

ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ
ΗΛΕΚΤΡΟΝΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ
ΕΦΑΡΜΟΣΜΕΝΑ ΗΛΕΚΤΡΟΝΙΚΑ ΣΥΣΤΗΜΑΤΑ
ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

«Δημιουργία συστήματος επιτήρησης ωραρίου
εργαζομένων και ελέγχου σωστής λειτουργίας κτιρίου
επιχείρησης με τον STM32»



Του φοιτητή
Ζαγκούλη Νικόλαου
Αρ.Μητρώου: 51904Μ

Επιβλέπων
Άγγελος Γιακουμής
Επίκουρος

Ιούνιος 2022

Δημιουργία συστήματος επιτήρησης ωραρίου εργαζομένων και ελέγχου σωστής λειτουργίας κτιρίου επιχείρησης με τον STM32

Κωδικός Δ.Ε. 21124 Ζαγκούλης Νικόλαος

Άγγελος Γιακουμής

Ημερομηνία ανάληψης Δ.Ε. 8/3/21

Ημερομηνία περάτωσης Δ.Ε. 20/6/21

Βεβαιώνω ότι είμαι ο συγγραφέας αυτής της μεταπτυχιακής εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, έχω καταγράψει τις όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών, εικόνων και κειμένου, είτε αυτές αναφέρονται ακριβώς είτε παραφρασμένες. Επιπλέον, βεβαιώνω ότι αυτή η εργασία προετοιμάστηκε από εμένα προσωπικά, ειδικά ως διπλωματική εργασία, στο Μεταπτυχιακό Πρόγραμμα Σπουδών «Εφαρμοσμένα Ηλεκτρονικά Συστήματα» στο Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του ΔΙ.ΠΑ.Ε.

Η παρούσα εργασία αποτελεί πνευματική ιδιοκτησία του φοιτητή **Ζαγκούλη Νικόλαου** που την εκπόνησε. Στο πλαίσιο της πολιτικής ανοικτής πρόσβασης, ο συγγραφέας/δημιουργός εκχωρεί στο Διεθνές Πανεπιστήμιο της Ελλάδος άδεια χρήσης του δικαιώματος αναπαραγωγής, δανεισμού, παρουσίασης στο κοινό και ψηφιακής διάχυσης της εργασίας διεθνώς, σε ηλεκτρονική μορφή και σε οποιοδήποτε μέσο, για διδακτικούς και ερευνητικούς σκοπούς, άνευ ανταλλάγματος. Η ανοικτή πρόσβαση στο πλήρες κείμενο της εργασίας, δεν σημαίνει καθ' οιονδήποτε τρόπο παραχώρηση δικαιωμάτων διανοητικής ιδιοκτησίας του συγγραφέα/δημιουργού, ούτε επιτρέπει την αναπαραγωγή, αναδημοσίευση, αντιγραφή, πώληση, εμπορική χρήση, διανομή, έκδοση, μεταφόρτωση (downloading), ανάρτηση (uploading), μετάφραση, τροποποίηση με οποιονδήποτε τρόπο, τμηματικά ή περιληπτικά της εργασίας, χωρίς τη ρητή προηγούμενη έγγραφη συναίνεση του συγγραφέα/δημιουργού.

Η έγκριση της διπλωματικής εργασίας από το Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του Διεθνούς Πανεπιστημίου της Ελλάδος, δεν υποδηλώνει απαραίτητως και αποδοχή των απόψεων του συγγραφέα, εκ μέρους του Τμήματος.

«Αφιέρωση»

Στους γονείς μου για την στήριξη όλα αυτά τα χρόνια όσον αφορά τις προπτυχιακές και μεταπτυχιακές σπουδές.

Πρόλογος

Η παρούσα εργασία έχει ως στόχο την δημιουργία ενός συστήματος ελέγχου ωρομέτρησης παρουσιών προσωπικού μίας επιχείρησης, παρέχοντας πληροφορίες στην επιχείρηση όσον αφορά το ωράριο του εργαζομένου (ώρα άφιξης, ώρα αποχώρησης). Με την εκπόνηση διαφόρων έργων τα οποία περιλάμβαναν ενασχόληση με IoT πρωτόκολλα (MQTT, Websockets), PCB design (πλακέτες για έλεγχο αισθητήρων), front-end (σελίδες απεικόνισης δεδομένων), back-end (Python scripts για αλληλεπίδραση βάσης δεδομένων με hardware), το παρών θέμα της εργασίας επιλέχθηκε λόγω του μεγάλου φάσματος εργασιών που πρέπει να γίνουν, με σκοπό το αποτέλεσμα να είναι λειτουργικό.

Περίληψη

Η παρούσα εργασία παρουσιάζει την διαδικασία δημιουργίας ενός τέτοιου συστήματος, καλύπτοντας όλο το φάσμα δημιουργίας του. Πιο συγκεκριμένα, η εργασία αποτελείται από τρία διαφορετικά κομμάτια:

- hardware
- back-end
- front-end

Το κομμάτι του hardware περιλαμβάνει:

- την δημιουργία της πλακέτας με την οποία ο χρήστης σκανάρει το βραχιόλι του
- την δημιουργία της πλακέτας με το οποίο ο χρήστης κάνει εγγραφή του βραχιολιού του στο σύστημα

Το κομμάτι του back-end περιλαμβάνει:

- την βάση δεδομένων για την αποθήκευση πληροφοριών
- τον MQTT broker για την ανταλλαγή μηνυμάτων του συστήματος
- το Python script για την επεξεργασία των μηνυμάτων από τα MQTT topics και αποθήκευσή τους στην βάση δεδομένων

Το κομμάτι του front-end περιλαμβάνει:

- την δημιουργία της ιστοσελίδας του συστήματος στο οποίο παρουσιάζονται οι πληροφορίες των εργαζομένων

«Establishment of a system for monitoring the working hours of employees and controlling the proper operation of a business building with STM32»

«Zagkoulis Nikolaos»

Abstract

This diploma presents the process of implementing an access control system, covering all the appropriate steps that needed in order the system to be finished. More specifically, this diploma consists of 3 parts:

- hardware
- back-end
- front-end

The part of hardware consists of:

- The PCB design of the access control PCB that the employee scans his/her RFID
- The PCB design of register PCB that the employee registers to the access control system

The part of back-end consists of:

- The database for information saving
- The MQTT broker for messages exchanging
- The Python script for messages processing from MQTT topics and saving to the database

The part of front-end consists of:

- The site implementation in which all the information of employees are provided

Ευχαριστίες

Θα ήθελα να πω ένα μεγάλο ευχαριστώ στον άνθρωπο και καθηγητή κ.Αγγελο Γιακουμή για την βοήθεια και τις γνώσεις που μου προσέφερε κατά την διάρκεια της εκπόνησης αυτής της διπλωματικής εργασίας.

Σκοπός της εργασίας

Η εργασία αυτή έχει εφαρμοσμένο χαρακτήρα και σκοπός της είναι η δημιουργία ενός συστήματος ωρομέτρησης παρουσιών εργαζομένων, το οποίο είναι σε θέση να είναι λειτουργικό και ευέλικτο ώστε η εγκατάστασή του να μπορεί να γίνει εύκολα σε οποιαδήποτε επιχείρηση. Για την εγκατάσταση του συστήματος, το μόνο που απαιτείται είναι η εγκατάσταση του τερματικού στην είσοδο του κτιρίου της επιχείρησης καθώς και η εγκατάσταση του software (MQTT,Python script,Database) στον server της επιχείρησης. Από αυτό το σημείο και μετά, το σύστημα είναι λειτουργικό χωρίς περαιτέρω ενέργειες.

Περιεχόμενα

Κατάλογος εικόνων.....	10
Κεφάλαιο 1 ^ο : Access Control	13
1.1 Εισαγωγή	13
1.2 Τι είναι Access Control.....	13
1.3 Είδη και χαρακτηριστικά	13
1.4 Κύρια οφέλη-δυνατότητες	15
1.5 Παρόμοια Συστήματα	15
Κεφάλαιο 2 ^ο : Το πρωτόκολλο MQTT	17
2.1 Εισαγωγή	17
2.2 Χαρακτηριστικά του πρωτοκόλλου MQTT	17
Κεφάλαιο 3 ^ο : Το back-end του συστήματος	18
3.1 Δημιουργία βάσης δεδομένων.....	18
3.2 Δημιουργία συστήματος εγγραφής εργαζομένου στο σύστημα	20
3.3 Δημιουργία Python script.....	20
Κεφάλαιο 4 ^ο : Το front-end του συστήματος.....	22
4.1 Δημιουργία σελίδας εισόδου	22
4.2 Δημιουργία κεντρικής σελίδας.....	24
4.3 Δημιουργία σελίδας μελών της επιχείρησης.....	25
4.4 Δημιουργία σελίδας εγγραφής	26
Κεφάλαιο 5 ^ο : Το hardware του συστήματος.....	28
5.1 Σειριακή Επικοινωνία	28
5.2 Το ESP32	29
5.3 Ο STM32F103C8	30
5.4 Σχεδιασμός της πλακέτας.....	39
Κεφάλαιο 6 ^ο : Ο server του συστήματος.....	44
6.1 Επιλογή Server	44
6.2 Δημιουργία Server	44
Κεφάλαιο 7 ^ο : Μοντέλο του συστήματος	45
Κεφάλαιο 8 ^ο : Μελλοντικά σχέδια-Βελτιώσεις	46
Κεφάλαιο 9 ^ο : Συμπεράσματα.....	46
Κεφάλαιο 10 ^ο : Datasheet	47
10.1 RFID RC522 Module.....	47
10.2 STM32F103C8.....	47
Κώδικες του συστήματος	50
Python Codes	50
Arduino Codes	66

CSS codes	74
Javascript files.....	91
PHP files (files that connect javascript with MYSQL)	117
HTML files	127
STM32 codes	137

Κατάλογος εικόνων

1.5-1 ZK ACC-SKW-H2 Αυτόνομο Σύστημα Ελέγχου Πρόσβασης	15
1.5-2 ZK SC-103 Σύστημα Ωρομέτρησης Προσωπικού	17
1.5-3 ZK TF1700 Βιομετρικό Σύστημα Ωρομέτρησης.....	17
2.1-1 Παράδειγμα απλής επικοινωνίας του Subscriber, του Publisher και του Broker	17
3.1-1 Η καρτέλα «Attendance» της βάσης «rfid_system»	18
3.1-2 Η καρτέλα «Daily Report» της βάσης «rfid_system»	18
3.1-3 Η καρτέλα «History» της βάσης «rfid_system»	19
3.1-4 Η καρτέλα «Members» της βάσης «rfid_system»	19
3.1-5 Η καρτέλα «Users» της βάσης «rfid_system»	19
3.2-1 Το κύκλωμα εγγραφής του συστήματος.....	20
3.2-2 Τα logs του Python Script κατά την εκκίνησή του	21
3.2-3 Το διάγραμμα ροής του Python Script για την επικοινωνία με το ESP32.....	22
3.2-4 Παράδειγμα e-mail της ημερήσιας αναφοράς.....	22
4.1-1 Η σελίδα εισόδου του συστήματος	23
4.1-2 Το διάγραμμα ροής της σελίδας εισόδου	23
4.2-1 Η κεντρική σελίδα του συστήματος.....	24
4.2-2 Το Pop-up Window των εργαζομένων που βρίσκονται εντός της επιχείρησης.....	24
4.3-1 Η σελίδα με τα μέλη της επιχείρησης	25
4.3-2 Παράδειγμα διαγραφής ενός μέλους της επιχείρησης.....	25
4.4-1 Η σελίδα εγγραφής.....	26
4.4-2 Το διάγραμμα ροής της σελίδας εγγραφής.....	27
5.1-1 Τα pins του STM32F103C8 για την σειριακή επικοινωνία	28
5.1-2 Το mode της σειριακής επικοινωνίας.....	28
5.1-3 Το configuration της σειριακής επικοινωνίας.....	29
5.2-1 Παράδειγμα δημιουργίας JSON μηνύματος	29
5.2-2 Διάγραμμα ροής της λειτουργίας του ESP32	30
5.3-1 LCD TFT screen 160x128 with ST7735	30
5.3-2 LCD TFT κατά την εκκίνηση του προγράμματος.....	31
5.3-3 LCD TFT κατά την προσπάθεια εισόδου μη εξουσιοδοτημένου βραχιολιού.....	31
5.3-4 LCD TFT κατά την επιτυχής είσοδο ενός εργαζομένου στο κτίριο.....	32
5.3-5 LCD TFT κατά την έξοδο ενός εργαζομένου από το κτίριο	32
5.3-6 Τα pins του STM32F103C8 για το SPI της LCD οθόνης	33
5.3-7 Το mode του STM32F103C8 για το SPI της οθόνης	33
5.3-8 Το configuration του STM32F103C8 για το SPI της οθόνης	34
5.3-9 RC522 RFID Card Reader Module.....	34
5.3-10 Τα pins του STM32F103C8 για το SPI του RFID	35
5.3-11 Το mode του STM32F103C8 για το SPI του RFID	35
5.3-12 Το configuration του STM32F103C8 για το SPI της RFID	36
5.3-13 Το buzzer του συστήματος	36

5.3-14 Το pin του STM32F103C8 για το buzzer του συστήματος	37
5.3-15 Το pin του STM32F103C8 για το relay του συστήματος.....	37
5.3-16 Το διάγραμμα ροής του προγράμματος του STM32F103C8.....	38
5.4-1 Το σχηματικό της πλακέτας.....	39
5.4-2 Η πλακέτα του συστήματος σε 3D όψη.....	40
5.4-3 Το πάνω μέρος της πλακέτας του συστήματος	41
5.4-4 Το κάτω μέρος της πλακέτας του συστήματος	42
5.4-5 Η πλακέτα του συστήματος	43
6.2-1 Η εφαρμογή MQTT Explorer	44
6.2-2 Η εφαρμογή XAMPP	45
7-1 Το μοντέλο του συστήματος.....	45
10.1-1 RC522 RFID Card Reader Module	47
10.2-1 Η αναπτυξιακή πλακέτα blue pill (STM32F103C8T6).....	48

Συντομογραφίες

RFID Radio-frequency identification

MQTT MQ Telemetry Transport

Json JavaScript Object Notation

IoT Internet of Things

Ετυμολογίες

Τερματικό Η συσκευή εισόδου-εξόδου στην οποία σκανάρει ο εργαζόμενος το βραχιόλι του

Κεφάλαιο 1^ο: Access Control

1.1 Εισαγωγή

Είναι λογικό ότι οι εταιρείες με μεγάλο αριθμό εργαζομένων ψάχνουν λύσεις όσον αφορά τον έλεγχο των ωραρίων του προσωπικού τους χωρίς να ξοδεύουν είτε αρκετό χρόνο είτε επιπλέον ανθρώπινο δυναμικό. Γι' αυτόν τον λόγο, όλο και περισσότερες εταιρείες στρέφονται προς τα Access Control συστήματα, τα οποία με μία απλή κάρτα (την οποία λαμβάνει ο κάθε εργαζόμενος) προσφέρουν αρκετές πληροφορίες για κάθε εργαζόμενο.

1.2 Τι είναι Access Control

Τα access control συστήματα αποτελούν αναπόσπαστο κομμάτι της σύγχρονης βιομηχανίας της ασφάλειας. Πρόκειται για συστήματα που ελέγχουν την πρόσβαση σε συγκεκριμένο χώρο (είτε πρόκειται για επιχείρηση είτε για προσωπική κατοικία) με την χρήση της να αποτελεί όλο και πιο αναγκαία με την πάροδο του χρόνου.

Άξιο σημείο αναφοράς είναι το γεγονός ότι η αγορά του ελέγχου πρόσβασης αξιολογήθηκε στα 7,65 δις δολάρια το 2020, ενώ οι προβλέψεις κάνουν λόγο για άνοδο του ποσού στα 11,7 δολάρια. Επίσης, ο μέσος ετήσιος ρυθμός ανάπτυξης ανέρχεται στο 10.6%, σύμφωνα με έρευνα της Markets and Markets.

Με τον όρο Access Control αναφερόμαστε στον έλεγχο πρόσβασης, μία τεχνική ασφαλείας που καθορίζει για το ποιος εισέρχεται σε έναν συγκεκριμένο χώρο. Επιπλέον, ένα Access Control σύστημα μπορεί ακόμα και να περιορίζει την πρόσβαση σε ένα κτίριο ή σε ένα δωμάτιο.

Ένα παράδειγμα ελέγχου εισόδου σε ένα κτίριο είναι η πόρτα εισόδου να χρειάζεται σωστό κωδικό για να ανοίξει. Με την πληκτρολόγηση του σωστού κωδικού, η πόρτα ανοίγει αυτόματα μέσω της σύνδεσης της μαγνητικής κλειδαριάς της πόρτας με το Access Control σύστημα. Αντίθετα, με την πληκτρολόγηση λανθασμένου κωδικού, η πρόσβαση στο κτίριο δεν είναι εφικτή, με αποτέλεσμα την ειδοποίηση προσωπικού ασφαλείας για μη εξουσιοδοτημένη απόπειρα εισόδου.

Η συγκεκριμένη τεχνολογία χρησιμοποιείται πιο συχνά σε επιχειρήσεις και όχι τόσο σε κατοικίες. Τα συστήματα ελέγχου πρόσβασης αποτελούνται κυρίως από:

- Κάρτες πρόσβασης
- Καρταναγνώστες
- Keypads
- Ηλεκτρομαγνητικές κλειδαριές
- Σύστημα συναγερμού
- Πίνακες
- Λογισμικό

1.3 Είδη και χαρακτηριστικά

Τα συστήματα ελέγχου πρόσβασης χωρίζονται σε δύο κατηγορίες:

- Φυσικά συστήματα (physical)
- Λογικά συστήματα (logical)

Οι δύο παραπάνω κατηγορίες σχετίζονται με το ποιος μπορεί να έχει πρόσβαση σε μία περιοχή.

Το logical system ελέγχει την εικονική πρόσβαση σε δεδομένα, ψηφιακούς τρόπους και σε δίκτυα. Ο έλεγχος ταυτότητας για να εξουσιοδοτηθεί η πρόσβαση πραγματοποιείται με ψηφιακές τεχνολογίες.

Το physical system ελέγχει την είσοδο ή έξοδο ανθρώπων σε ένα συγκεκριμένο χώρο. Τα μέσα που χρησιμοποιούνται είναι κυρίως μαγνητικές κλειδαριές, κάμερες, μπάρες.

Άλλες μορφές ελέγχου πρόσβασης είναι οι εξής:

- Mandatory (MAC)
- Discretionary (DAC)
- Role-based (RBAC)

Ο MAC έλεγχος αφορά κυρίως κυβερνητικές ή στρατιωτικές τοποθεσίες, διότι απαιτεί συνεχή παρακολούθηση. Η πρόσβαση ταξινομείται σε κλάσεις και επίπεδα άδειας. Όταν κάποιος ζητήσει άδεια πρόσβασης, το σύστημα ελέγχει την ταξινόμησή του για να δει αν μπορεί να του επιτρέψει να περάσει.

Ο DAC έλεγχος έχει την ευελιξία οι ιδιοκτήτες να αποφασίζουν τις πολιτικές εισόδου και το ποιος είναι εξουσιοδοτημένος ή όχι.

Με τον RBAC έλεγχο, ο κάθε εργαζόμενος έχει πρόσβαση μόνο σε όσα αφορούν την εργασία του. Μ' αυτόν τον τρόπο, έχει συγκεκριμένο ρόλο και μπορεί να κινηθεί μόνο στους χώρους στους οποίους έχει εξουσιοδότηση. Ο έλεγχος αυτός αφορά κυρίως μεγάλες επιχειρήσεις με ξεχωριστούς χώρους εργασίας.

