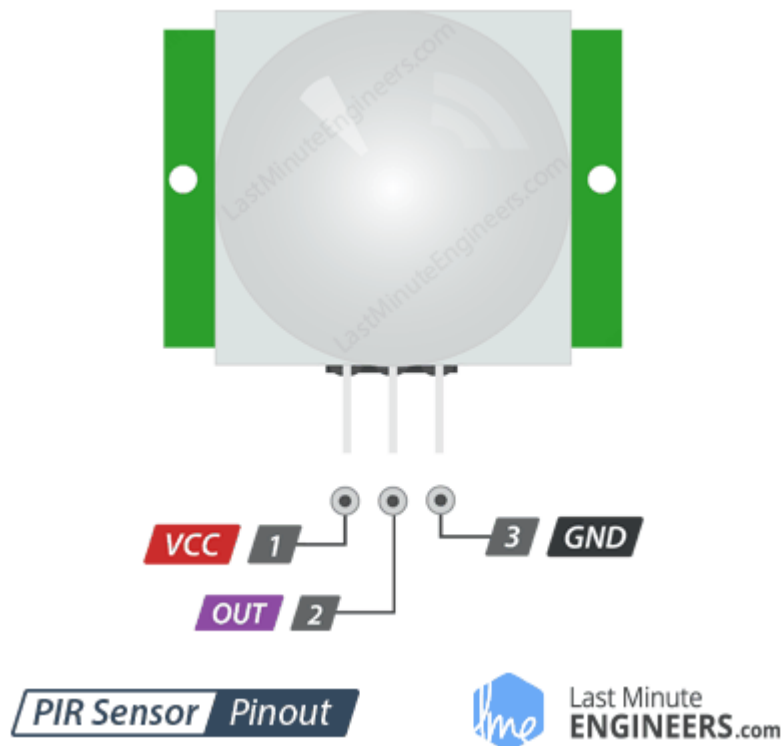
Για την συγκεκριμένη εργασία στις παραπάνω θύρες θα συνδέσουμε τον αισθητήρα κίνησης και πιο συγκεκριμένα στα pin2 (5V Power), pin6 (Ground), pin8 (GPIO14).

Τα τεχνικά χαρακτηριστικά του αισθητήρα HC-SR501 PIR ακολουθούν παρακάτω καθώς ακόμη εμφανίζεται και στις παρακάτω εικόνες [48].

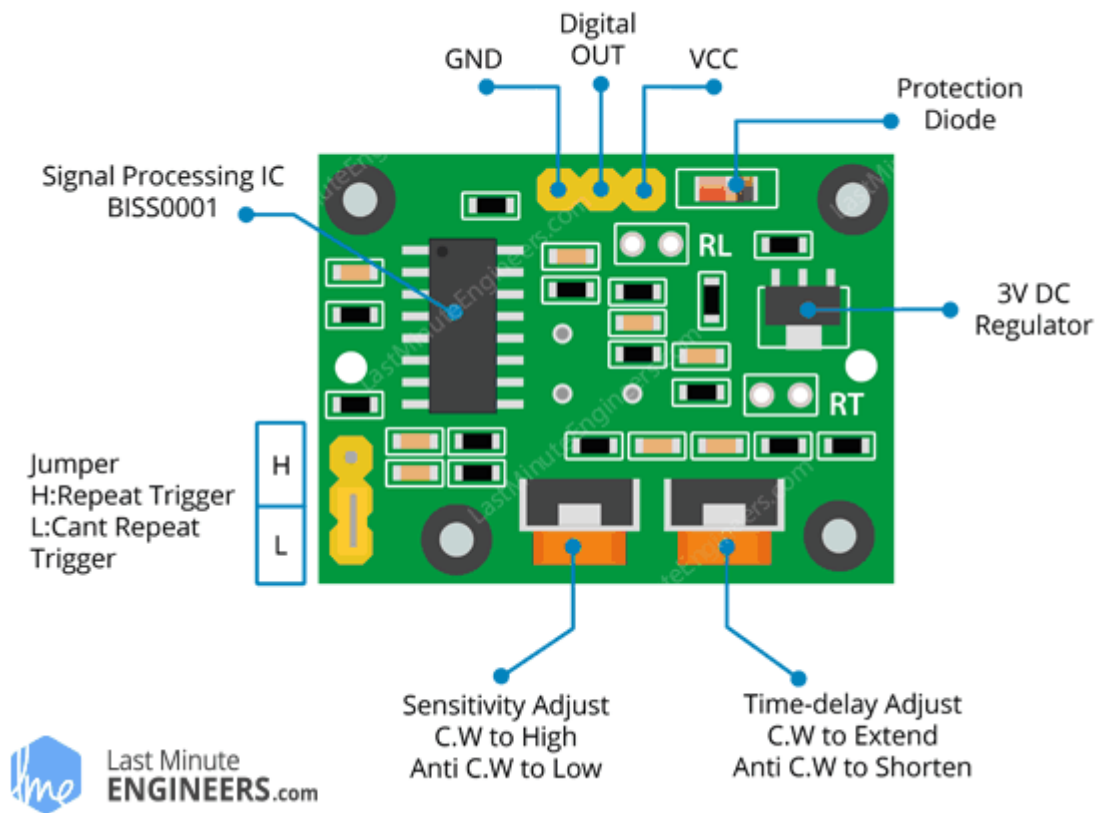
HC-SR501 PIR Sensor - Technical Specifications