Στην αγορά υπάρχουν ακόμα δύο είδη ελέγχου πρόσβασης. Το Ruler-based όπου ο διαχειριστής του συστήματος ελέγχει την πρόσβαση σε «αντικείμενα», όπως η ώρα της ημέρας και χρησιμοποιείται σε συνδυασμό με κάποιο από τα παραπάνω. Επίσης γνωστό είναι το Attribute-based, στο οποίο συνδυάζονται χρήστες, συστήματα και περιβαλλοντικές συνθήκες, δημιουργώντας ένα πιο πολύπλευρο σύνολο κανόνων.

Η κυρίαρχη μορφή της επαλήθευσης ταυτότητας είναι η πλαστική κάρτα πρόσβασης. Αν και εξακολουθεί να χρησιμοποιείται στα περισσότερα εγκατεστημένα συστήματα, οι βιομετρικές λύσεις (επικύρωση ταυτότητας μέσω αναγνώρισης προσώπου, δακτυλικών αποτυπωμάτων και αναγνώρισης της ίριδας) γίνονται όλο και πιο αξιόπιστα, οικονομικά προσιτά και περισσότερο διαδεδομένες.

Οι έλεγχοι πρόσβασης απευθύνονται σε κάθε χώρο με αυξημένες απαιτήσεις ασφαλείας και μπορούν να αποβούν εξαιρετικά χρήσιμοι. Οι χώροι που χρησιμοποιούνται κυρίως είναι:

- Νοσοκομεία και ιατρικά κέντρα
- Δημόσιες υπηρεσίες
- Εμπορικά κέντρα ή μεγάλα καταστήματα
- Όμιλοι επιχειρήσεων
- Εκπαιδευτικά ιδρύματα
- Πολυκατοικίες
- Αθλητικοί χώροι
- Χώροι ψυχαγωγίας
- Ξενοδοχεία
- Βιομηχανικές εγκαταστάσεις

1.4 Κύρια οφέλη-δυνατότητες

Το μεγαλύτερο όφελος των συστημάτων ελέγχου πρόσβασης είναι η αποτροπή εγκληματικών δραστηριοτήτων και μη εξουσιοδοτημένων εισόδων και εξόδων. Σημαντικά οφέλη επίσης είναι:

- Ασφαλή και άμεση πληροφόρηση μέσω σύγχρονου λογισμικού
- Εξοικονόμηση χρόνου και χρήματος
- Εύκολη διαχείριση της παρουσίας εργαζομένων της επιχείρησης
- Δυνατότητα πλήρους διαχείρισης αδειών εργαζομένων
- Ολοκληρωμένη διαχείριση μισθοδοσίας εργαζομένων με δυνατότητες παραμετροποίησης
- Χαμηλά έξοδα προμήθειας και εγκατάστασης

Οι δυνατότητες των συστημάτων οι οποίες προσφέρουν, είναι:

- Αναγνώριση μέσω δακτυλικών αποτυπωμάτων
- Αναγνώριση του προσώπου με ασύρματες κάμερες
- Αναγνώριση ματιών και ίριδας με κάμερες ασφαλείας
- Ορισμό κωδικού PIN μέσω του πληκτρολογίου
- Αναγνώριση με χρήση επαγωγικής-μαγνητικής κάρτας RFID
- Βιομετρικά συστήματα

1.5 Παρόμοια Συστήματα

Τα συστήματα τα οποία υπάρχουν ήδη στην αγορά ποικίλλουν ανάλογα με τις υπηρεσίες που προσφέρουν. Αυτό σημαίνει ότι τέτοιου είδους συστήματα μπορούν να περιλαμβάνουν ένα απλό τερματικό στο οποίο είναι αποθηκευμένες οι κάρτες RFID (οι οποίες είναι εξουσιοδοτημένες για είσοδο), με τους εργαζομένους να σκανάρουν την κάρτα τους με μόνο σκοπό την είσοδό τους στο κτίριο, είτε φτάνοντας σε πλήρη συστήματα ελέγχου ωρομέτρησης εργαζομένων, με την είσοδο στο κτίριο να πραγματοποιείται με αναγνώριση ματιών και ίριδας με κάμερες ασφαλείας.

Αναφορικά με το κόστος των συγκεκριμένων συστημάτων, παρουσιάζεται ενδεικτικά τιμές τερματικών από την εταιρεία NOVATRON.

Ένα αυτόνομο σύστημα ελέγχου με πληκτρολόγιο και επαγωγικές κάρτες για έως και 1000 χρήστες και έξοδος για μαγνητική κλειδαριά κοστολογείται στα 165€.



Εικόνα 1.5-1 ZK ACC-SKW-H2 Αυτόνομο Σύστημα Ελέγχου Πρόσβασης

Το σύστημα ωρομέτρησης προσωπικού SC-103 διαβάζει επαγωγικές (proximity) κάρτες προσωπικού και μπορεί να λειτουργήσει είτε ανεξάρτητα, είτε δικτυακά καταγράφοντας τις κινήσεις σε πραγματικό χρόνο. Στο κάθε τερματικό μπορούν να καταχωρηθούν 30.000 κάρτες και να αποθηκευθούν έως 50.000 εγγραφές. Το κόστος ανέρχεται στα 560€.



Εικόνα 1.5-2 ZK SC-103 Σύστημα Ωρομέτρησης Προσωπικού

Το βιομετρικό σύστημα ωρομέτρησης και ελέγχου πρόσβασης εργαζομένων TF1700 (κόστος 890€) «διαβάζει» δακτυλικά αποτυπώματα καθώς και επαγωγικές κάρτες προσωπικού. Μπορεί να λειτουργήσει είτε ανεξάρτητα, είτε δικτυακά καταγράφοντας τις κινήσεις σε πραγματικό χρόνο. Στο κάθε τερματικό μπορούν να καταχωρηθούν 10.000 κάρτες, 1.500 δακτυλικά αποτυπώματα και να αποθηκευθούν έως 30.000 εγγραφές. Η αναγνώριση μπορεί να γίνει με 3 τρόπους:

- Δακτυλικό αποτύπωμα
- Επαγωγική κάρτα
- Συνδυασμός και των 2 για μέγιστη ασφάλεια



Εικόνα 1.5-3 ZK TF1700 Βιομετρικό Σύστημα Ωρομέτρησης

Κεφάλαιο 2^ο: Το πρωτόκολλο MQTT

2.1 Εισαγωγή

MQTT ή Queue Telemetry Transport είναι ένα ελαφρύ συμπαγές πρωτόκολλο ανταλλαγής μηνυμάτων για την μεταφορά δεδομένων σε απομακρυσμένες τοποθεσίες όπου απαιτείται "αποτύπωμα μικρού κώδικα" ή το εύρος ζώνης του δικτύου είναι περιορισμένο. Αυτά τα πλεονεκτήματα επιτρέπουν την εφαρμογή αυτού του πρωτοκόλλου στα συστήματα M2M (Machine to Machine) και IoT (Industrial Internet of Things). Υπάρχει επίσης μια παραλλαγή του πρωτοκόλλου MQTT-SN (MQTT για τα δίκτυα αισθητήρων), γνωστή ως MQTT-S, η οποία έχει σχεδιαστεί για τις ενσωματωμένες ασύρματες συσκευές χωρίς υποστήριξη δικτύων TCP / IP, για παράδειγμα ZigBee.

2.2 Χαρακτηριστικά του πρωτοκόλλου MQTT

Βασικά χαρακτηριστικά του πρωτοκόλλου MQTT:

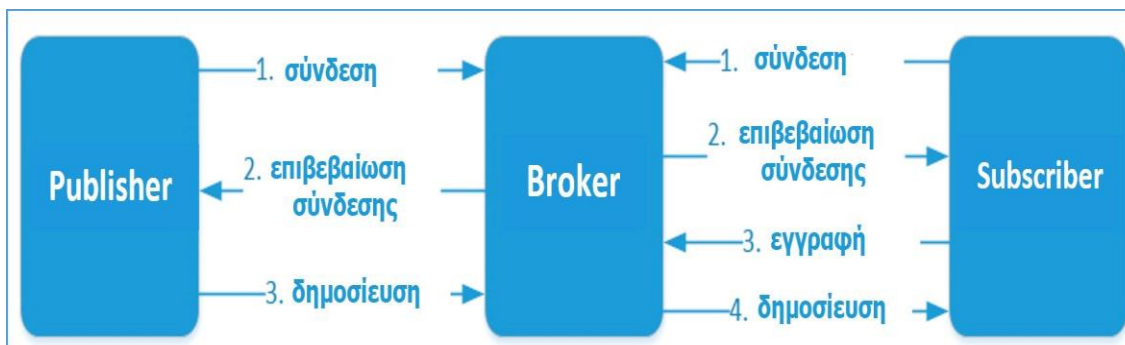
- Ασύγχρονο πρωτόκολλο
- Συμπαγή μηνύματα
- Λειτουργία σε συνθήκες ασταθούς σύνδεσης της γραμμής μετάδοσης των δεδομένων
- Υποστήριξη διαφόρων επιπέδων ποιότητας υπηρεσιών (QoS)
- Εύκολη ενσωμάτωση νέων συσκευών

Στο πρωτόκολλο MQTT, η ανταλλαγή των μηνυμάτων πραγματοποιείται μεταξύ του Client, που μπορεί να είναι Publisher ή Subscriber των μηνυμάτων, και Broker των μηνυμάτων (π.χ. Mosquitto MQTT).

Ο Publisher στέλνει τα δεδομένα σε Broker MQTT ορίζοντας ένα συγκεκριμένο θέμα στο μήνυμα. Οι Subscribers μπορούν να λαμβάνουν διάφορα δεδομένα από πολλούς Publishers, ανάλογα με τη συνδρομή σε αντίστοιχα θέματα.

Οι συσκευές MQTT χρησιμοποιούν ορισμένους τύπους μηνυμάτων για την επικοινωνία τους με τον Broker. Εδώ είναι οι βασικοί τύποι:

- Connect - δημιουργία σύνδεσης με το Broker μηνυμάτων
- Disconnect - διακοπή σύνδεσης με το Broker μηνυμάτων
- Publish - δημοσίευση δεδομένων σχετικά με κάποιο θέμα μέσα στο Broker μηνυμάτων
- Subscribe - εγγραφή σε ένα θέμα στο Broker μηνυμάτων
- Unsubscribe - διαγραφή του θέματος



2.1-1 Παράδειγμα απλής επικοινωνίας του Subscriber, του Publisher και του Broker

Κεφάλαιο 3^ο: Το back-end του συστήματος

3.1 Δημιουργία βάσης δεδομένων

Ένα από τα κυριότερα κομμάτια του συστήματος είναι η βάση δεδομένων. Η βάση αυτή πρέπει να κρατάει τα δεδομένα των εργαζομένων με σκοπό την παρουσίασή τους μέσω της ιστοσελίδας του συστήματος.

Η βάση «rfid_system» που δημιουργήθηκε, περιέχει τις εξής στήλες που περιγράφονται παρακάτω:

- Attendance
- Daily Report
- History
- Members
- Users

Η καρτέλα «Attendance» περιέχει:

- CardID (το ID του βραχιολιού)
- Arrival (η ώρα άφιξης του εργαζομένου)

#	Όνομα	Τύπος	Σύνθεση	Χαρακτηριστικά	Κενό	Προεπιλογή	Σχόλια	Πρόσθετα	Ενέργεια
<input type="checkbox"/>	1	cardID	varchar(255)	latin1_swedish_ci	Όχι	Καμία			Αλλαγή Διαγραφή Περισσότερα
<input type="checkbox"/>	2	arrival	datetime		Όχι	Καμία			Αλλαγή Διαγραφή Περισσότερα

3.1-1 Η καρτέλα «Attendance» της βάσης «rfid_system»

Η καρτέλα «Daily Report» περιέχει:

- CardID (το ID του βραχιολιού)
- Arrival (ώρα εισόδου του εργαζομένου)
- Exodus (ώρα εξόδου του εργαζομένου)
- ComeLate (την διαφορά της ώρας εισόδου με την κανονική ώρα έναρξης του ωαρίου εργασίας)
- LeftEarly (την διαφορά της ώρας εξόδου με την κανονική ώρα λήξης του ωαρίου εργασίας)
- Overtime (την διαφορά μεταξύ των ωρών εργασίας του εργαζομένου με αυτές του κανονικού ωαρίου εργασίας)

#	Όνομα	Τύπος	Σύνθεση	Χαρακτηριστικά	Κενό	Προεπιλογή	Σχόλια	Πρόσθετα	Ενέργεια
<input type="checkbox"/>	1	cardID	varchar(255)	latin1_swedish_ci	Όχι	Καμία			Αλλαγή Διαγραφή Περισσότερα
<input type="checkbox"/>	2	arrival	datetime(6)		Όχι	Καμία			Αλλαγή Διαγραφή Περισσότερα
<input type="checkbox"/>	3	exodus	datetime(6)		Όχι	Καμία			Αλλαγή Διαγραφή Περισσότερα
<input type="checkbox"/>	4	comeLate	int(255)		Όχι	Καμία			Αλλαγή Διαγραφή Περισσότερα
<input type="checkbox"/>	5	leftEarly	int(255)		Όχι	Καμία			Αλλαγή Διαγραφή Περισσότερα
<input type="checkbox"/>	6	hoursWorked	time		Όχι	Καμία			Αλλαγή Διαγραφή Περισσότερα
<input type="checkbox"/>	7	overtime	int(255)		Όχι	Καμία			Αλλαγή Διαγραφή Περισσότερα

3.1-2 Η καρτέλα «Daily Report» της βάσης «rfid_system»

Στην καρτέλα «History» βρίσκονται οι καταχωρήσεις που μεταφέρονται από την καρτέλα «Daily Report» μετά το πέρας της ημέρας.

Η μόνη διαφορά μεταξύ τους είναι ένα επιπλέον μέλος ID στην καρτέλα «History», με το οποίο διευκολύνεται η διαγραφή μιας καταχώρησης.

#	Όνομα	Τύπος	Σύνθεση	Χαρακτηριστικά	Κενό	Προεπιλογή	Σχόλια	Πρόσθετα	Ενέργεια
<input type="checkbox"/>	1	id	int(11)		Όχι	Καμία		AUTO_INCREMENT	Αλλαγή Διαγραφή Περισσότερα
<input type="checkbox"/>	2	cardID	varchar(255) latin1_swedish_ci		Όχι	Καμία			Αλλαγή Διαγραφή Περισσότερα
<input type="checkbox"/>	3	arrival	datetime		Όχι	Καμία			Αλλαγή Διαγραφή Περισσότερα
<input type="checkbox"/>	4	exodus	datetime		Όχι	Καμία			Αλλαγή Διαγραφή Περισσότερα
<input type="checkbox"/>	5	comeLate	int(255)		Όχι	Καμία			Αλλαγή Διαγραφή Περισσότερα
<input type="checkbox"/>	6	leftEarly	int(255)		Όχι	Καμία			Αλλαγή Διαγραφή Περισσότερα
<input type="checkbox"/>	7	hoursWorked	time(6)		Όχι	Καμία			Αλλαγή Διαγραφή Περισσότερα
<input type="checkbox"/>	8	overtime	int(255)		Όχι	Καμία			Αλλαγή Διαγραφή Περισσότερα

3.1-3 Η καρτέλα «History» της βάσης «rfid_system»

Η καρτέλα «Members» περιέχει τα μέλη της επιχείρησης με τα εξής στοιχεία:

- ID (auto increment ID)
- Name (όνομα εργαζομένου)
- Surname (επίθετο εργαζομένου)
- Job (είδος εργασίας στην επιχείρηση)
- E-mail (ηλεκτρονικό ταχυδρομείο)
- Password (προσωπικός κωδικός)
- CardID (το ID του RFID)
- Picture (το όνομα αρχείου της φωτογραφίας του μέλους)

#	Όνομα	Τύπος	Σύνθεση	Χαρακτηριστικά	Κενό	Προεπιλογή	Σχόλια	Πρόσθετα	Ενέργεια
<input type="checkbox"/>	1	id	int(255)		Όχι	Καμία		AUTO_INCREMENT	Αλλαγή Διαγραφή Περισσότερα
<input type="checkbox"/>	2	name	varchar(255) latin1_swedish_ci		Όχι	Καμία			Αλλαγή Διαγραφή Περισσότερα
<input type="checkbox"/>	3	surname	varchar(255) latin1_swedish_ci		Όχι	Καμία			Αλλαγή Διαγραφή Περισσότερα
<input type="checkbox"/>	4	job	varchar(255) latin1_swedish_ci		Όχι	Καμία			Αλλαγή Διαγραφή Περισσότερα
<input type="checkbox"/>	5	email	varchar(255) latin1_swedish_ci		Όχι	Καμία			Αλλαγή Διαγραφή Περισσότερα
<input type="checkbox"/>	6	password	varchar(255) latin1_swedish_ci		Όχι	Καμία			Αλλαγή Διαγραφή Περισσότερα
<input type="checkbox"/>	7	cardID	varchar(255) latin1_swedish_ci		Όχι	Καμία			Αλλαγή Διαγραφή Περισσότερα
<input type="checkbox"/>	8	picture	varchar(255) latin1_swedish_ci		Όχι	Καμία			Αλλαγή Διαγραφή Περισσότερα

3.1-4 Η καρτέλα «Members» της βάσης «rfid_system»

Στην καρτέλα «Users» καταχωρούνται «usernames» και «passwords» τα οποία έχουν πρόσβαση στην ιστοσελίδα διαχείρισης των εργαζομένων.

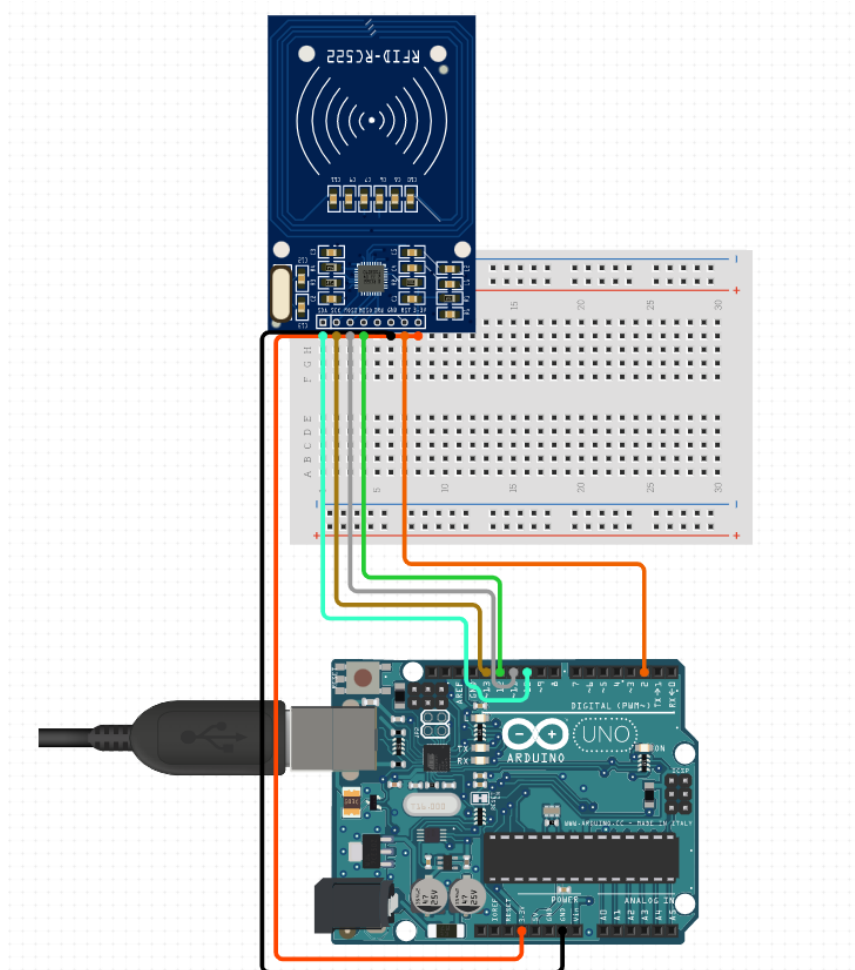
#	Όνομα	Τύπος	Σύνθεση	Χαρακτηριστικά	Κενό	Προεπιλογή	Σχόλια	Πρόσθετα	Ενέργεια
<input type="checkbox"/>	1	username	varchar(255) latin1_swedish_ci		Όχι	Καμία			Αλλαγή Διαγραφή Περισσότερα
<input type="checkbox"/>	2	password	varchar(255) latin1_swedish_ci		Όχι	Καμία			Αλλαγή Διαγραφή Περισσότερα

3.1-5 Η καρτέλα «Users» της βάσης «rfid_system»

3.2 Δημιουργία συστήματος εγγραφής εργαζομένου στο σύστημα

Την πρώτη φορά που ο εργαζόμενος κάνει εγγραφή στο σύστημα, πρέπει να ηλεκτρολογήσει τα στοιχεία του και να σκανάρει το βραχιόλι το οποίο θα χρησιμοποιεί για την είσοδο και έξοδο από το κτίριο.

Γι' αυτό τον λόγο, δημιουργήθηκε ένα κύκλωμα για την εγγραφή του χρήστη. Το κύκλωμα μπορεί να συνδεθεί σε έναν υπολογιστή με πρόσβαση στο τοπικό δίκτυο της επιχείρησης, με σκοπό την αποστολή του ID του βραχιολιού στην ιστοσελίδα του συστήματος για την αποθήκευσή του στην βάση μαζί με τα στοιχεία του κατόχου του.



3.2-1 Το κύκλωμα εγγραφής του συστήματος

Το σύστημα εγγραφής αποτελείται από ένα Arduino Uno και ένα RFID RC522 συνδεδεμένα με το πρωτόκολλο SPI. Κάθε φορά που κάποιος βραχιόλι βρίσκεται στην εμβέλεια του RFID, το ID του βραχιολιού αποστέλλεται στην θύρα του υπολογιστή με σκοπό την επεξεργασία του από το Python Script που εκτελείται στον υπολογιστή (βλ. Δημιουργία Python Script).

3.3 Δημιουργία Python script

Για να είναι σε θέση το σύστημα να δουλεύει κατάλληλα, κρίθηκε αναγκαία η δημιουργία ενός Python Script, το οποίο είναι υπεύθυνο για:

- Έλεγχο των μηνυμάτων που στέλνονται στο MQTT topic «DoorRfid» από το ESP32 και αποστολή στην βάση με την κατάλληλη επεξεργασία
- Έλεγχο της σειριακής θύρας στην οποία είναι συνδεδεμένη η συσκευή εγγραφής RFID και αποστολή του κατάλληλου μηνύματος στο MQTT Topic «RfidScanner»

- Επεξεργασία των εγγραφών της βάσης και αποστολή ημερήσιας αναφοράς εισόδου-εξόδου εργαζομένων σε e-mail

Ο ρόλος του Python Script (όσον αφορά το σύστημα εγγραφής) είναι να λαμβάνει τα ID των βραχιολιών σειριακά μέσω του Arduino Uno και να τα στέλνει σε ένα MQTT topic στο οποίο η ιστοσελίδα έχει κάνει Subscribe. Μ'αυτόν τον τρόπο, το ID του βραχιολιού καταχωρείται στην βάση μαζί με τα στοιχεία του εργαζομένου.

Με την εκκίνηση του Python Script, γίνεται αναζήτηση αν υπάρχει κάποιο συνδεδεμένο Arduino στις θύρες της συσκευής που τρέχει το script, με σκοπό να ξεκινήσει η σειριακή επικοινωνία. Εφόσον η επικοινωνία ξεκινήσει, κάθε μήνυμα που έρχεται μέσω του Arduino, στέλνεται στο MQTT topic «Rfid Scanner» σε μορφή Json. Σε περίπτωση διακοπής της σειριακής επικοινωνίας, γίνεται προσπάθεια επανασύνδεσης.

```

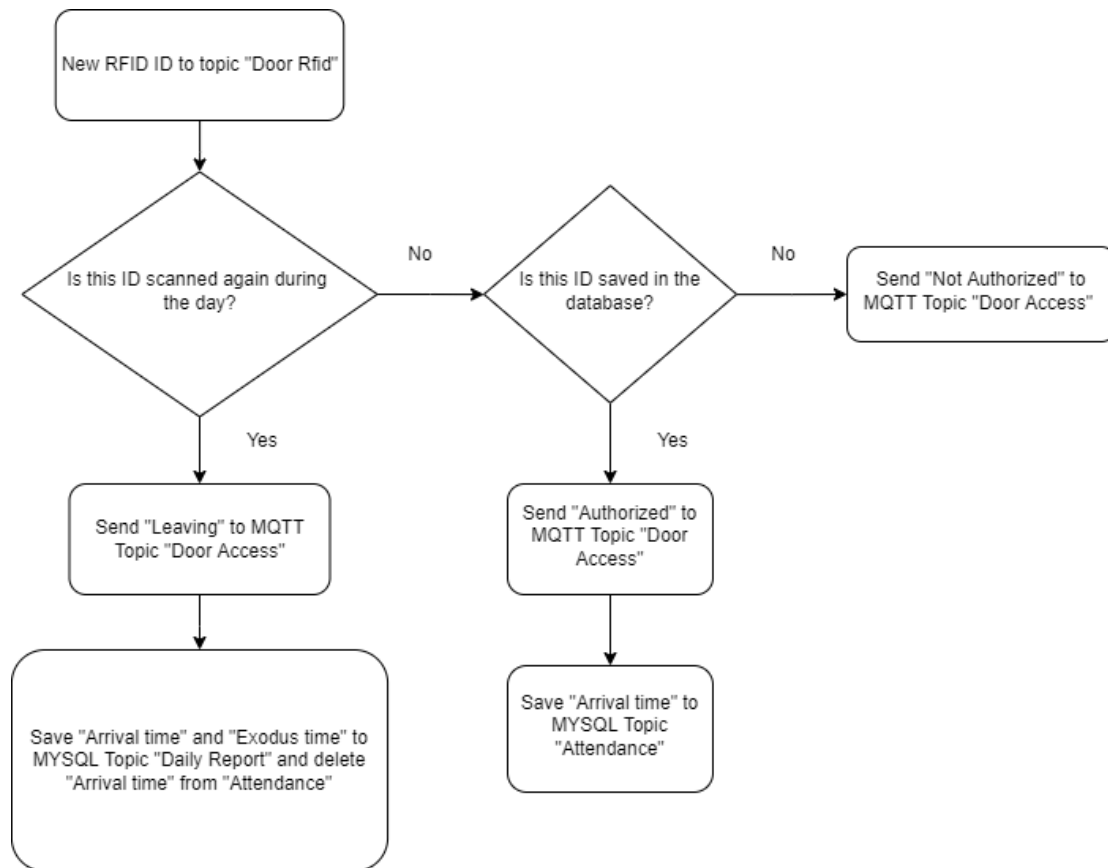
Creating new instance
Connecting to broker:192.168.100.4
Waiting for client connected flag
Client connected with the Broker with code: 0.
Connected Flag:True
Arduino detected on com: COM10
Port:COM10
----Serial communication started----
Message from Arduino: A64E4353

```

3.2-2 Τα logs του Python script κατά την εκκίνησή του

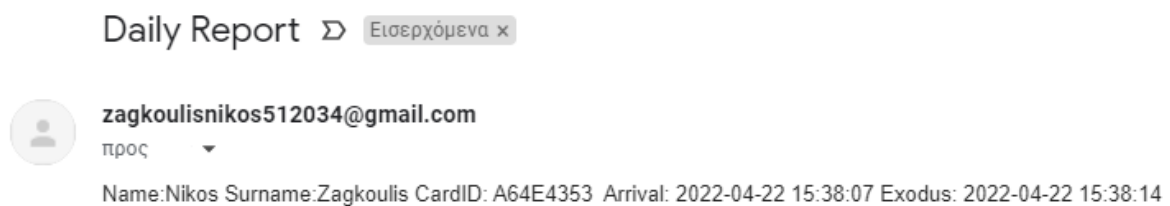
Όπως αναφέρθηκε παραπάνω, το σύστημα πρέπει να είναι σε θέση να κρατάει πληροφορίες εισόδου-εξόδου των εργαζομένων. Δεδομένου ότι το σύστημα σκαναρίσματος είναι το ίδιο και για την είσοδο αλλά και για την έξοδο, κρίθηκε αναγκαία η δημιουργία ενός Python script που ξεχωρίζει αν το σκανάρισμα αναφέρεται σε ώρα εισόδου ή εξόδου. Μ'αυτόν τον τρόπο, οι ώρες αυτές καταχωρούνται στα σωστά σημεία στην βάση. Η διαδικασία αυτή περιγράφεται παρακάτω.

Με το που σκανάρει ο εργαζόμενος και το ID του βραχιολιού φτάσει στο MQTT topic «DoorRfid», γίνεται αναζήτηση αν το βραχιόλι έχει σκαναριστεί ξανά μέσα στην ημέρα. Αν δεν έχει σκαναριστεί, γίνεται αναζήτηση αν το συγκεκριμένο βραχιόλι υπάρχει εγγεγραμμένο στην βάση. Αν δεν υπάρχει, αυτό σημαίνει ότι το βραχιόλι δεν ανήκει σε κάποιο εργαζόμενο. Έτσι, αποστέλλεται στο MQTT topic «DoorAccess» το μήνυμα “NotAuthorized” για να εμφανιστεί και το ανάλογο μήνυμα στην LCD οθόνη του συστήματος. Αν είναι εγγεγραμμένο στην βάση, αυτό σημαίνει ότι ο εργαζόμενος εισέρχεται στο κτίριο, και η ώρα σκαναρίσματος καταχωρείται ως ώρα εισόδου και αποθηκεύεται στην βάση στην καρτέλα «Attendance». Το μήνυμα τώρα που αποστέλλεται στο MQTT Topic «DoorAccess» είναι «Authorized», με τον STM32 να ανοίγει το ρελέ στο οποίο είναι συνδεδεμένη η μαγνητική κλειδαριά της πόρτας. Αν έχει σκαναριστεί το βραχιόλι ξανά μέσα στην ημέρα, αυτό σημαίνει ότι ο εργαζόμενος φεύγει από το κτίριο. Έτσι, η ώρα του σκαναρίσματος καταχωρείται ως ώρα εξόδου και μαζί με την ώρα εισόδου αποθηκεύονται στην βάση του συστήματος στην καρτέλα «Daily Report». Η καταχώρηση της ώρας εισόδου του διαγράφεται από την καρτέλα «Attendance» και το μήνυμα που αποστέλλεται στο «Door Access» είναι «Leaving».



3.2-3 Το διάγραμμα ροής του Python script για την επικοινωνία με το ESP32

Κατά την είσοδο και έξοδο των εργαζομένων, οι εγγραφές βρίσκονται στην καρτέλα «Daily Report» της βάσης . Με την βοήθεια του Python script, οι πληροφορίες των εργαζομένων της καρτέλας «Daily Report» αποθηκεύονται ως μήνυμα κειμένου και αποστέλλονται με e-mail στο διαχειριστικό e-mail της εταιρείας.



3.2-4 Παράδειγμα e-mail της ημερήσιας αναφοράς

Έπειτα, οι εγγραφές από την καρτέλα «Daily Report» αποστέλλονται στην καρτέλα «History» και διαγράφονται από την «Daily Report».

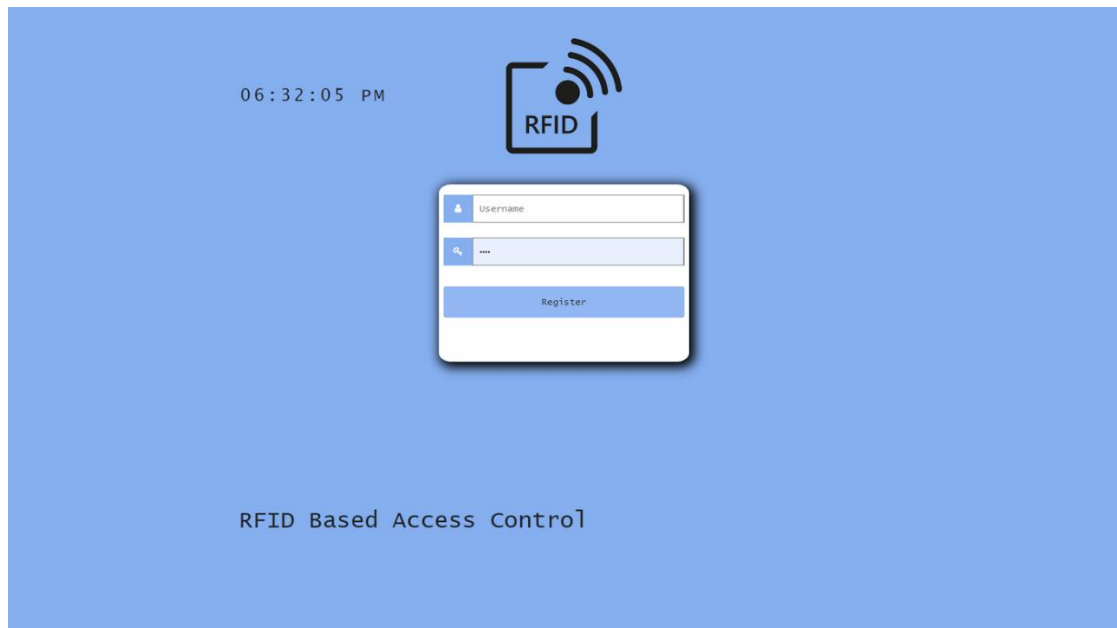
Κεφάλαιο 4^ο: Το front-end του συστήματος

4.1 Δημιουργία σελίδας εισόδου

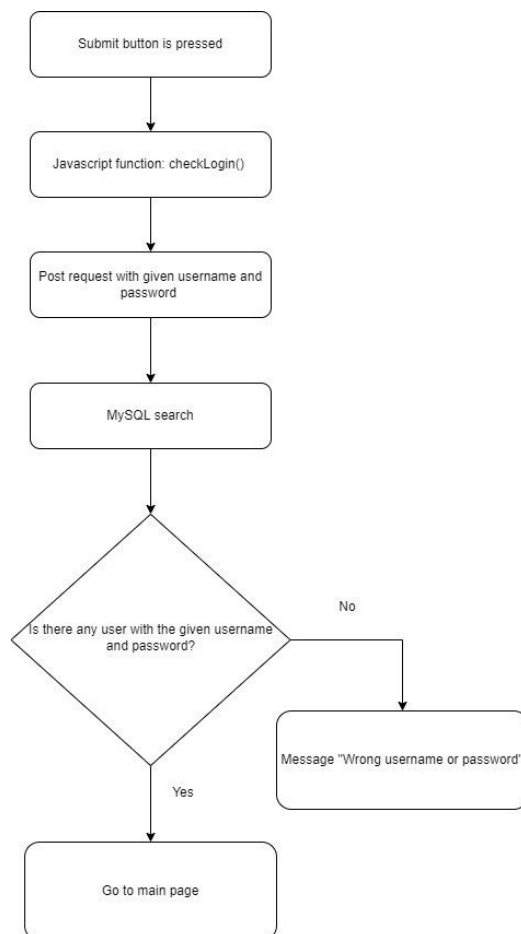
Η σελίδα εισόδου είναι η πρώτη σελίδα που εμφανίζεται κατά την είσοδο στην ιστοσελίδα του συστήματος.

Για να είναι σε θέση να εισέλθει κάποιος στην κύρια σελίδα, πρέπει πρώτα να πληκτρολογήσει το σωστό username και password.

Με την πληκτρολόγηση του σωστών credentials, δημιουργείται ένα request στην βάση μέσω PHP για επαλήθευση των στοιχείων (βλ. κεφάλαιο 3). Έπειτα, εμφανίζεται η κύρια σελίδα του συστήματος (βλ. δημιουργία κεντρικής σελίδας).

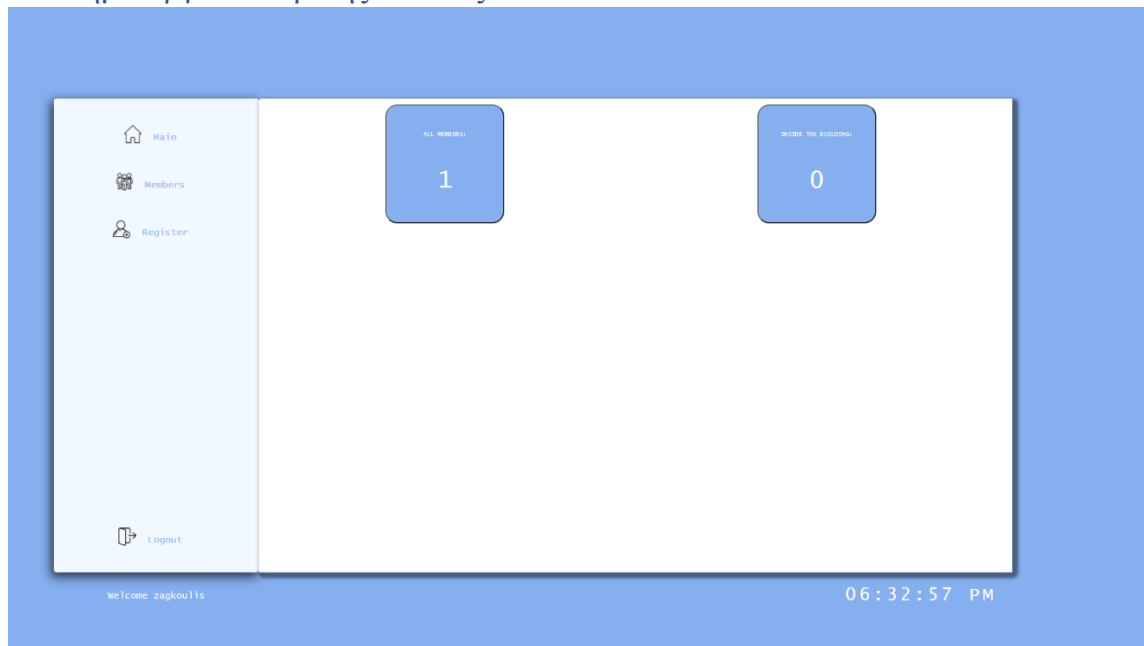


4.1-1 Η σελίδα εισόδου του συστήματος



4.1-2 Το διάγραμμα ροής της σελίδας εισόδου

4.2 Δημιουργία κεντρικής σελίδας

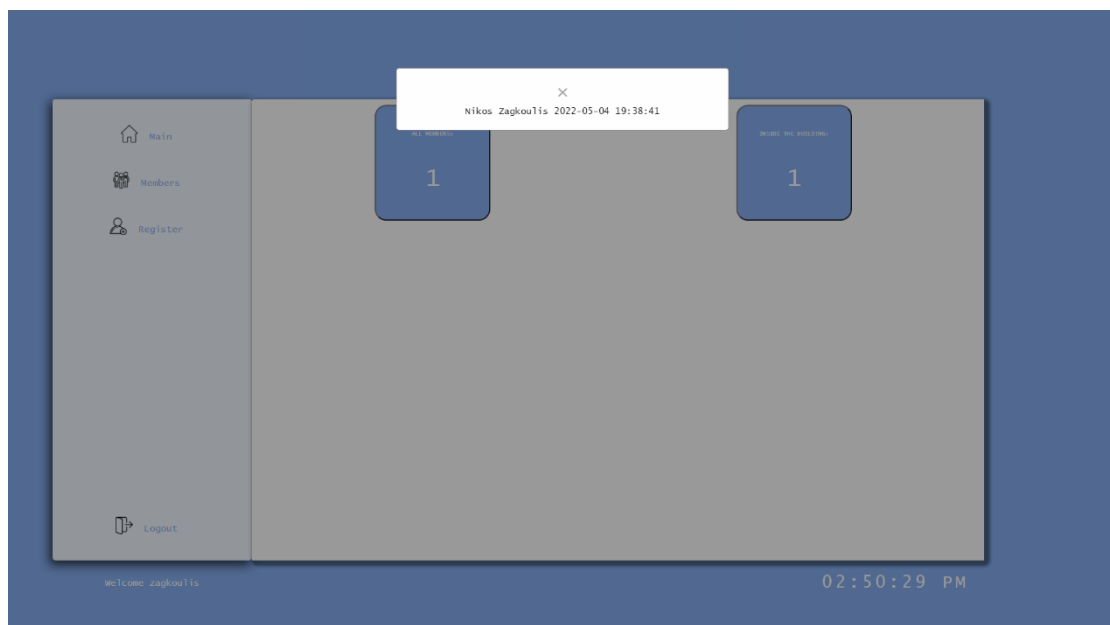


4.2-1 Η κεντρική σελίδα του συστήματος

Η κύρια σελίδα του συστήματος αποτελείται από ένα Sidebar με επιλογές όπως:

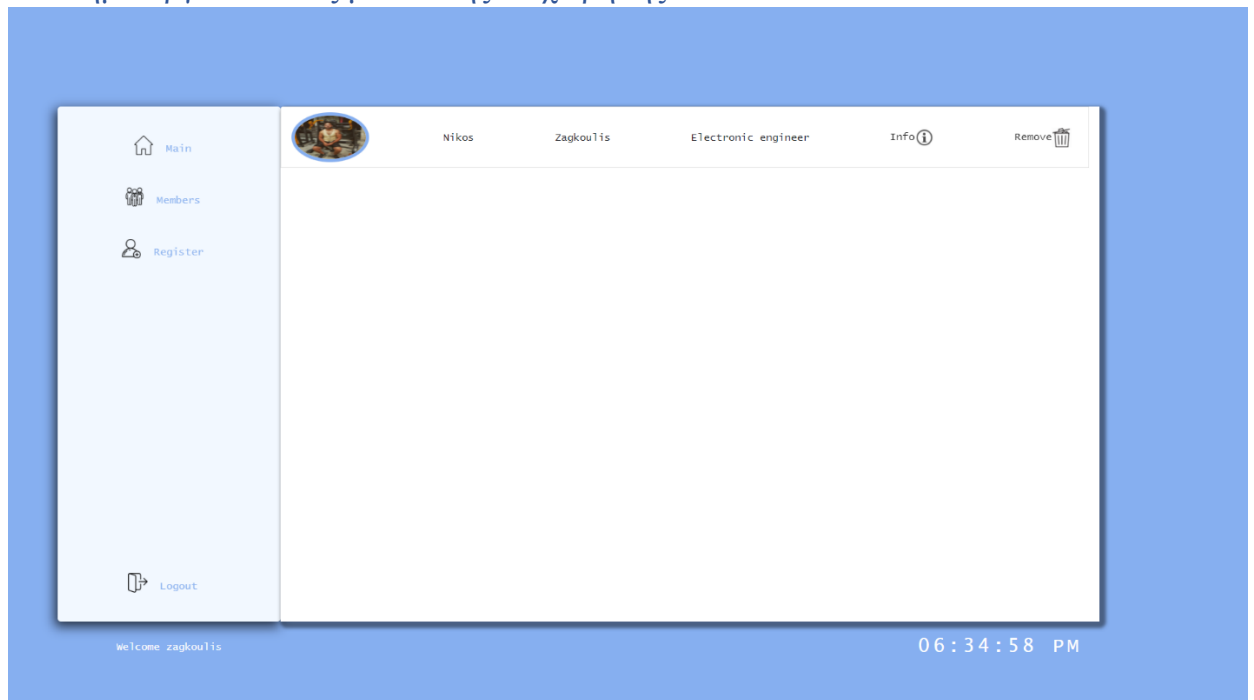
- Main Page, η οποία επιστρέφει στην παρούσα σελίδα
- Members, στην οποία εμφανίζεται η λίστα των μελών της επιχείρησης
- Register, όπου μπορεί να γίνει εγγραφή ενός νέου μέλους της επιχείρησης

Επιπλέον, υπάρχουν δύο Boxes στα οποία αναφέρονται τα συνολικά μέλη της επιχείρησης και αυτά τα οποία βρίσκονται μέσα στο κτίριο αντίστοιχα. Αν υπάρχουν μέλη μέσα στο κτίριο, με το πάτημα του Box εμφανίζεται ένα Pop-up Window με τα στοιχεία των εργαζομένων που βρίσκονται μέσα στο κτίριο καθώς και η ώρα εισόδου τους.



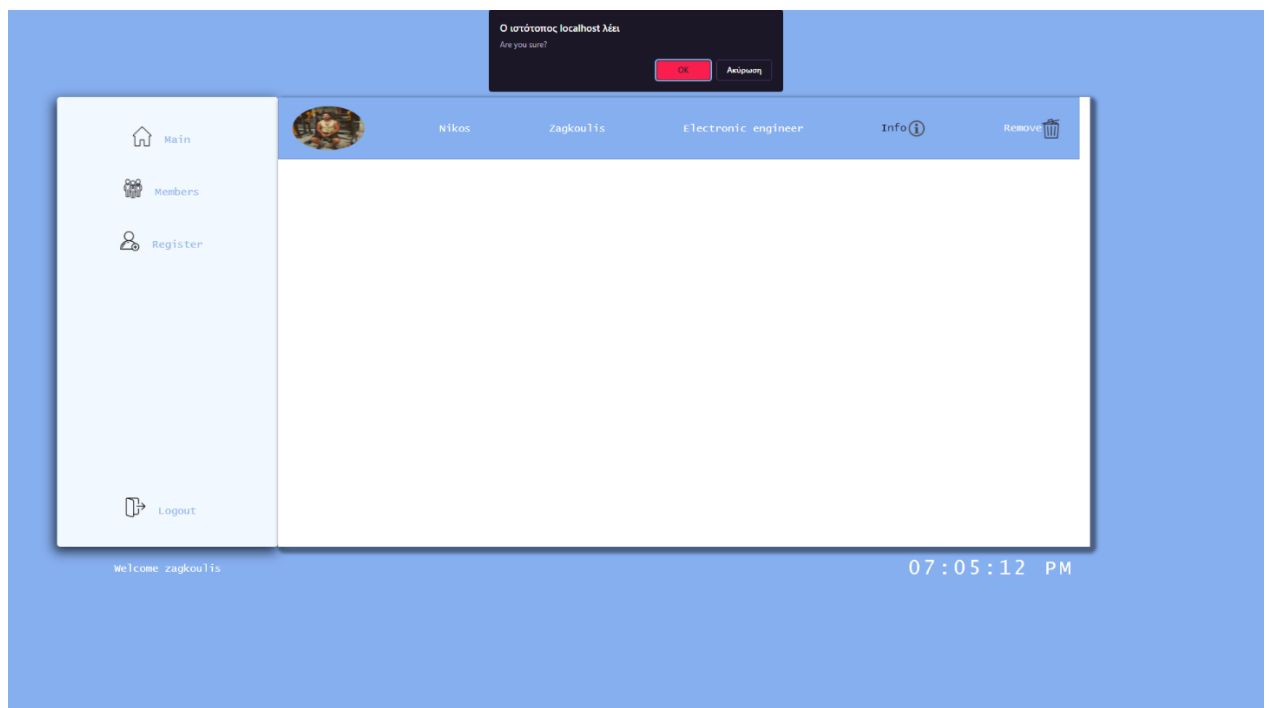
4.2-2 Το Pop-up Window των εργαζομένων που βρίσκονται εντός της επιχείρησης

4.3 Δημιουργία σελίδας μελών της επιχείρησης



4.3-1 Η σελίδα με τα μέλη της επιχείρησης

Στην σελίδα αυτή, παρουσιάζονται τα μέλη της επιχείρησης σε μία λίστα με την επιλογή Info και Remove. Με την επιλογή Info παρουσιάζεται το ιστορικό εισόδου-εξόδου του εργαζομένου ενώ με την επιλογή Remove ο εργαζόμενος διαγράφεται από την επιχείρηση και δεν έχει πλέον δυνατότητα εισόδου στο κτίριο.



4.3-2 Παράδειγμα διαγραφής ενός μέλους της επιχείρησης

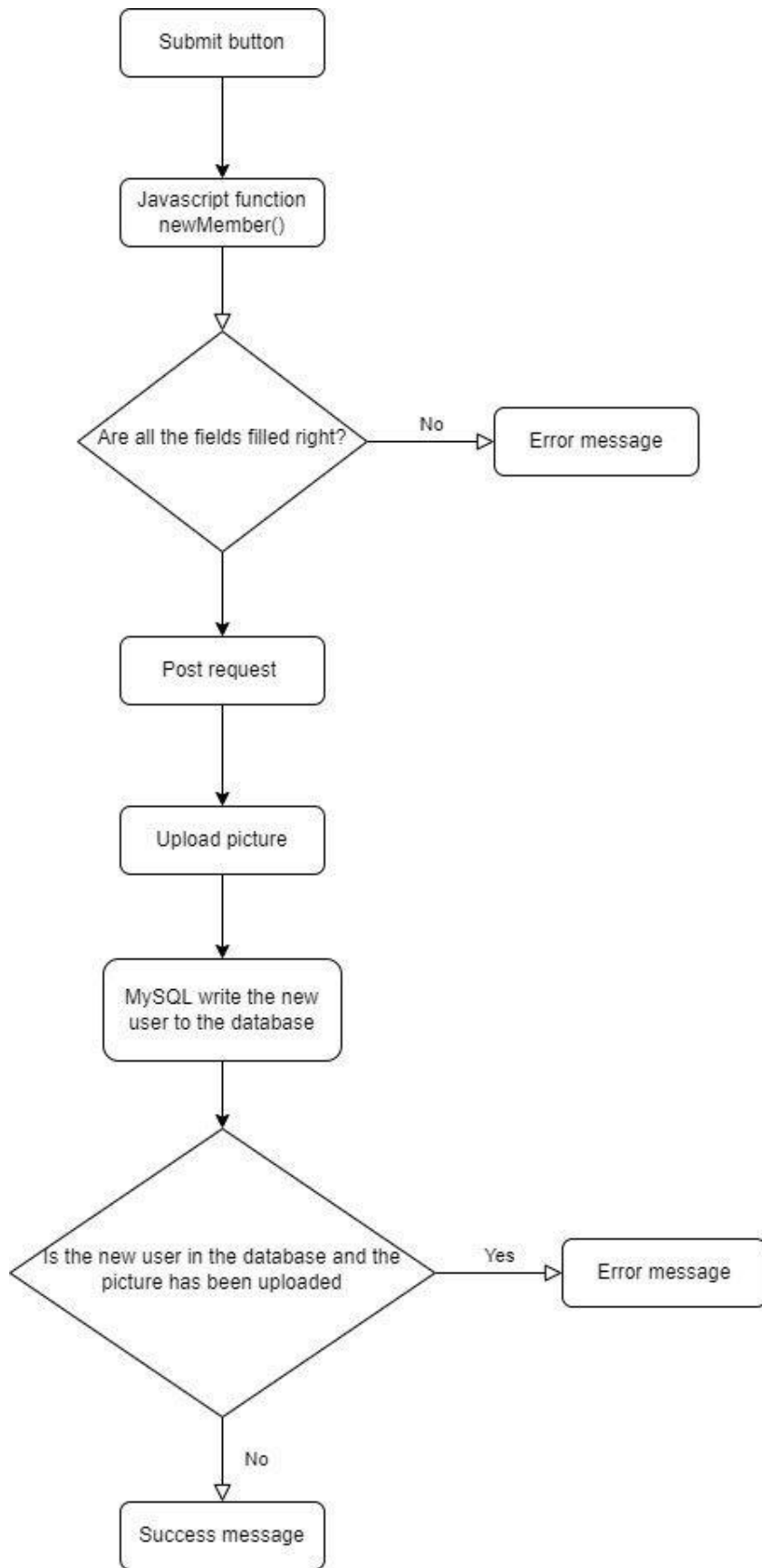
4.4 Δημιουργία σελίδας εγγραφής

The screenshot shows a web application interface for registration. On the left is a sidebar with navigation links: 'Main' (home icon), 'Members' (group icon), 'Register' (person icon), and 'Logout' (door icon). The main content area is titled 'Upload your image' and contains a registration form with the following fields: 'Enter your name', 'Surname', 'Enter your surname', 'Job' (with 'Graphic Designer' entered), 'Email address', 'Enter email', 'Password' (masked with dots), 'Confirm password', and 'Password' (repeated). Below the form is a 'Scan your wristband' section with a QR code icon and a 'Submit' button. At the bottom left, it says 'welcome zagkoulis' and at the bottom right, it shows the time '06:35:43 PM'.

4.4-1 Η σελίδα εγγραφής

Στην σελίδα αυτή πραγματοποιείται η εγγραφή ενός νέου χρήστη ως μέλος της επιχείρησης. Για να θεωρείται επιτυχής εγγραφή, ο χρήστης πρέπει να εισάγει:

- Φωτογραφία του νέου μέλους
- Όνομα
- Επίθετο
- Είδος εργασίας στο κτίριο (ηλεκτρονικός, graphic designer)
- Ηλεκτρονικό ταχυδρομείο
- Κωδικό καθώς και επιβεβαίωση κωδικού (οι δύο αυτοί κωδικοί πρέπει να συμπίπτουν)
- Τον κωδικό του βραχιολιού που θα χρησιμοποιεί ο νέος χρήστης, σκανάροντας το βραχιόλι στην συσκευή εγγραφής

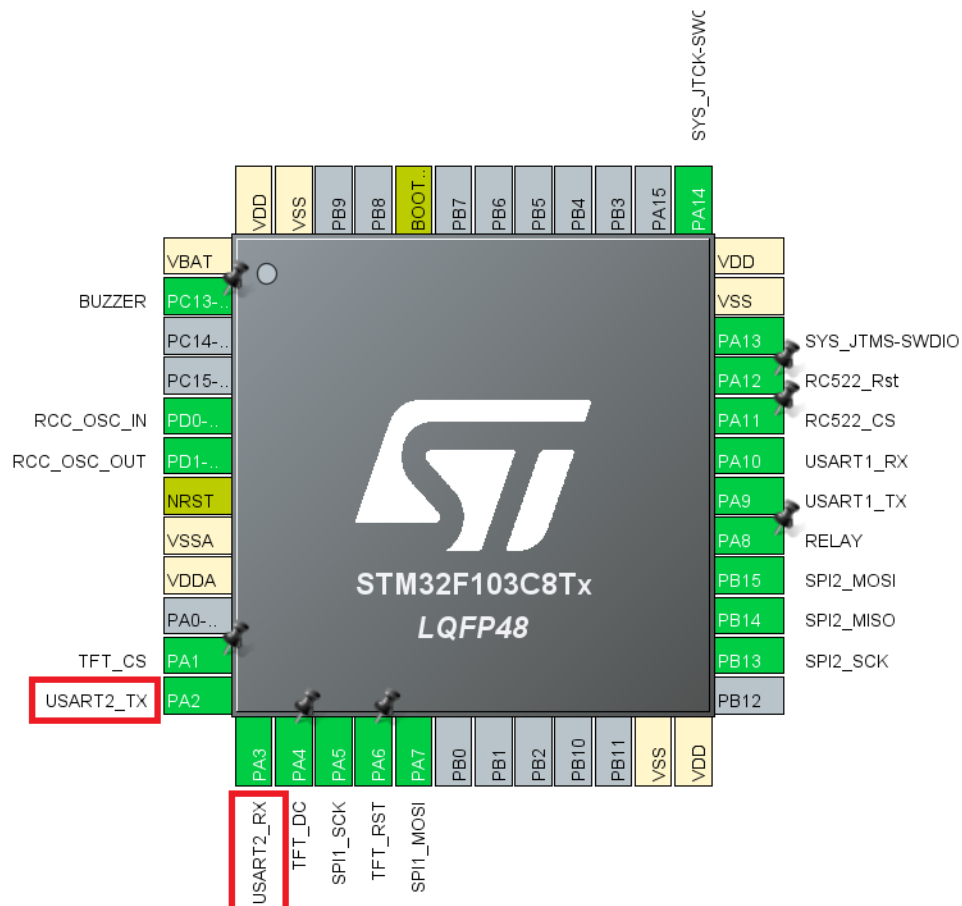


4.4-2 Το διάγραμμα ροής της σελίδας εγγραφής

Κεφάλαιο 5^ο: Το hardware του συστήματος

5.1 Σειριακή Επικοινωνία

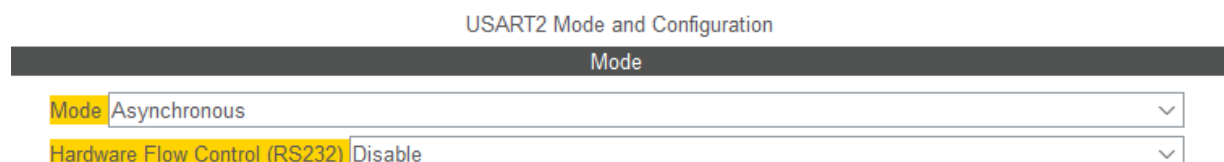
Η επικοινωνία ανάμεσα σε ESP32 και STM32 πραγματοποιείται σειριακά. Ο STM32 στέλνει το ID του βραχιολιού στο ESP ενώ το ESP32 στέλνει την απάντηση στον STM32. Για την σειριακή επικοινωνία, χρησιμοποιήθηκε η USART2 του STM32 όπως φαίνεται παρακάτω με baudrate 9600 bits/s.



5.1-1 Τα pins του STM32F103C8 για την σειριακή επικοινωνία

Τα pins του STM32F103C8 για την σειριακή επικοινωνία είναι:

- PA2 για TX
- PA3 για RX



5.1-2 Το mode της σειριακής επικοινωνίας



5.1-3 To configuration της σειριακής επικοινωνίας

5.2 To ESP32

Το IDE με το οποίο πραγματοποιήθηκε ο προγραμματισμός του ESP32 είναι το Arduino IDE. Το ESP32 προτιμήθηκε λόγω του Wi-Fi Module για την επικοινωνία ανάμεσα στον MQTT και STM32. Οι λειτουργίες του είναι οι εξής:

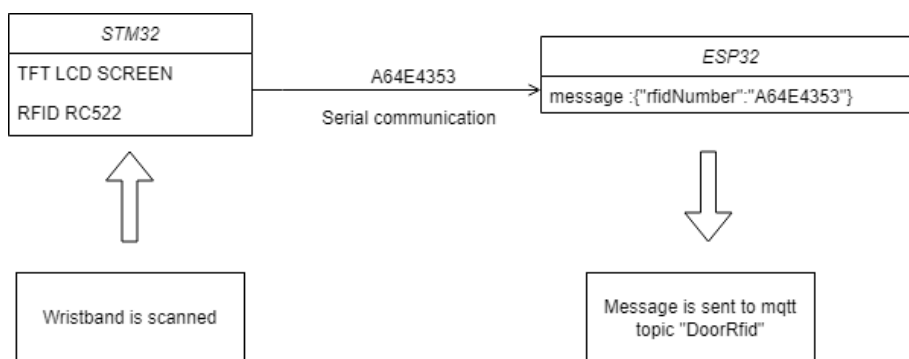
- Σύνδεση στο Wi-Fi δίκτυο
- Εγγραφή ως MQTT Client και Subscribe στο MQTT topic «Door Access»
- Σειριακή επικοινωνία με τον STM32F103C8

Κατά την εκκίνηση του προγράμματος, γίνεται προσπάθεια σύνδεσης στο Wi-Fi για να είναι εφικτή η σύνδεση με τον MQTT broker του συστήματος.