- **Product Type:** HC--SR501 Body Sensor Module
- **Input voltage:** DC 4.5-20V
- **Static Current:** <50uA
- **Level output:** High 3.3 V / Low 0V
- **Trigger:** L cannot be repeated trigger / H can be repeated trigger
- **Delay time:** 5-200s
- **Block time:** 2.5s
- **Board Dimensions:** 32mm*24mm
- **Angle Sensor:** <100 cone angle
- **Operation Temp:** -15-+70 degrees
- **Lens size sensor:** Diameter: 23mm

Αναφορικά με τον αισθητήρα μπορούμε να πούμε ότι ο αισθητήρας HC-SR501 PIR ανιχνεύει την κίνηση μετρώντας τις αλλαγές στα επίπεδα υπέρυθρης ακτινοβολίας που εκπέμπονται από τα γύρω αντικείμενα. Όταν ανιχνεύεται κίνηση, ο αισθητήρας PIR εξάγει ένα υψηλό σήμα στον ακροδέκτη εξόδου του. Επίσης ο αισθητήρας PIR HC-SR501 διαθέτει ρυθμιζόμενη καθυστέρηση πριν από την πυροδότηση και ρυθμιζόμενη ευαισθησία. Λειτουργεί με 4,5-20V, η έξοδος ψηφιακού σήματος είναι High 3,3 V / Low 0V. Το εύρος ανίχνευσης είναι περίπου 7 μέτρα (κώνος 100 μοιρών). Στρέφοντας το ποτενσιόμετρο δεξιόστροφα αυξάνεται η εμβέλεια ανίχνευσης (περίπου 7 μέτρα), αντίθετα, η εμβέλεια ανίχνευσης μειώνεται (περίπου 3 μέτρα) [49].