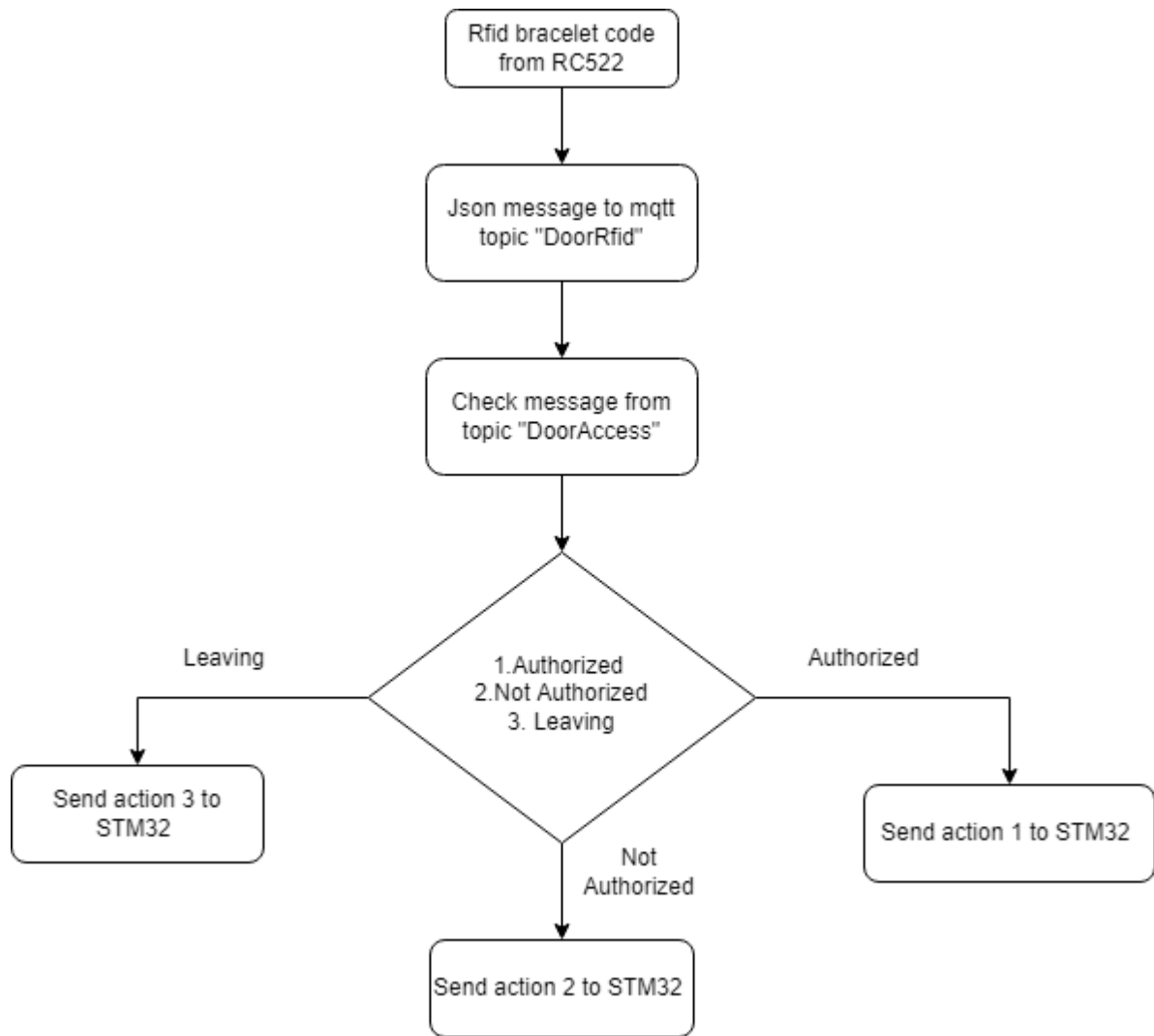
Κατά την διάρκεια του προγράμματος, σε περίπτωση που για κάποιο λόγο γίνει αποσύνδεση του ESP από το Wi-Fi, γίνεται προσπάθεια επανασύνδεσης.

Μετά την εγγραφή στο Wi-Fi, γίνεται προσπάθεια σύνδεσης στον MQTT broker του συστήματος. Εφόσον το ESP32 έχει συνδεθεί ως MQTT Client, γίνεται Subscribe στο topic «DoorAccess». Το topic αυτό είναι υπεύθυνο για τα μηνύματα τα οποία θα στείλει πίσω το ESP32 στον STM32. Ομοίως με την σύνδεση στο Wi-Fi, σε περίπτωση αποσύνδεσης από τον MQTT κατά την διάρκεια του προγράμματος, γίνεται προσπάθεια επανασύνδεσης.

Κατά την λήψη του ID του βραχιολιού από τον STM32, το ID για να σταλθεί στο MQTT topic «DoorRfid» πρέπει να μετατραπεί σε μορφή Json. Το παράδειγμα αυτής της διαδικασίας περιγράφεται παρακάτω.



5.2-1 Παράδειγμα δημιουργίας Json μηνύματος



5.2-2 Διάγραμμα ροής της λειτουργίας του ESP32

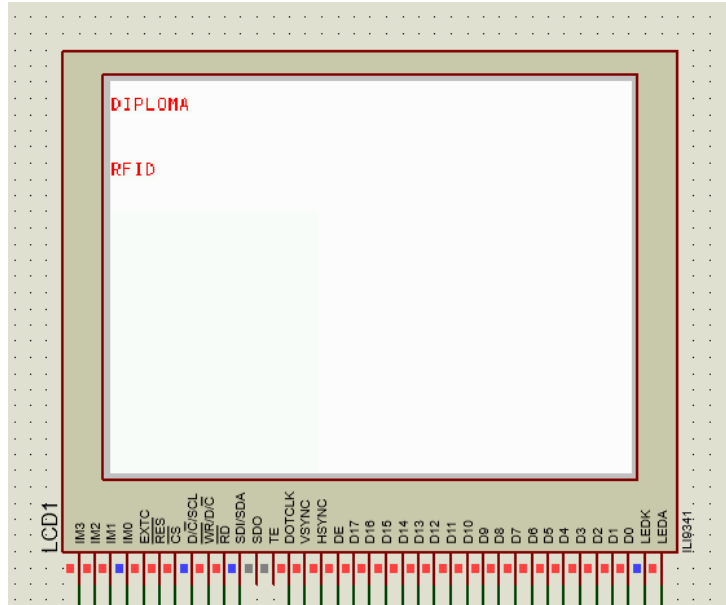
5.3 Ο STM32F103C8

Ο κώδικας του STM32 πραγματοποιήθηκε με το λογισμικό Cube IDE (για τη εξαγωγή του αρχείου HEX) και το λογισμικό STM32 ST-LINK Utility για τον προγραμματισμό του.



5.3-1 LCD TFT screen 160x128 with ST7735

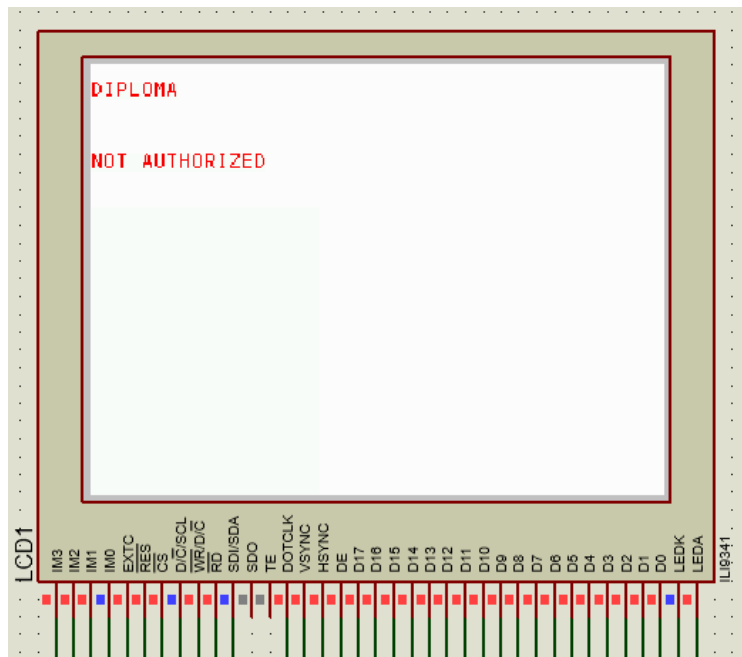
Για να είναι σε θέση ο εργαζόμενος να γνωρίζει την κατάσταση του σκαναρίσματος του (επιτυχές , ανεπιτυχές σκανάρισμα), χρησιμοποιήθηκε μία οθόνη LCD 2.2 ιντσών. Το πρωτόκολλο επικοινωνίας ανάμεσα στον STM32 και στην οθόνη είναι το SPI.



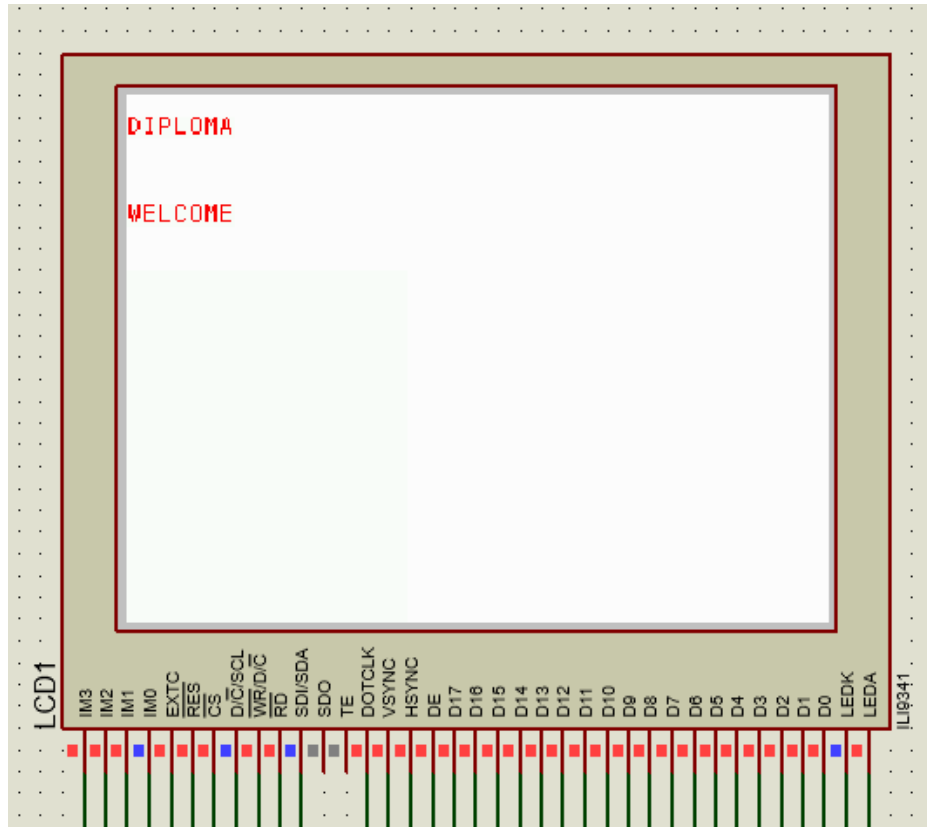
5.3-2 LCD TFT κατά την εκκίνηση του προγράμματος

Κατά την εκκίνηση του συστήματος, εμφανίζονται στην οθόνη DIPLOMA RFID. Κατά το σκανάρισμα του βραχιολιού εμφανίζονται 3 πιθανές απαντήσεις:

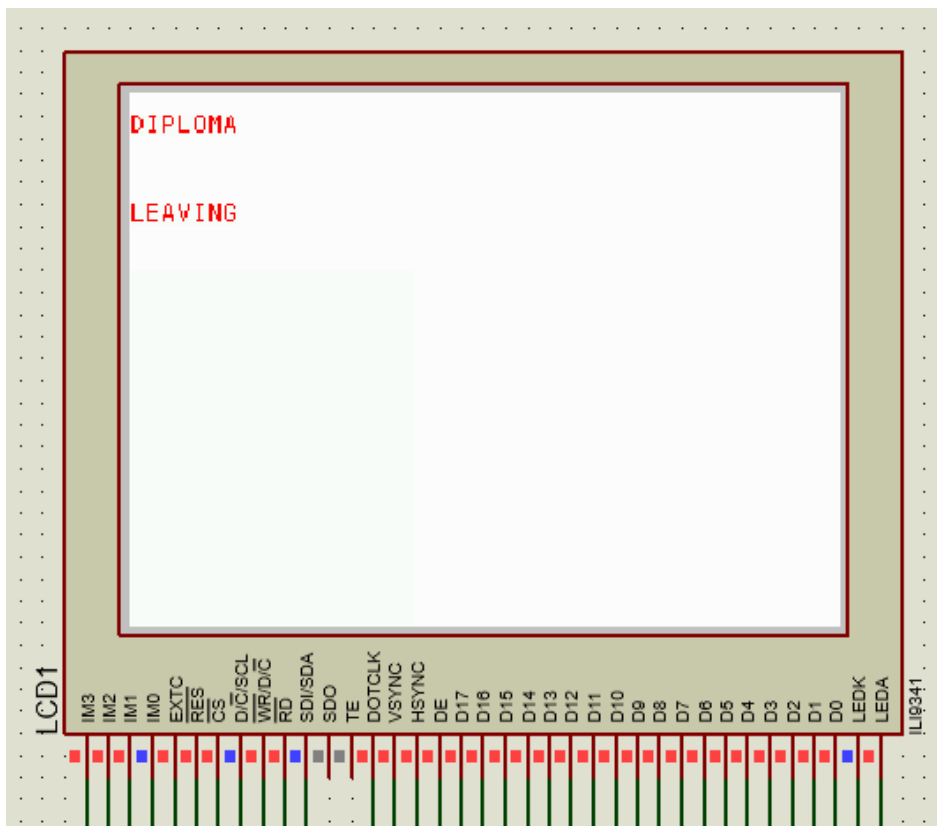
- Welcome
- Leaving
- Not Authorized



5.3-3 LCD TFT κατά την προσπάθεια εισόδου μη εξουσιοδοτημένου βραχιολιού

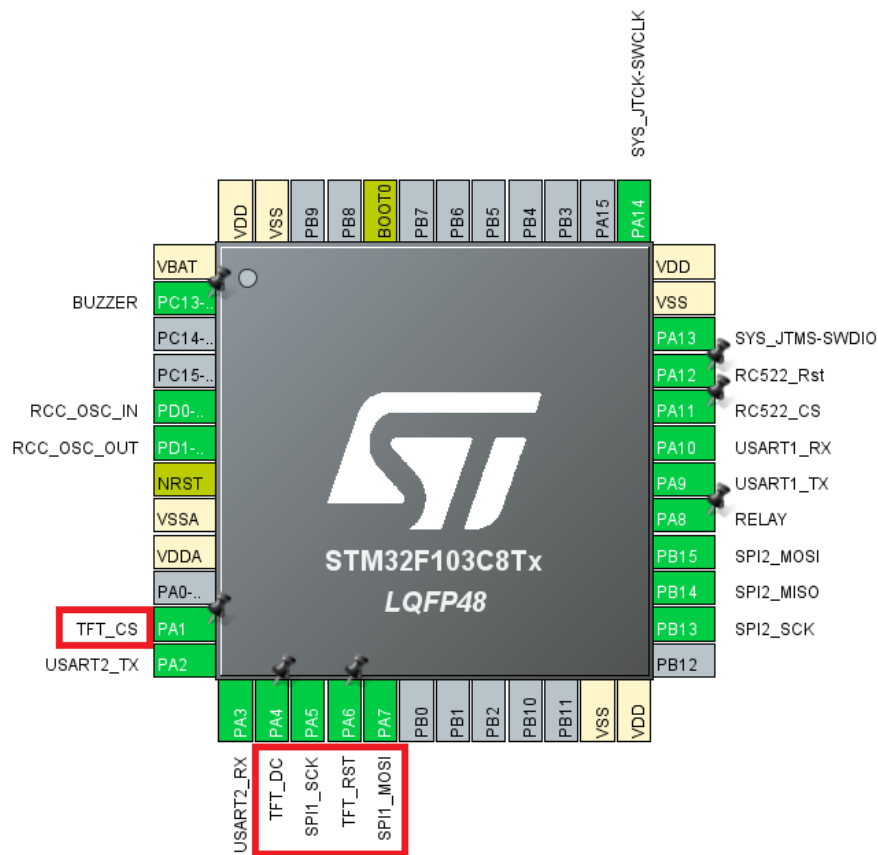


5.3-4 LCD TFT κατά την επιτυχής είσοδο ενός εργαζομένου στο κτίριο



5.3-5 LCD TFT κατά την έξοδο ενός εργαζομένου από το κτίριο

Το πρώτο βήμα για την σύνδεση ανάμεσα στο STM32 με την TFT οθόνη είναι ο ορισμός των pins στην εφαρμογή Cube IDE. Με τον ορισμό των pins του SPI σ'αυτό το σημείο, το Cube είναι υπεύθυνο να κάνει generate τις κατάλληλες εντολές με τις οποίες αρχικοποιείται το SPI.



5.3-6 Τα pins του STM32F103C8 για το SPI της LCD οθόνης

Για την επικοινωνία ανάμεσα STM32 και οθόνης, χρησιμοποιήθηκε το SPI1 του μικροελεγκτή όπως φαίνεται στην παραπάνω εικόνα. Συγκεκριμένα τα pins είναι:

- PA1 για το CS της οθόνης
- PA4 για το DC της οθόνης
- PA5 για SCK του SPI1
- PA6 για το RST της οθόνης
- PA7 για το MOSI του SPI1

SPI1 Mode and Configuration

Mode	
Mode	Half-Duplex Master
Hardware NSS Signal	Disable

5.3-7 Το mode του STM32F103C8 για το SPI της οθόνης

Το mode του SPI1 έχει ρυθμιστεί ως Half-Duplex Master. Αυτό σημαίνει ότι και οι δύο συσκευές μπορούν να στείλουν δεδομένα, αλλά όχι ταυτόχρονα. Η επικοινωνία γίνεται προς μία κατεύθυνση κάθε φορά (είτε STM32 προς οθόνη, είτε οθόνη προς STM32).

Configuration

Reset Configuration

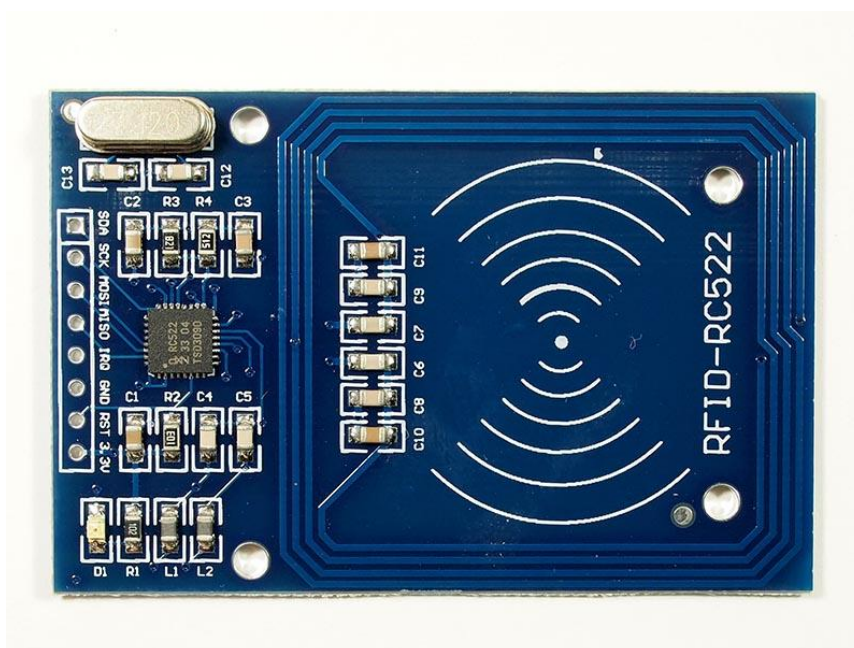
✓ NVIC Settings	✓ DMA Settings	✓ GPIO Settings
✓ Parameter Settings	✓ User Constants	

Configure the below parameters :

Search (Ctrl+F) [↶] [↷] ⓘ

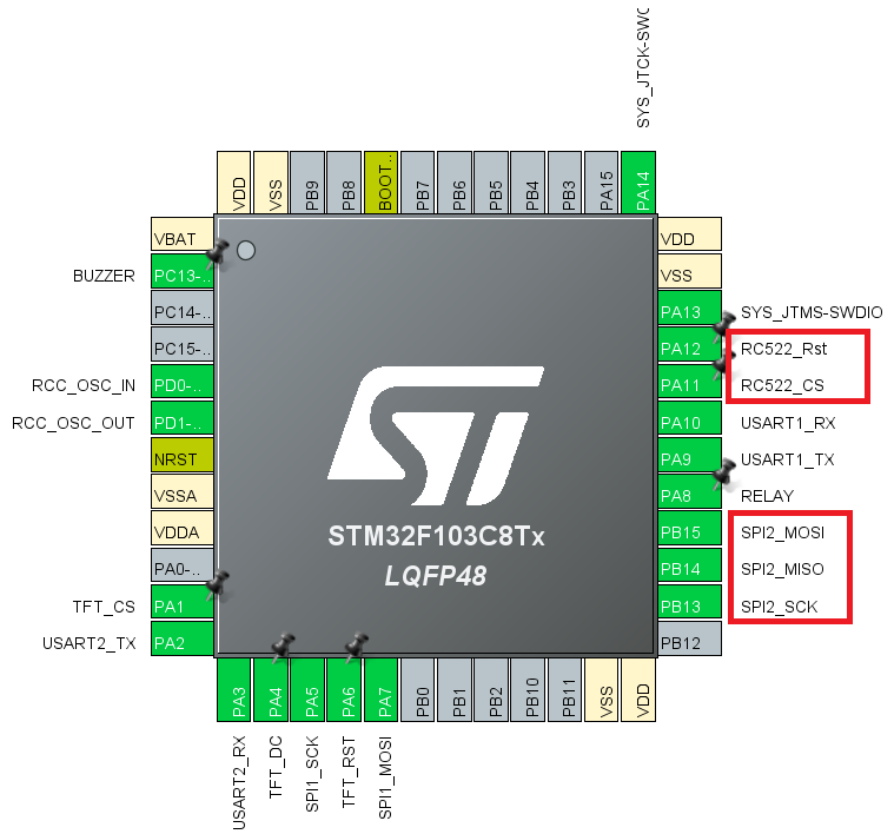
- ✓ Basic Parameters
 - Frame Format: Motorola
 - Data Size: 8 Bits
 - First Bit: MSB First
- ✓ Clock Parameters
 - Prescaler (for Baud Rate): 64
 - Baud Rate: 1.125 MBits/s
 - Clock Polarity (CPOL): Low
 - Clock Phase (CPHA): 1 Edge
- ✓ Advanced Parameters
 - CRC Calculation: Disabled
 - NSS Signal Type: Software

5.3-8 Το configuration του STM32F103C8 για το SPI της οθόνης



5.3-9 RC522 RFID Card Reader Module

Το RFID RC522 χρησιμοποιεί επίσης το πρωτόκολλο επικοινωνίας SPI για την επικοινωνία του με τον STM32. Ο ορισμός των pins έγινε με την εφαρμογή Cube IDE όπως φαίνεται παρακάτω.



5.3-10 Τα pins του STM32F103C8 για το SPI του RFID

Για την επικοινωνία ανάμεσα STM32 και RFID, χρησιμοποιήθηκε το SPI2 του μικροελεγκτή όπως φαίνεται στην παραπάνω εικόνα. Συγκεκριμένα τα pins είναι:

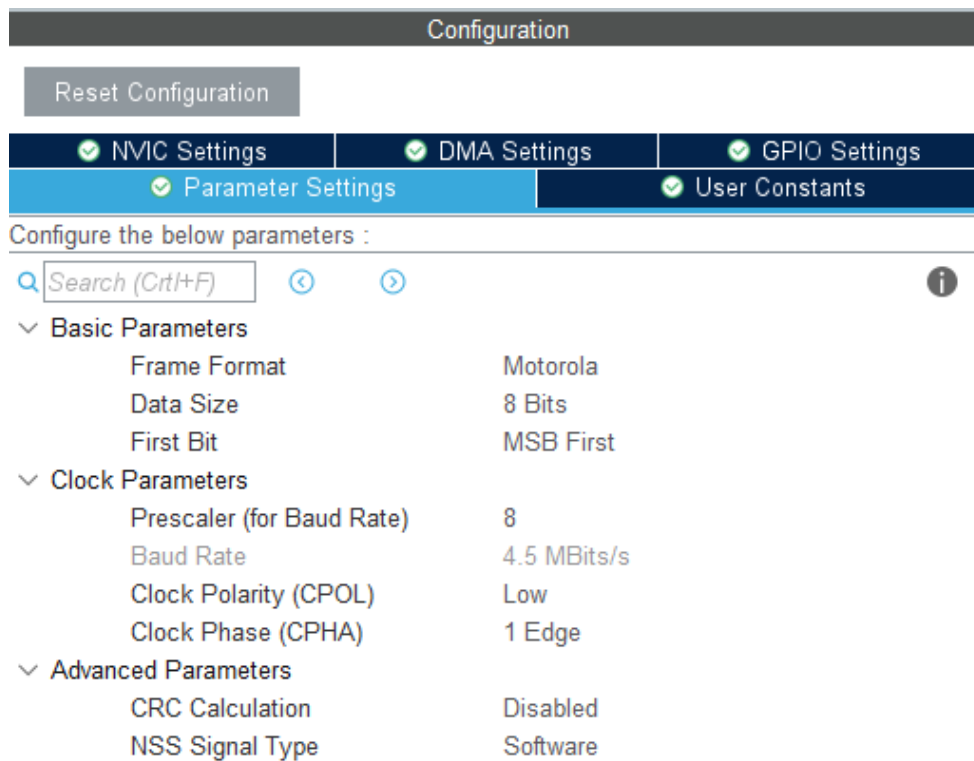
- PA11 για το CS του RFID
- PA12 για το RST του RFID
- PB13 για SCK του SPI2
- PB14 για το MISO του SPI2
- PB15 για το MOSI του SPI2

SPI2 Mode and Configuration

Mode	
Mode	Full-Duplex Master
Hardware NSS Signal	Disable

5.3-11 Το mode του STM32F103C8 για το SPI του RFID

Το mode του SPI2 έχει ρυθμιστεί ως Full-Duplex Master. Αυτό σημαίνει ότι και οι δύο συσκευές μπορούν να στείλουν ταυτόχρονα δεδομένα.

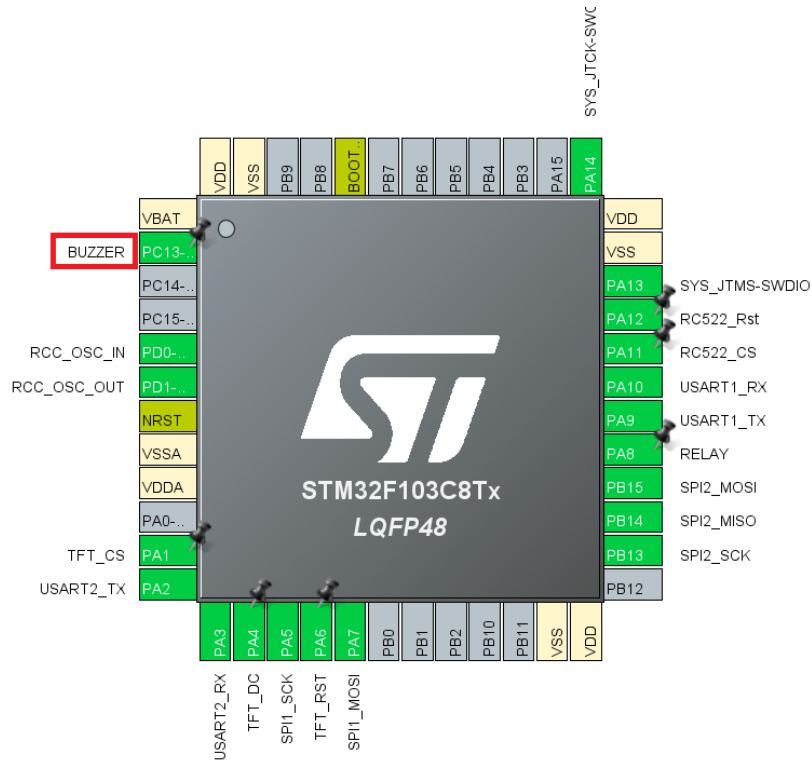


5.3-12 To configuration του STM32F103C8 για το SPI της RFID



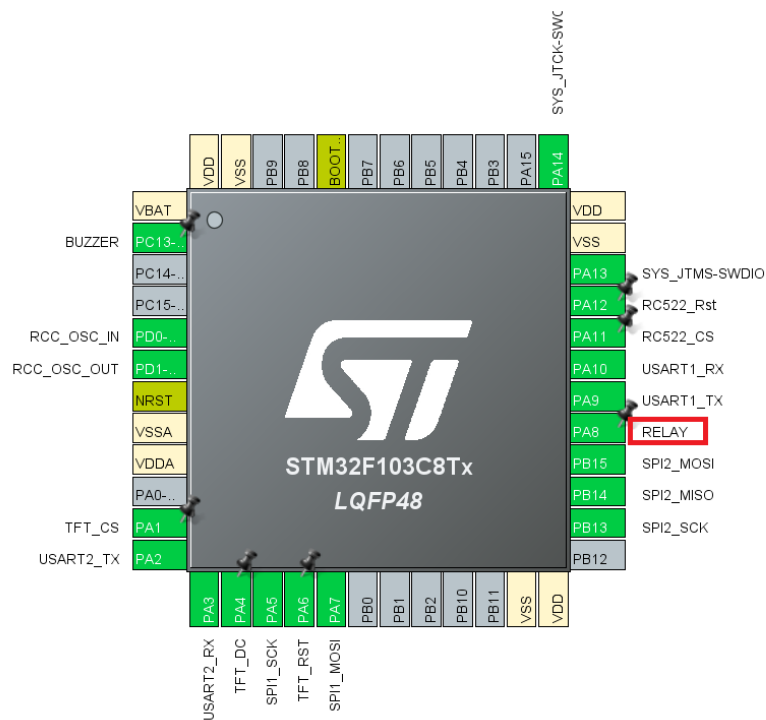
5.3-13 Το buzzer του συστήματος

Για να είναι σε θέση ο χρήστης να ενημερώνεται αν το βραχιόλι του έχει σκαναριστεί, τοποθετήθηκε ένα buzzer το οποίο ηχεί για δύο δευτερόλεπτα από την στιγμή που σκανάρει ο χρήστης.

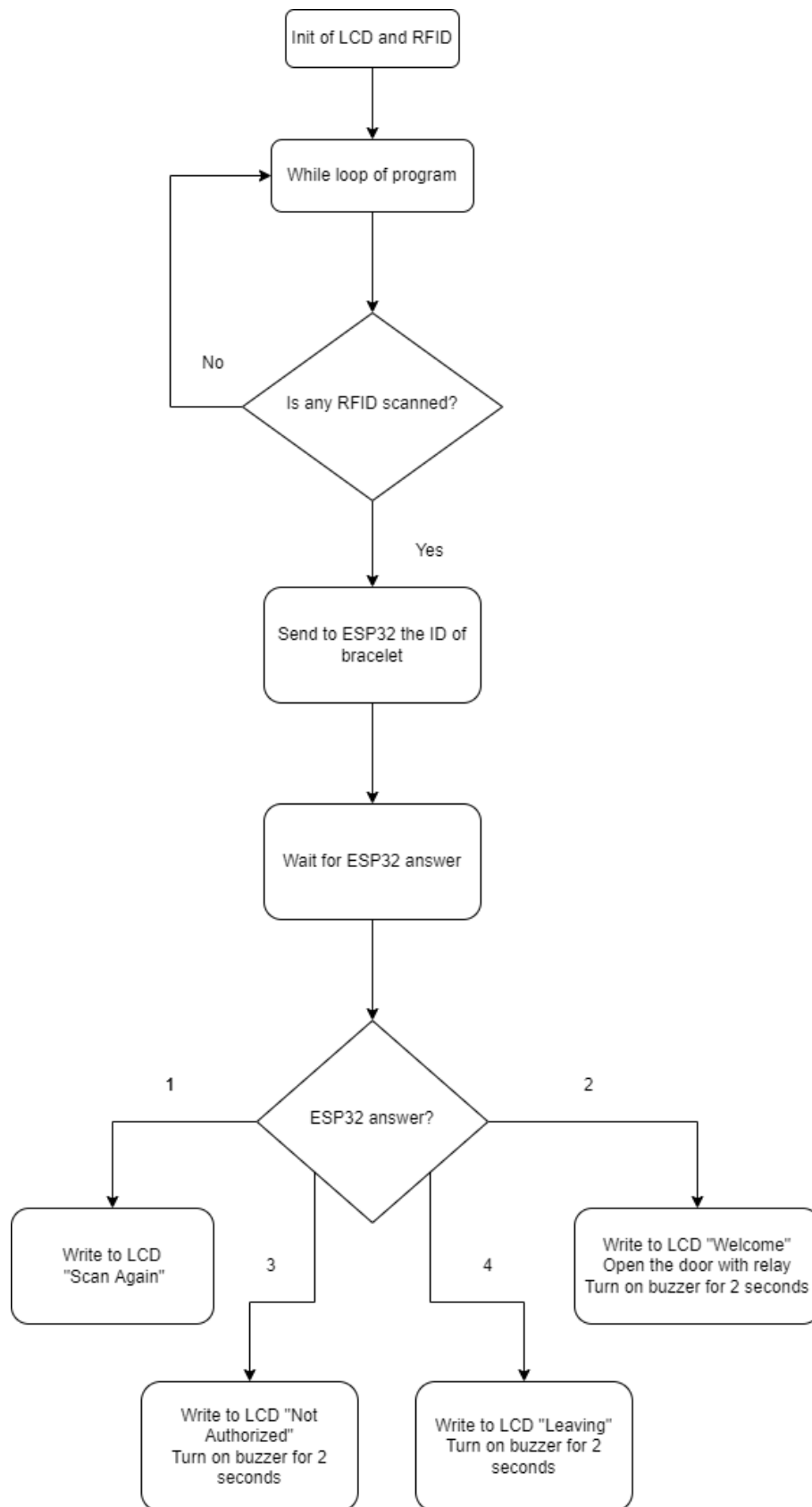


5.3-14 Το pin του STM32F103C8 για το buzzer του συστήματος

Το pin του buzzer είναι το PC13 (GPIO OUTPUT). Επίσης, προστέθηκε ένα κύκλωμα relay στο οποίο μπορεί να συνδεθεί η μαγνητική κλειδαριά του κτιρίου. Μ'αυτόν τον τρόπο, η πόρτα ανοίγει μόνο όταν το βραχίολι το οποίο σκαναρίστηκε ανήκει σε κάποιον εργαζόμενο.