Εικόνα 71 HC-SR501 PIR sensor



Εικόνα 72 HC-SR501PIR sensor

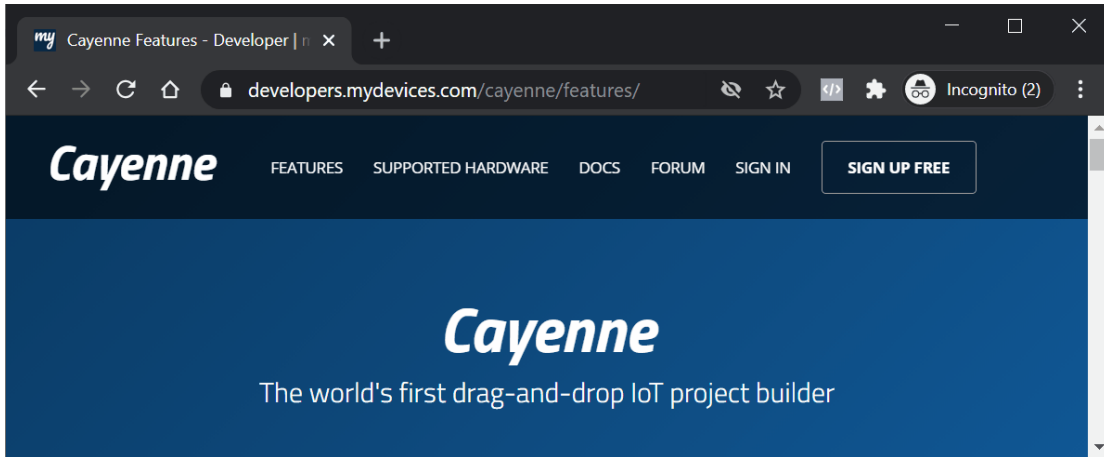
Περιγραφή θέματος

Όπως αναφέρουμε πιο πάνω στη σύνδεση των pin με τους αισθητήρες, θα πρέπει να τονίσουμε ότι για τη διασύνδεση των συσκευών με την πλατφόρμα χρειάζεται να «πούμε» στην πλατφόρμα από ποιες GPIO θα στέλνει και θα λαμβάνει δεδομένα.

Το πρώτο βήμα που οφείλουμε να κάνουμε είναι να συνδέσουμε τη συσκευή μας με την πλατφόρμα(Cayenne my Devices) και έπειτα να λαμβάνουμε και να στέλνουμε τα δεδομένα μας μέσω αυτής στη συσκευή μας καθώς ακόμη και να δημιουργήσουμε κάποια «event» όπου στην πλατφόρμα που χρησιμοποιούμε λέγονται Triggers.

Για τη διασύνδεση της συσκευής μας Raspberry Pi θα πρέπει να ακολουθήσουμε την ακόλουθη διαδικασία.

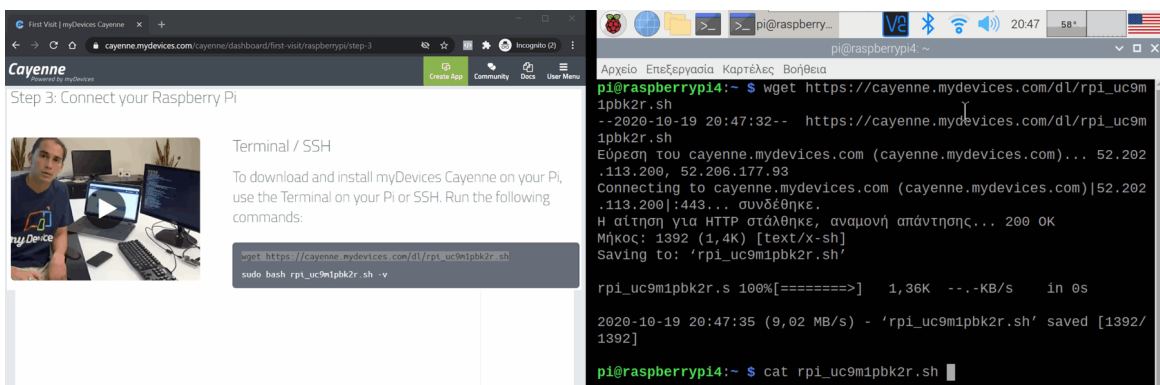
Για να ξεκινήσουμε θα πρέπει αρχικά να δημιουργήσουμε ένα λογαριασμό στο Cayenne myDevices. Η δημιουργία του λογαριασμού δεν έχει κάποιο κόστος. Πατάμε το κουμπί SIGN UP FREE που είναι διαθέσιμο στην πάνω δεξιά μεριά της σελίδας.



Εικόνα 73 Cayenne Features

Καθώς θα ολοκληρώσουμε τη δημιουργία του λογαριασμού στην πλατφόρμα μπορούμε να προχωρήσουμε στο επόμενο στάδιο της διασύνδεσης του Raspberry Pi με την πλατφόρμα μας. Ακόμα και αν δεν γνωρίζουμε πώς γίνεται αυτό, το documentation της Cayenne μας αναφέρει αναλυτικά το πώς γίνεται η παραπάνω διαδικασία. Επιλέγουμε το Raspberry Pi και συνεχίζουμε στο δεύτερο βήμα, όπου μας ρωτάει αν το Raspberry Pi είναι ανοιχτό και συνδεδεμένο στο διαδίκτυο. Αφού βεβαιωθούμε ότι ισχύουν τα ερωτήματα που θέτει, επιλέγουμε το **NEXT**.