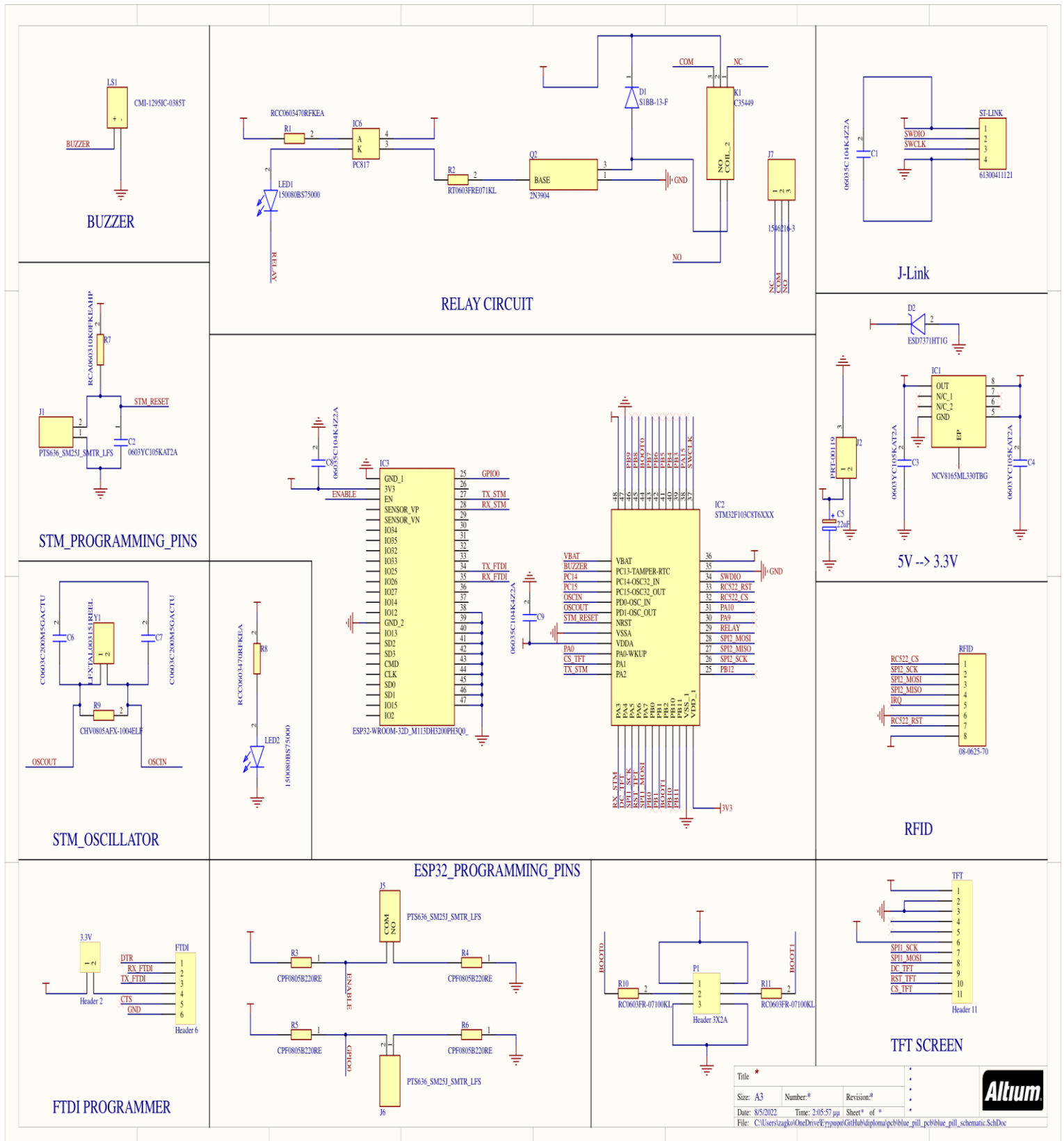
5.3-15 Το pin του STM32F103C8 για το relay του συστήματος



5.3-16 Το διάγραμμα ροής του προγράμματος του STM32F103C8

5.4 Σχεδιασμός της πλακέτας

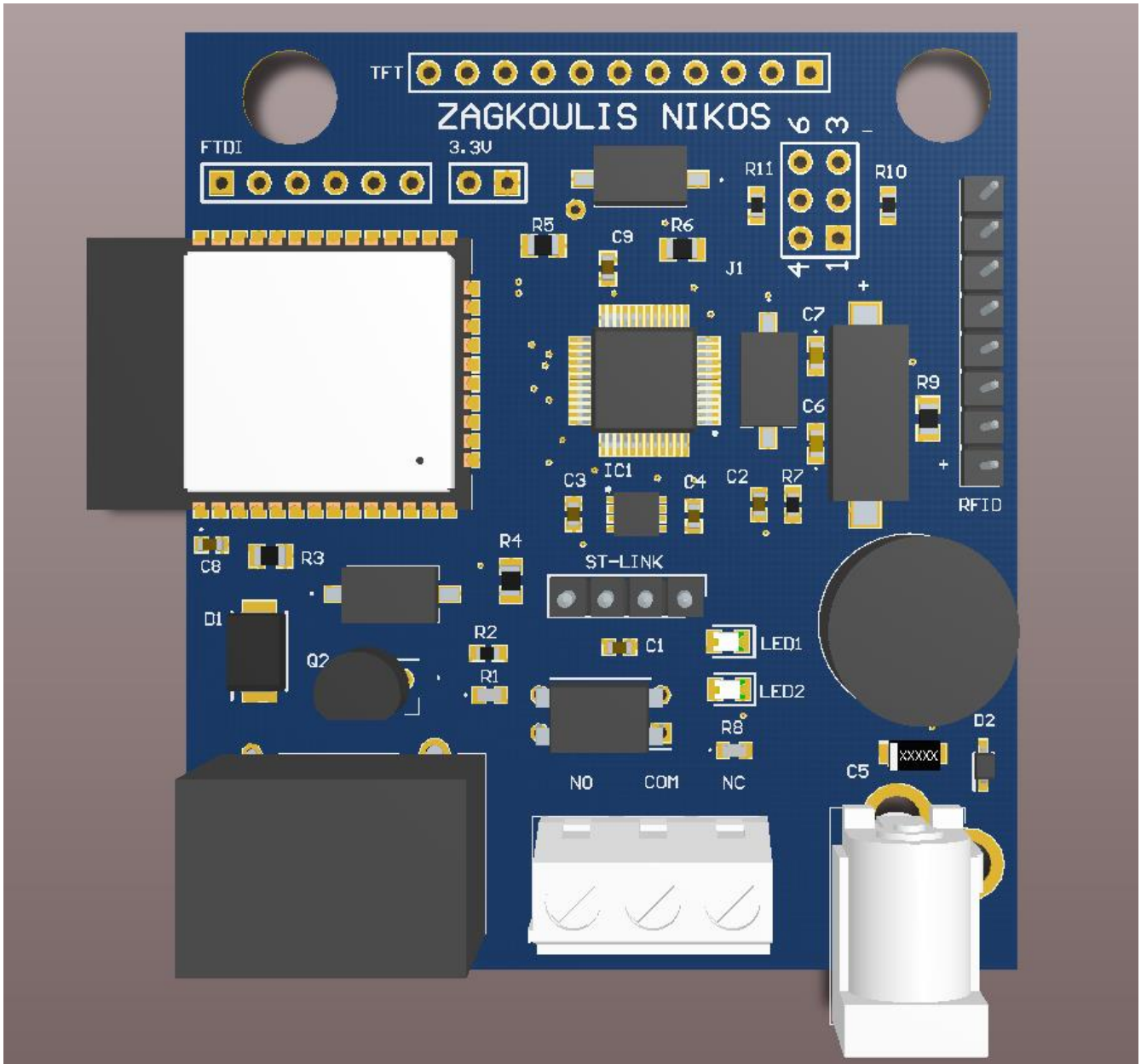
Για την δημιουργία της πλακέτας του συστήματος προτιμήθηκε το Altium software.



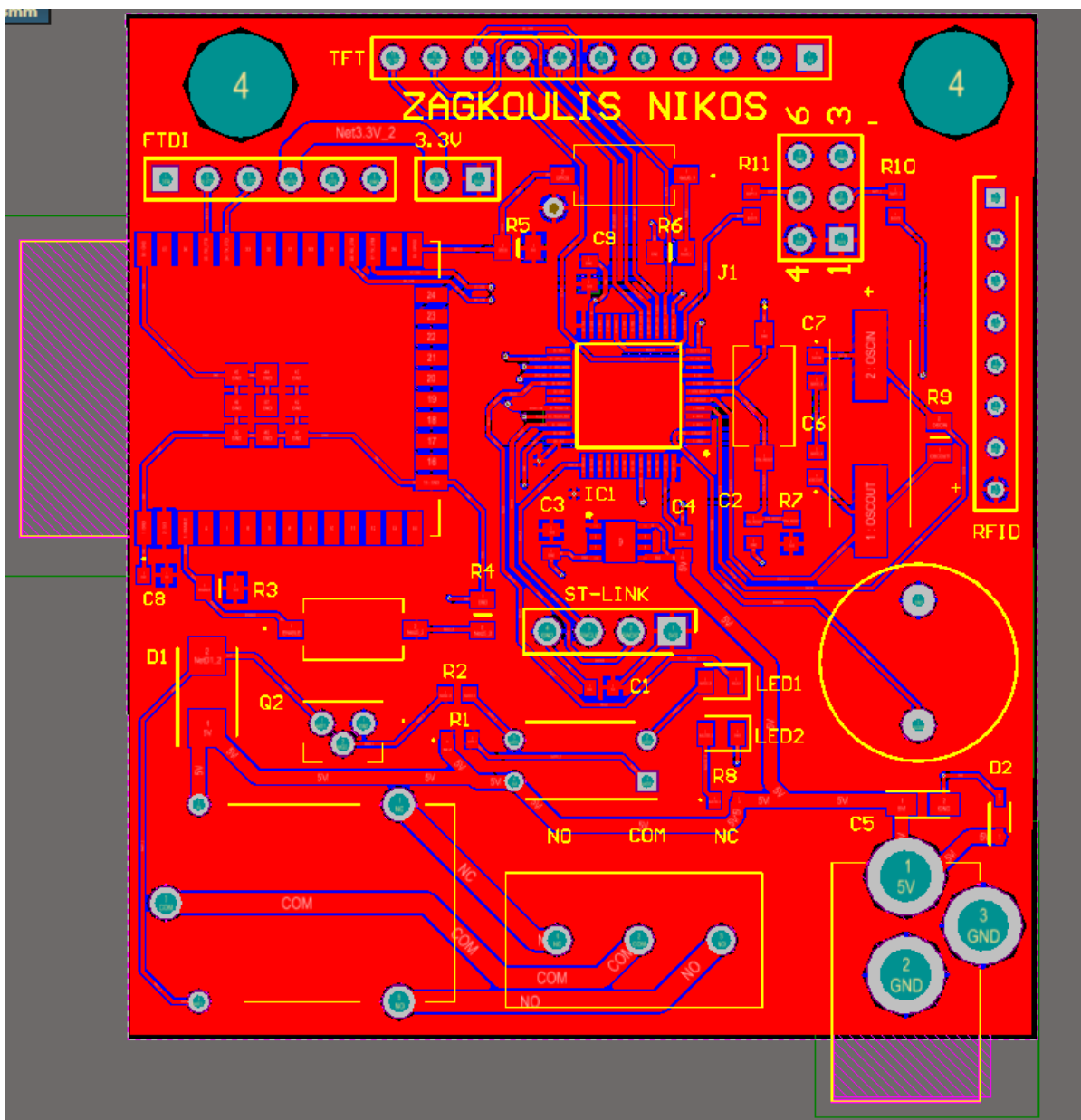
5.4-1 Το σχηματικό της πλακέτας

Η πλακέτα αποτελείται από:

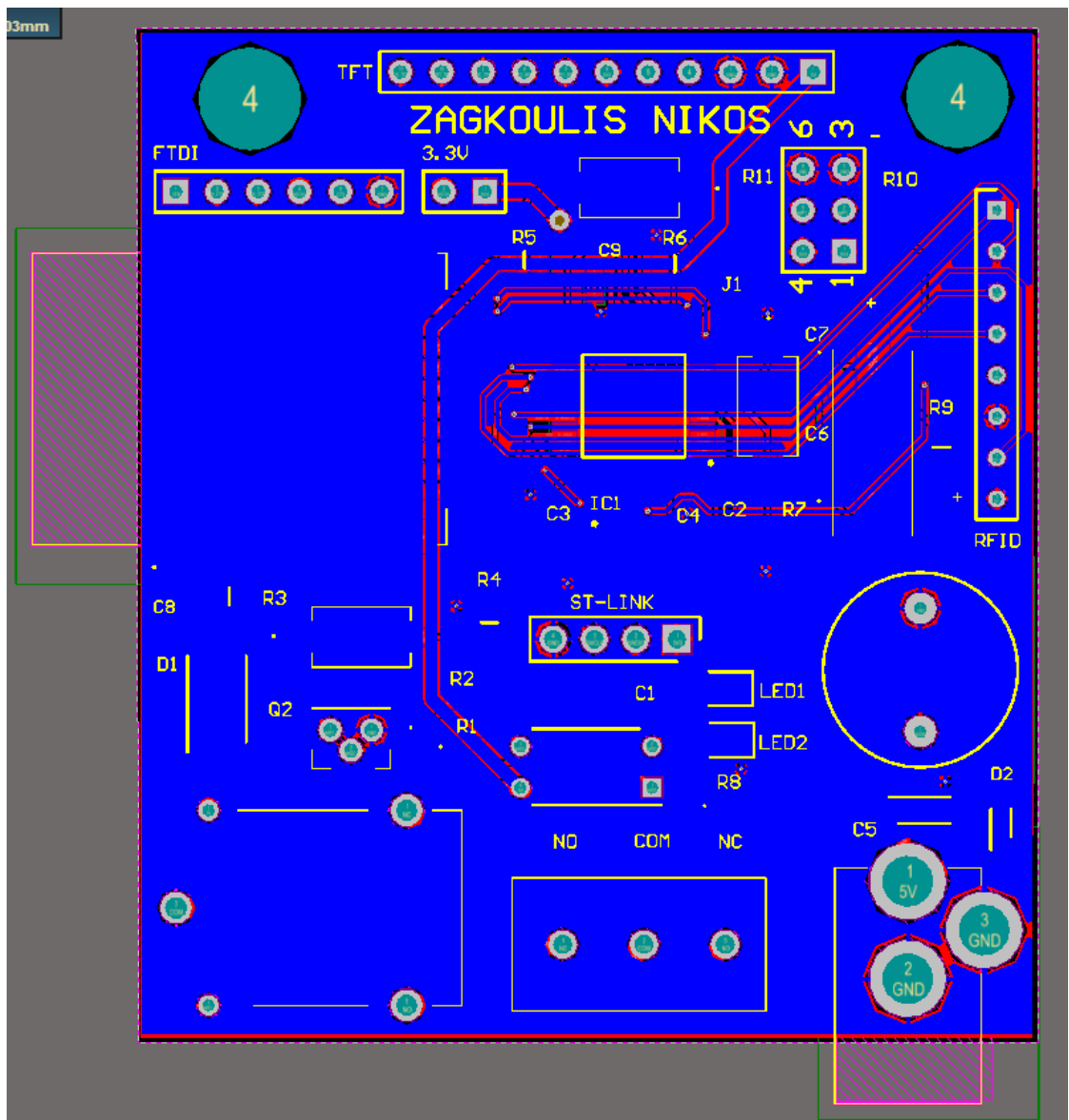
- ESP32 μαζί με tactile buttons (enable pin, reset pin) και υποδοχή για FTDI για τον προγραμματισμό του
- STM32F103C8T6 με pin headers για επιλογή των enable pin και reset pin και υποδοχή για ST-LINK για τον προγραμματισμό του
- Κύκλωμα relay με optocoupler για isolation για την προσθήκη μαγνητικής κλειδαριάς
- Υποδοχή (pin headers) για RFID RC522
- Υποδοχή (pin headers) για TFT LCD



5.4-2 Η πλακέτα του συστήματος σε 3D όψη



5.4-3 Το πάνω μέρος της πλακέτας του συστήματος



5.4-4 Το κάτω μέρος της πλακέτας του συστήματος



5.4-5 Η πλακέτα του συστήματος

Κεφάλαιο 6^ο: Ο server του συστήματος

6.1 Επιλογή Server

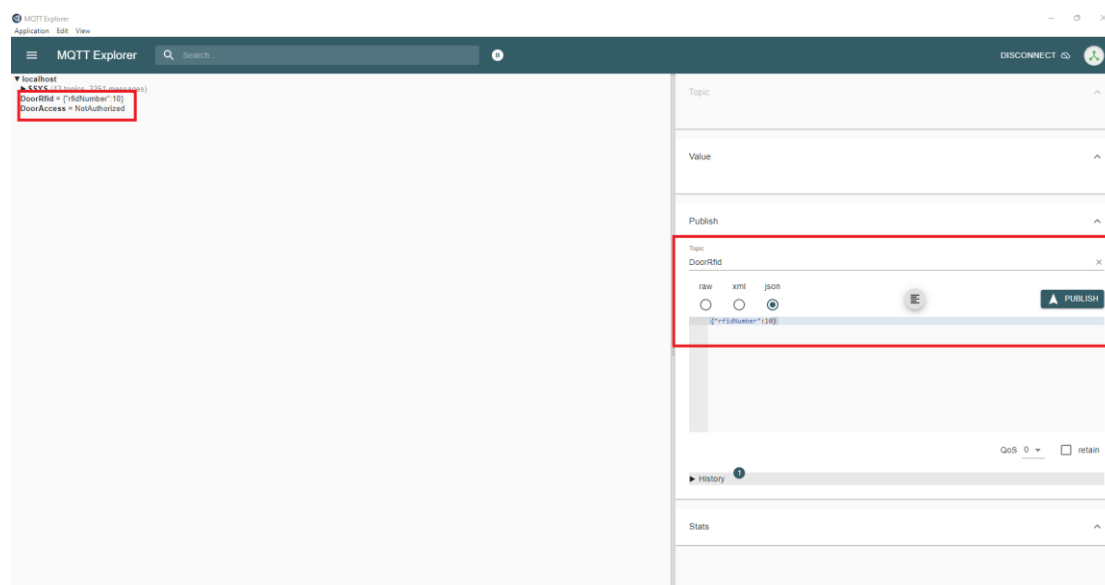
Για να είναι σε θέση το σύστημα να τεθεί σε λειτουργία, κρίθηκε αναγκαία η χρησιμοποίηση ενός υπολογιστή για τις εξής λειτουργίες:

- MQTT mosquito broker
- MYSQL βάση δεδομένων
- Apache webserver software
- Python script του συστήματος

Σε συστήματα με τόσο συγκεκριμένη λειτουργία ενός υπολογιστή και με όχι τόσο μεγάλες απαιτήσεις απόδοσης, η επιλογή ενός Raspberry Pi είναι η συνηθέστερη. Για το συγκεκριμένο σύστημα, η προσπάθεια ενσωμάτωσης των παραπάνω λειτουργιών σε Raspberry Pi (Zero 2 W) είχε ως αποτέλεσμα την όχι και τόσο γρήγορη απόκριση του συστήματος. Συγκεκριμένα, κατά το σκανάρισμα ενός βραχιολιού και την αποστολή του RFID ID στο MQTT topic, η αναζήτηση του ID στην βάση και η αποστολή του μηνύματος εισόδου (έγκριση εισόδου, απόρριψη εισόδου) γινόταν με μία διάρκεια της τάξεως των μερικών δευτερολέπτων (3-4 δευτερόλεπτα). Το γεγονός αυτό ήταν το σημαντικότερο για την απόφαση ο server του συστήματος να στηθεί σε έναν υπολογιστή με περισσότερες δυνατότητες απ' αυτές ενός RPI.

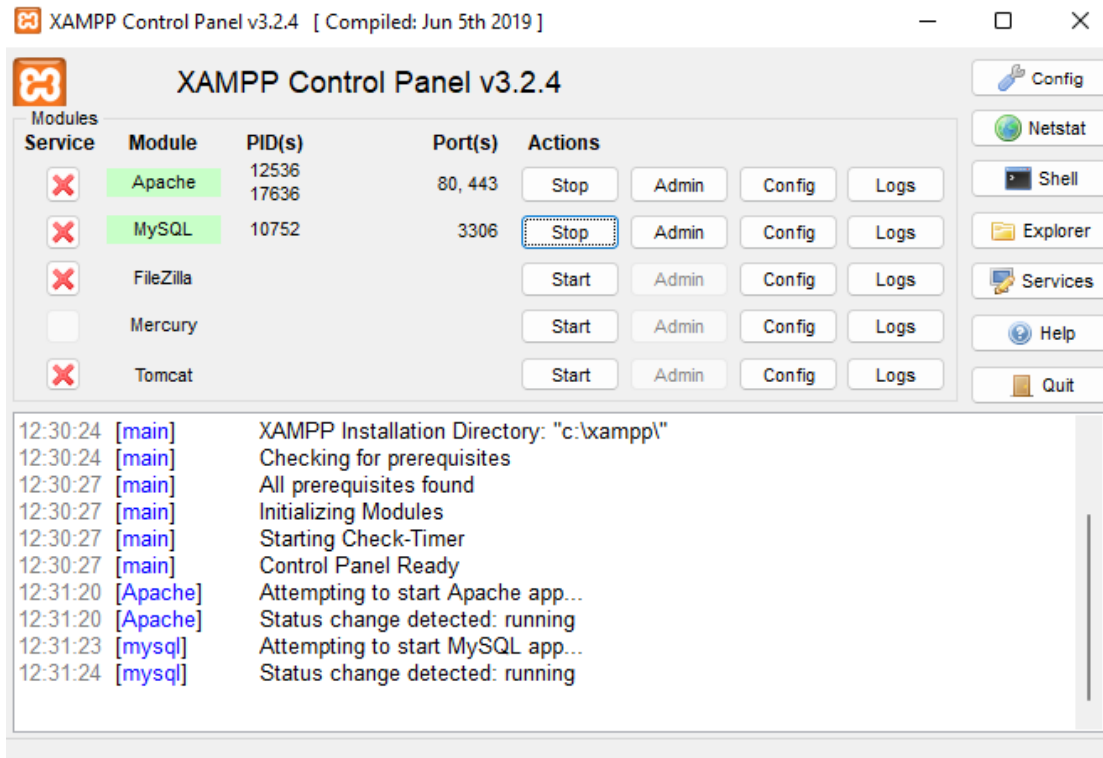
6.2 Δημιουργία Server

Το πρώτο βήμα για την δημιουργία του Server είναι η εγκατάσταση ενός MQTT broker με τον οποίο υποστηρίζεται η αποστολή και λήψη μηνυμάτων από τα MQTT topics. Για MQTT broker προτιμήθηκε ο Mosquitto, αρκετά διαδεδομένος και με εύκολη εγκατάσταση. Το debugging του πραγματοποιήθηκε με την εφαρμογή MQTT Explorer, μία εφαρμογή η οποία συνδέεται στον MQTT broker και εμφανίζει τα μηνύματα και τα topics του συστήματος κατά την εκτέλεση του συστήματος.