Εδώ φτάσαμε στο σημείο, όπου πρέπει να εγκαταστήσουμε τα απαραίτητα εργαλεία στη συσκευή μας.



Εικόνα 74 Cayenne platform - Εγκατάσταση απαραίτητων εργαλείων

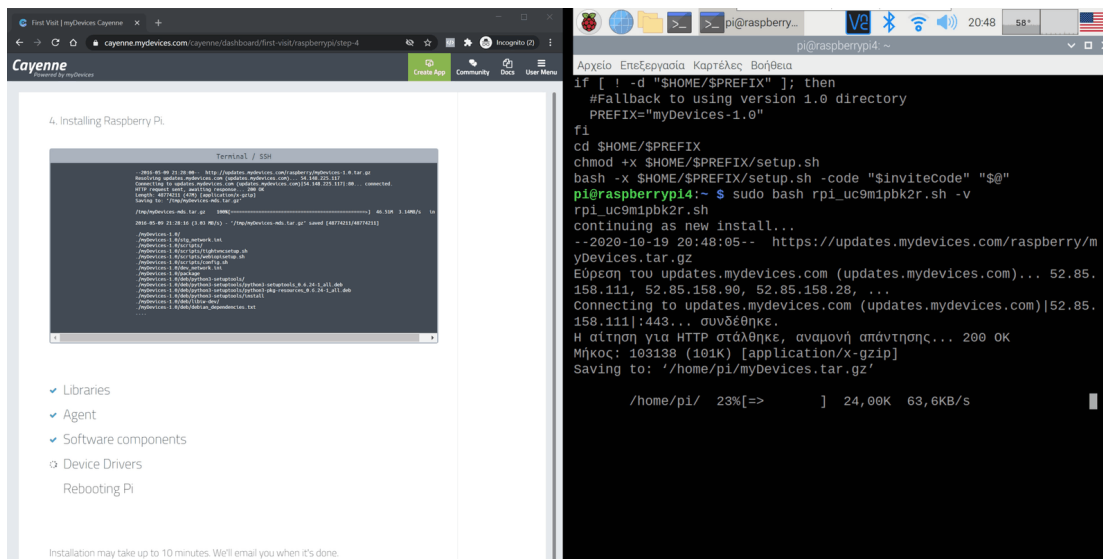
Για να γίνει αυτό μας προτρέπει να τρέξουμε δύο εντολές στο terminal για να γίνει η εγκατάσταση.

wget https://cayenne.mydevices.com/dl/rpi_9u0jxe9cum.sh

Παρατηρούμε ότι η πρώτη εντολή ξεκινά με το **wget**. Το wget είναι μια εφαρμογή που μας βοηθά να κατεβάσουμε ένα αρχείο ή μια σελίδα από το διαδίκτυο, αρκεί να γνωρίζουμε τη διεύθυνση. Αντιγράφουμε την πρώτη εντολή στο τερματικό (terminal). Αυτή η εντολή κατεβάζει ένα αρχείο από τους servers του cayenne.mydevices, με μοναδικό όνομα για κάθε συσκευή που προσθέτουμε σε αυτήν την υπηρεσία. Το αρχείο αυτό έχει κατάληξη **.sh** το όνομα της οποίας προέρχεται από το bash και δηλώνει ότι πρόκειται για αρχείο που περιλαμβάνει εντολές τους συστήματος, αλλά ίσως να περιέχει και εντολές της ενσωματωμένης γλώσσας προγραμματισμού του bash.

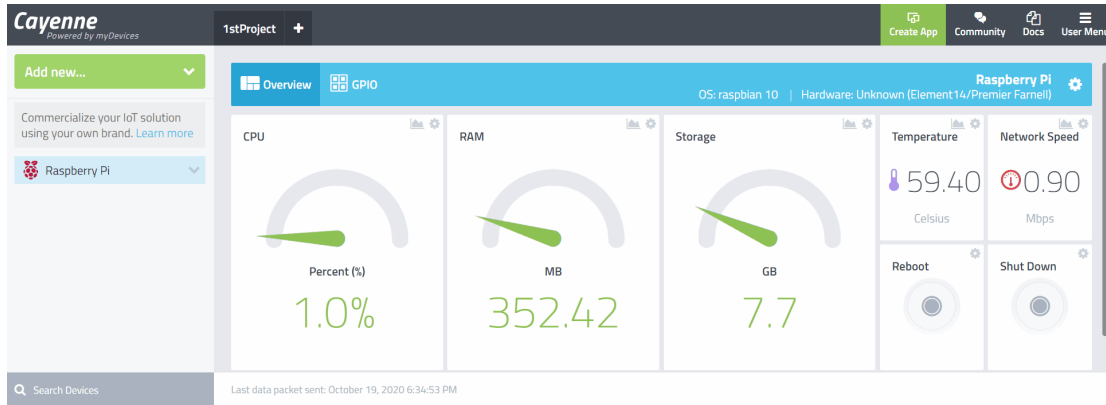
Στο συγκεκριμένο αρχείο περιέχονται όλες οι εντολές που θα πρέπει να εκτελεστούν η μία μετά την άλλη, προκειμένου να εγκατασταθούν στο λειτουργικό σύστημα οι απαραίτητες εφαρμογές για το cayenne myDevices. **Προσοχή:** Πριν ξεκινήσουμε την εγκατάσταση, θα πρέπει να βεβαιωθούμε πως έχουμε αποθηκεύσει τυχόν ανοιχτά έγγραφα στο περιβάλλον του Raspberry Pi OS, διότι κατά τη διάρκεια της εγκατάστασης θα γίνει υποχρεωτική επανεκκίνηση.

Για να ξεκινήσει η διαδικασία εγκατάστασης αντιγράφουμε την επόμενη εντολή ή πληκτρολογούμε **sudo bash rpi_9u0jxe9cum.sh -v** στο τερματικό, η οποία εκτελεί το αρχείο που μόλις κατεβάσαμε.



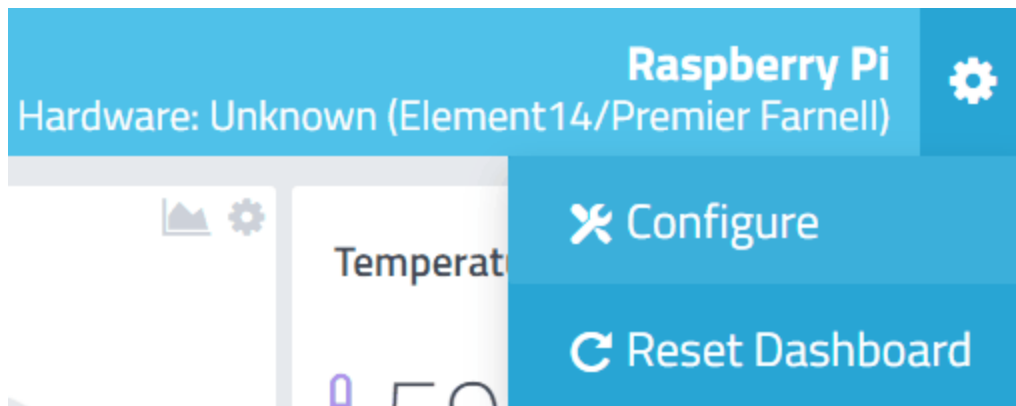
Εικόνα 75 Cayenne platform - Διαδικασία εγκατάστασης

Μπορούμε να παρακολουθήσουμε την πορεία της εγκατάστασης μέσα στο Terminal, αλλά και από το φυλλομετρητή του προσωπικού υπολογιστή μας. Η διαδικασία αυτή μπορεί να διαρκέσει περίπου 10 λεπτά.



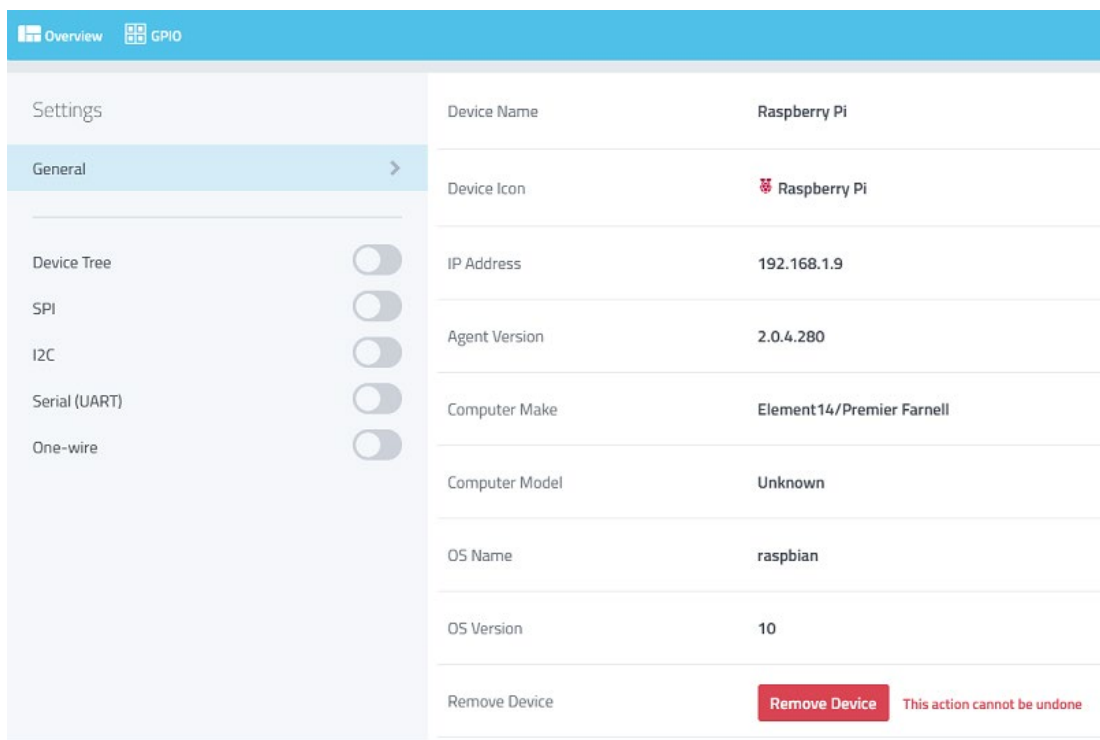
Εικόνα 76 Cayenne platform - Ολοκλήρωση εγκατάστασης

Μόλις ολοκληρωθεί η εγκατάσταση, στο περιβάλλον του Cayenne myDevices εμφανίζονται κάποια στοιχεία που αφορούν την συσκευή μας. Ανάμεσα σε αυτά τα στοιχεία εμφανίζονται οι χρήσεις του επεξεργαστή, της μνήμης RAM, αλλά και της κάρτας SD. Ακόμη θα βρούμε την θερμοκρασία του επεξεργαστή και την ταχύτητα του δικτύου. Έχουμε επίσης την δυνατότητα να κάνουμε επανεκκίνηση της συσκευής μας, ή να την απενεργοποιήσουμε με χρήση των κουμπιών **Reboot** και **Shut Down** αντίστοιχα.



Εικόνα 77 Raspberry Pi Settings

Πάνω δεξιά θα δούμε την ονομασία που έχει δώσει η Cayenne στη συσκευή μας. Δίπλα από το όνομα θα βρούμε ένα γρανάτζι με το οποίο μπορούμε να πλοηγηθούμε στο **Configure**.



Εικόνα 78 Raspberry Pi Settings (2)

Από εκεί αποκτούμε πρόσβαση σε μερικές πληροφορίες για την συσκευή μας, όπως την διεύθυνση ip που έχει λάβει στο δίκτυό μας. Στο πεδίο **Device Name** πατώντας το μολύβι που θα βρούμε δεξιά, μπορούμε να αλλάξουμε το όνομα της συσκευής μας, είναι καλό να το κάνουμε αυτό στην περίπτωση που θελήσουμε να συνδέσουμε και άλλες Raspberry Pi συσκευές στο περιβάλλον του Cayenne. Επίσης αν θέλουμε να διαγράψουμε την συσκευή μας από την Cayenne, είναι πολύ εύκολο να το κάνουμε πατώντας το κουμπί **Remove Device**. Μάλιστα εκτός από την διαγραφή της συσκευής από την Cayenne myDevices, γίνεται αυτόματα απεγκατάσταση των βοηθητικών προγραμμάτων που έχουν εγκατασταθεί στο Raspberry Pi. Πατώντας πάνω στο **Overview** επιστρέφουμε στην βασική σελίδα με τα δεδομένα που συλλέγει η cayenne από τη συσκευή.

Προσθήκη Motion Detection

Κάνουμε κλικ στην επιλογή Add new > Device / Widget στην πλατφόρμα της Cayenne έπειτα από τους Sensors επιλέγουμε Motion

Sensors



Εικόνα 79 Motion detection sensors

Στη συνέχεια Digital Motion Sensor και στο μενού που μας εμφανίζει κάνουμε τις κατάλληλες ρυθμίσεις για τη συσκευή Raspberry Pi 4.

Στο πεδίο Widget Name δίνουμε το όνομα του αισθητήρα, ώστε να ξεχωρίζουμε ποιος αισθητήρας είναι και τί δουλειά κάνει. Στο πεδίο select Service επιλέγουμε την συσκευή Raspberry Pi. Στο πεδίο connectivity επιλέγουμε την επιλογή integrated GPIO, στο επόμενο πεδίο επιλέγουμε το channel 2 μιας και αυτό αντιστοιχεί στο pin8 του αισθητήρα μας και τέλος στο πεδίο choose Widget επιλέγουμε 0/1 2state, 2 καταστάσεων, εάν δηλαδή έχει ανιχνευθεί κίνηση ή όχι.

Enter Settings



Digital Motion Sensor
Motion detector

[View Datasheet](#)

Please fix the following errors:

- Please turn on the device

Widget Name

Digital Motion Sensor

Select Device



Raspberry Pi



Connectivity

Integrated GPIO



Channel



Choose Widget

0/1 2 state



Invert Logic

Add Sensor

Εικόνα 80 Motion detection sensor settings

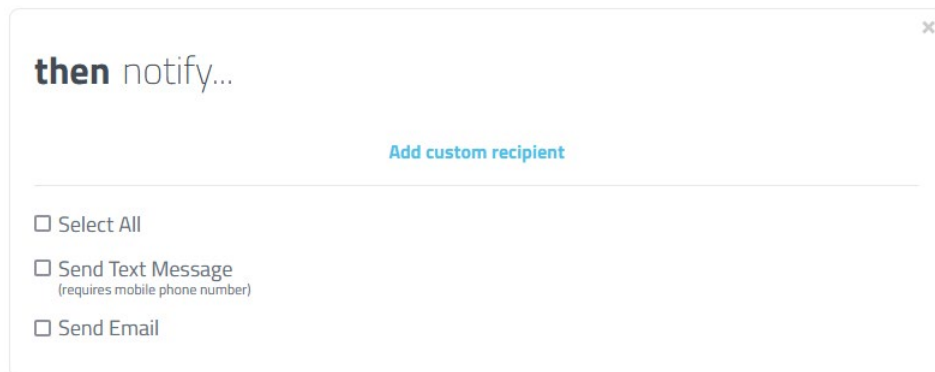
Έπειτα πατάμε Add sensor. Αφού τελειώσουμε με την διαδικασία αυτή θα δούμε ότι ο αισθητήρας μας έχει προστεθεί στη συσκευή Raspberry pi στα αριστερά μας, καθώς και στο main menu της Cayenne.

Τέλος θα προσθέσουμε Trigger στο οποία θα ανιχνεύει την κίνηση και θα μας στέλνει ειδοποίηση μέσω email. Η δημιουργία αυτών των triggers είναι εύκολη μιας και η cayenne μας παρέχει κατευθείαν τις επιλογές αυτές. Να προσθέσουμε ότι στα triggers μπορούμε από επιλογή να έχουμε και ειδοποίηση μέσω μηνύματος sms ή email. Η διαδικασία για την δημιουργία είναι η εξής:

Στην επιλογή Triggers & Alerts επιλέγουμε New Trigger

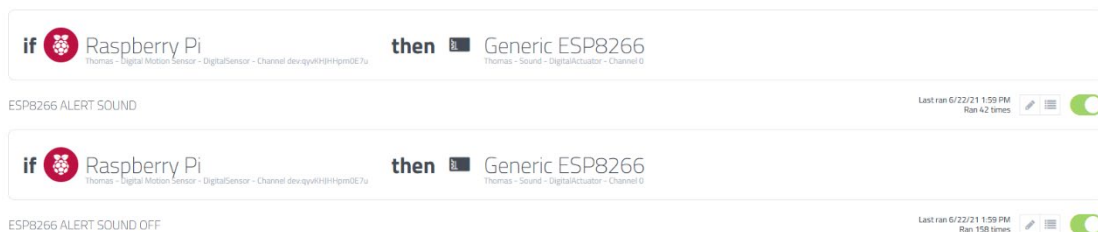
Έπειτα δίνουμε ένα όνομα στο Trigger στο πεδίο Name your trigger για παράδειγμα MOTION ALERT ή οτιδήποτε άλλο θελήσουμε εμείς.

Στο πεδίο **If** «σέρνουμε με (drag and drop)» τη συσκευή που θέλουμε να δημιουργήσουμε το trigger και επιλέγουμε τον αισθητήρα που μας ενδιαφέρει. Δηλαδή Raspberry Pi > Digital Motion Sensor και τικ επιλογή on τώρα στο πεδίο then επιλέγουμε το setup notification , επιλέγουμε το add custom recipient και πληκτρολογούμε διεύθυνση email ή αριθμό τηλεφώνου. Έχουμε τη δυνατότητα να αποστείλουμε και στα 2(email, κινητό) ειδοποιήσεις. Το Trigger αυτό πρέπει να έχει τικ στο on.



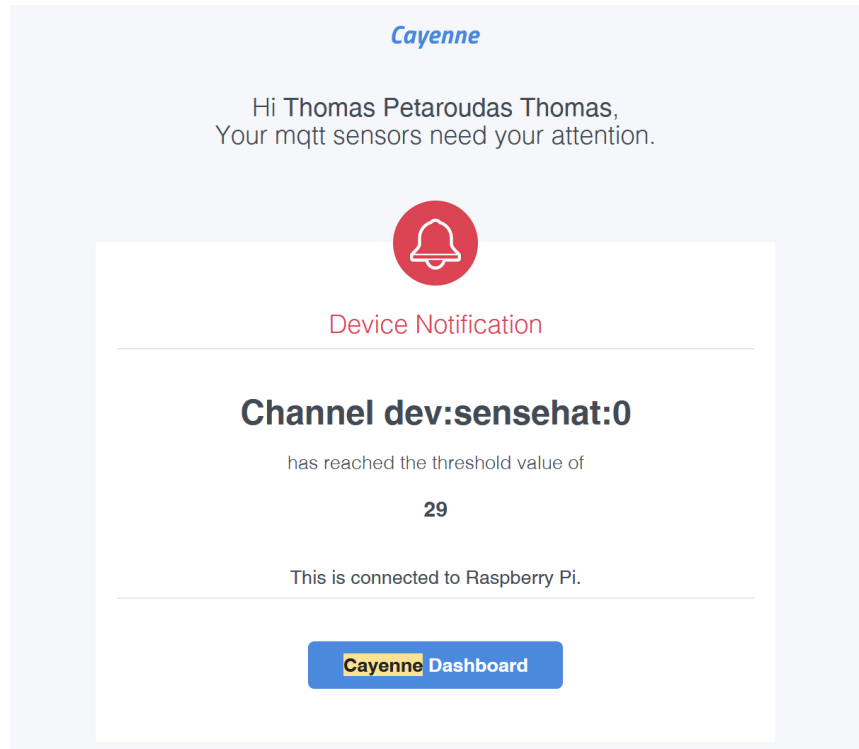
Εικόνα 81 Cayenne platform - Notifications

Επίσης μας δίνεται η δυνατότητα να ανοίγουμε ή να κλείνουμε τα triggers μας από ένα κουμπί που διαθέτει η πλατφόρμα. Μπορούμε δηλαδή να έχουμε μια λίστα από triggers και ανάλογα τις ανάγκες μας να τα διαχειριζόμαστε. Στην εικόνα 82 εμφανίζονται τα triggers μέσα από την πλατφόρμα της Cayenne.



Εικόνα 82 Cayenne platform - Trigger selection

Το αποτέλεσμα της αποστολής του Trigger μέσω email φαίνεται παρακάτω. Η αποστολή της εν λόγω ειδοποίησης γίνεται από την πλατφόρμα της Cayenne και πιο συγκεκριμένα από την εξής διεύθυνση email: no-reply@mydevices.com.



Εικόνα 83 Cayenne platform - email notification