6.2-1 Η εφαρμογή MQTT Explorer

Για την δημιουργία Apache και MYSQL χρησιμοποιήθηκε η εφαρμογή XAMPP, μία δωρεάν πλατφόρμα η οποία προσφέρει τις παραπάνω λειτουργίες, δημιουργίας μία τοπική βάσης δεδομένων και έναν τοπικό Apache με εύκολη εγκατάσταση.

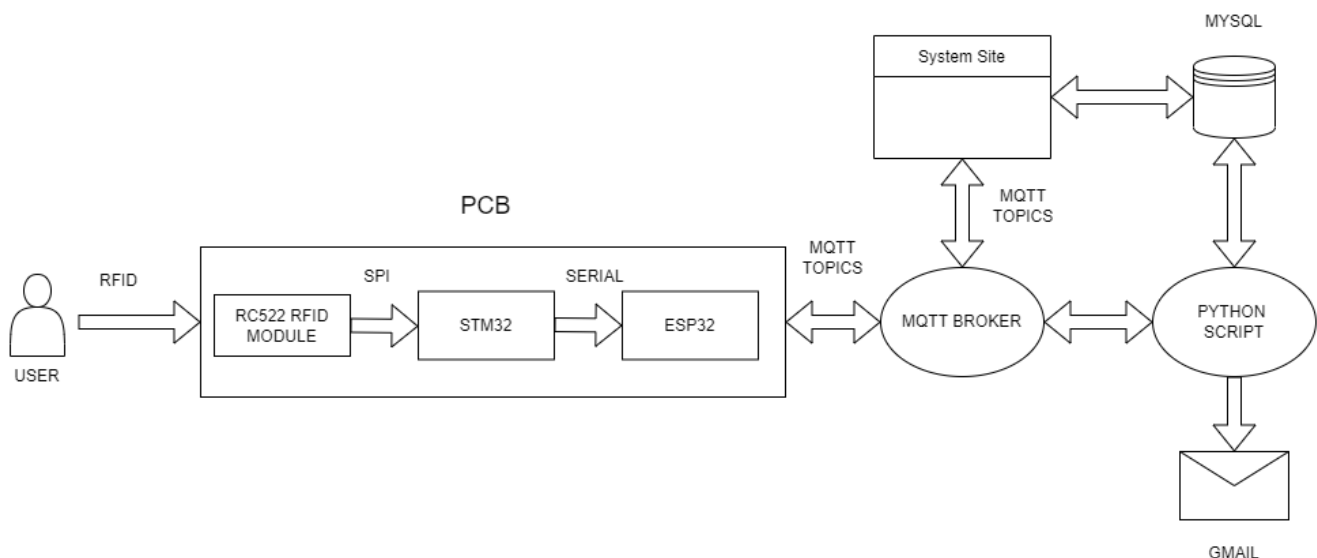


6.2-2 Η εφαρμογή XAMPP

Για την εκτέλεση του Python script, το μόνο που χρειάζεται είναι να είναι εγκατεστημένη κάποια έκδοση της Python 3.

Κεφάλαιο 7ο: Μοντέλο του συστήματος

Το μοντέλο επικοινωνίας ανάμεσα σε διάφορα μέρη του συστήματος παρουσιάζεται στο παρακάτω διάγραμμα.



Εικόνα 7-1 Το μοντέλο του συστήματος

Κεφάλαιο 8^ο: Μελλοντικά σχέδια-Βελτιώσεις

Το αποτέλεσμα της υλοποίησης παρουσιάζει ένα βασικό σύστημα παρακολούθησης ωραρίου εργαζομένων. Οι βελτιώσεις των δυνατοτήτων που μπορεί να προσφέρει το σύστημα επεκτείνονται στους παρακάτω τομείς:

- Τρόπος ελέγχου εισόδου-εξόδου των εργαζομένων
- Πλήθος τερματικών
- Δυνατότητες της σελίδας

Όσον αφορά τον έλεγχο εισόδου-εξόδου των εργαζομένων, υπάρχουν πιο προηγμένοι τρόποι αναγνώρισης του εργαζομένου εκτός από ένα RFID βραχιόλι. Οι τρόποι αυτοί είναι το δαχτυλικό αποτύπωμα, όπου ο εργαζόμενος τοποθετεί το δάχτυλο του στο τερματικό με στόχο την είσοδο και έξοδο από το κτίριο. Για την λειτουργία αυτή, απαραίτητη προϋπόθεση είναι η χρησιμοποίηση ενός αισθητήρα δαχτυλικού αποτυπώματος (fingerprint sensor).

Άλλος τρόπος επαλήθευσης της ταυτότητας του εργαζομένου είναι η αναγνώριση προσώπου. Ο τρόπος αυτός απαιτεί την χρήση κάμερας στο τερματικό της επιχείρησης. Όπως είναι αναμενόμενο, οι τρόποι αυτοί προσφέρουν πιο προηγμένο τρόπο αναγνώρισης των εργαζομένων, αλλά αυξάνουν και το κόστος του συστήματος.

Όσον αφορά το πλήθος των τερματικών, είναι προφανές ότι σε αρκετά μεγάλες επιχειρήσεις η χρήση ενός τερματικού δεν είναι η καλύτερη λύση. Σε περίπτωση που η εταιρεία αποτελείται από αρκετά γραφεία (εργαζόμενοι χωρισμένοι σε γραφεία μικρότερων ομάδων), η τοποθέτηση πολλαπλών τερματικών είναι η λύση για τον καλύτερο έλεγχο των εργαζομένων. Το γεγονός αυτό οδηγεί στην δυνατότητα που μπορεί να προστεθεί στο σύστημα, με το κάθε βραχιόλι να έχει την δυνατότητα εισόδου σε συγκεκριμένα μόνο σημεία της επιχείρησης (γραφείο του εργαζομένου, κουζίνα επιχείρησης).

Επιπλέον, η προσθήκη αυτή επιφέρει περισσότερες λεπτομέρειες όσον αφορά τον χρόνο εργασίας. Για παράδειγμα, με ένα τερματικό εισόδου-εξόδου η μόνη πληροφορία που μπορεί να επεξεργαστεί είναι η ώρα εισόδου και η ώρα εξόδου. Αντίθετα όμως, με την προσθήκη τερματικών σε διάφορα σημεία της εταιρείας, οι πληροφορίες που μπορούν να αντληθούν είναι το πόσο διάστημα ο εργαζόμενος έκατσε σε συγκεκριμένους χώρους της εταιρείας (πχ. ένας εργαζόμενος ενώ η ώρα εισόδου και εξόδου είναι η προβλεπόμενη, η ώρα που έκατσε στην τραπεζαρία της εταιρείας είναι αρκετά μεγάλη).

Κεφάλαιο 9^ο: Συμπεράσματα

Η παρούσα διπλωματική παρουσιάζει ένα μοντέλο υλοποίησης ενός συστήματος ωρομέτρησης ωραρίου εργαζομένων. Συγκριτικά με άλλα συστήματα που υπάρχουν στην αγορά, πρόκειται για ένα βασικό πακέτο που απευθύνεται σε μικρομεσαίες επιχειρήσεις.

Αναλυτικότερα, στην παρούσα διπλωματική παρουσιάζεται ένα σύστημα ωρομέτρησης ωραρίου εργαζομένων το οποίο προσφέρει:

- Έλεγχο εισόδου-εξόδου εργαζομένων μέσω σκαναρίσματος RFID
- Ιστορικό εργαζομένων με πληροφορίες όπως σύνολο ωρών εργασίας των εργαζομένων, υπερωρίες κ.λ.π
- Δυνατότητα σύνδεσης μαγνητικής κλειδαριάς με το τερματικό για είσοδο μόνο σε εργαζόμενους της εταιρείας

Επιπλέον, η απόφαση του συστήματος να στηρίζεται πάνω στο πρωτόκολλο MQTT το κάνει αρκετά εύκολο στην προσθήκη επιπλέον δυνατοτήτων.

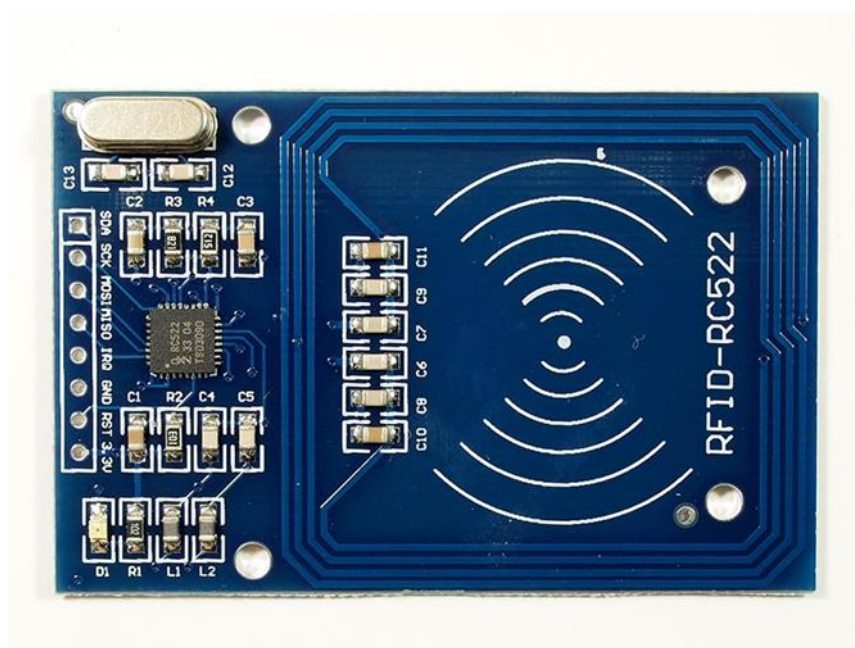
Κεφάλαιο 10^ο: Datasheet

10.1 RFID RC522 Module

Για το σκανάρισμα των καρτών RFID χρησιμοποιήθηκε το RC522 RFID Module. Πρόκειται για ένα αρκετά διαδεδομένο module το οποίο υποστηρίζει I2C, SPI και UART. Βασίζεται στον MFRC522 controller της NXP semiconductors. Δέχεται κάρτες RFID 13.56 MHz και η τάση λειτουργίας του είναι τα 3.3V. Συνήθως χρησιμοποιείται σε εφαρμογές όπου ο χρήστης πρέπει να ταυτοποιηθεί σε ένα σύστημα με ένα μοναδικό ID.

Οι κάρτες RFID έχουν 1kB μνήμη, η οποία μπορεί να χρησιμοποιηθεί για αποθήκευση δεδομένων. Ο RC522 reader είναι σε θέση και να διαβάσει αλλά και να γράψει την μνήμη αυτή.

Τις περισσότερες φορές, προτιμάται το πρωτόκολλο SPI καθώς προσφέρει ταχύτητα 10Mbps.



Εικόνα 10.1-1 RC522 RFID Card Reader Module

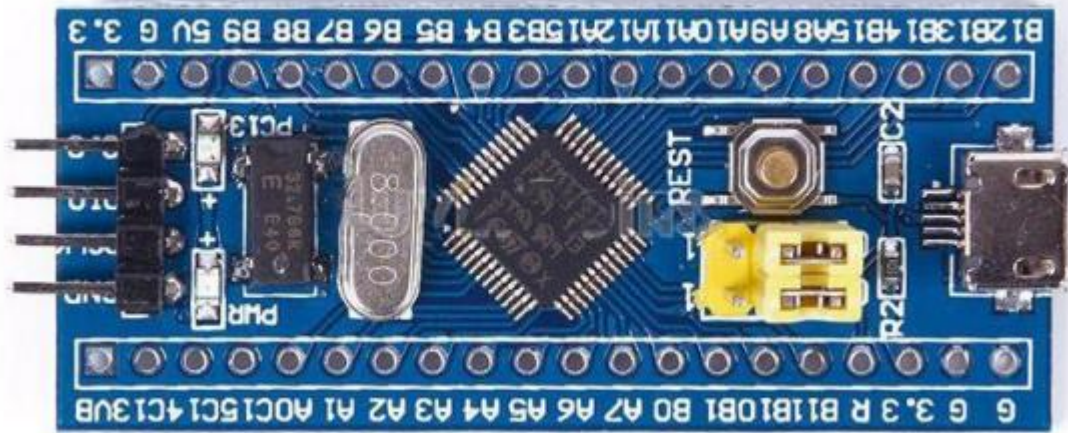
10.2 STM32F103C8

Η σειρά μικροελεγκτών STM32F103xx ανήκει στη σειρά ARM Cortex-M3 στα 32 bit με συχνότητας λειτουργίας τα 72 MHz. Έχουν flash memory 128 Kbytes και SRAM 20 Kbytes. Όλη η σειρά προσφέρει 2 12-bit ADCs, 3 general purpose 16-bit timers, έναν PWM timer καθώς και πρωτόκολλα επικοινωνίας:

- 2 I2Cs
- 2 SPIs
- 3 USARTSs
- 1 USB
- 1 CAN

Η τάση λειτουργίας του είναι από 2 έως 3.6V και η θερμοκρασία λειτουργίας από -40 έως +85°.

Η ST προσφέρει για την σειρά STM32F103 την αναπτυξιακή πλακέτα με μικροελεγκτή τον STM32F103C8T6 (γνωστή κι ως blue pill), η οποία χρησιμοποιήθηκε κατά την διάρκεια της δημιουργίας της διπλωματικής.



Εικόνα 10.2-1 Η αναπτυξιακή πλακέτα blue pill (STM32F103C8T6)

BIBΛΙΟΓΡΑΦΙΑ

[1] http://www.easycomtech.gr/orometrisiprosopikou/?gclid=CjwKCAjwgr6TBhAGEiwA3aVuIXNHN4QdLU4H6JcV_yykRFZ1S3KjIECMuGe1XxdAZWzXZtqp7-0ZuhoCiCEQAvD_BwE

[2] https://gqt.gr/el/xipblog/post/53_access-control-charaktiristika.html?page_type=post

[3] <https://zarifopoulos.com/access-control-systems/>

[4] <https://components101.com/wireless/rc522-rfid-module>

[5] <https://www.novatron.gr/el/plastic-card-solutions/elegxos-prosvasis-access-control/zk-acc-skw-h2-reader-controller>

[6] <https://www.novatron.gr/el/zk-sc-103>

[7] <https://www.novatron.gr/el/plastic-card-solutions/elegxos-prosvasis-access-control/zk-tf1700>

[8] <https://www.st.com/en/microcontrollers-microprocessors/stm32f103c8.html>

DATASHEET

[1] <https://www.nxp.com/docs/en/data-sheet/MFRC522.pdf>

[2] <https://www.st.com/resource/en/datasheet/stm32f103c8.pdf>

Κώδικες του συστήματος

Python Codes

mainControl.py (the main program for back-end)

```
from os import read
import re
import paho.mqtt.client as mqtt #import the client1
import time
import serial.tools.list_ports as port_list #port finder
import serial
from serial import SerialException
from serial.tools import list_ports
import configparser
from portFinder import *
import asyncio
from mqttConnection import *
import schedule

global arduinoConnected
arduinoConnected=False

##mqtt configuration
schedule,client=MQTTconfig()

async def ArduinoConnection():
    global ser
    try:
        port= await findCom()
        print(f"Port:{port}")
        ser= await makeConnection(port)
    except Exception as e:
```

```

        print("Serial communication failed,28 error {e}, trying to
connect")
        await asyncio.sleep(1)
    while True:
        try:
            port=await findCom()
            ser=await makeConnection(port)
            break
        except Exception as e:
            print(f"Can't create connection,36 error {e}")
            await asyncio.sleep(1)

#thread ksexorista
#test
async def Serial():
    global arduinoConnected
    global ser
    await ArduinoConnection()
    while True:
        schedule.run_pending()
        try:
            if ser.is_open and arduinoConnected==False:
                arduinoConnected=True
            if ser.in_waiting>0:
                read_serial=ser.readline().decode('utf-8').rstrip()
                print(f"Message from Arduino: {read_serial}")
                msg = {
                    "rfidNumber" : read_serial
                }
                msg= json.dumps(msg)
                client.publish(f"rfidScanner", f"{msg}", qos=2)

```

```

except Exception as e:
    arduinoConnected=False
    print(f"Serial communication stopped,61 error {e}")
    print(f"Serial state: {ser.is_open}")
    await asyncio.sleep(1)
    if ser.is_open:
        print("Ser is closing...")
        ser.close()
    print("Trying to reconnect...")
    port=await findCom()
    ser.port=port
    try:
        ser.open()
        print("Success...Arduino is now connected")
    except Exception as e:
        print(f"Can't create connection,74 error {e}")
        await asyncio.sleep(1)

```

```

loop=asyncio.get_event_loop()
loop.set_debug(True) # Enable debug
loop.create_task(Serial())
loop.run_forever()

# if __name__ == '__main__':
#     asyncio.run(Serial(), debug=True)

```

mqttConnection.py (the sub-file for back-end, controls the mqtt)

```
import paho.mqtt.client as mqtt #import the client1
import time
import asyncio
import json
#prosthikes
import mysql.connector
import datetime
import schedule
#mail
import smtplib, ssl
from email.mime.text import MIMEText
from email.mime.multipart import MIMEMultipart

global messageFlag
messageFlag=False

host='192.168.100.4'
username="mqtt"
password="zagkoulis"

datetimeFormat = '%Y-%m-%d %H:%M:%S.%f'
date=datetime.datetime.now().date()
date=date.strftime('%Y-%m-%d')
startTime='09:00:00.0'
stopTime='17:00:00.0'
print(startTime)

def on_message(client, userdata, message):
    print(f"Message received: {str(message.payload.decode('utf-8'))},
topic: {message.topic} , qos: {message.qos} , retain flag:
{message.retain} ")
```

```

    checkMessage(message.payload.decode('utf-8'))

def on_connect(client, userdata, flags, rc):
    if rc==0:
        client.connected_flag=True #set flag
        print (f"Client connected with the Broker with code: {rc}.")
        client.bad_connection_flag=False
        client.subscribe("DoorRfid")

    else:
        print(f"Bad connection Returned code: {rc}")
        client.bad_connection_flag=True
        #on_disconnect()

def on_disconnect(client, userdata, rc):
    print("Client disconnected")
    client.connected_flag=False

#####
def MQTTconfig():
    global client

    # shcedule
    schedule.every().minute.at(":30").do(dailyCheck)

    mqtt.Client.connected_flag=False
    mqtt.Client.bad_connection_flag=False
    print("Creating new instance")
    client = mqtt.Client("rfidScanner") #create new instance
    #####

```

```

client.on_message=on_message #attach functions to callback
client.on_disconnect = on_disconnect
client.on_connect=on_connect
#####
client.loop_start()
client.username_pw_set(username=username,password=password)
print(f"Connecting to broker:{host}")

try:
    client.connect(host) #connect to broker
except Exception as e:
    print(f"71.Connection failed. Error {e}")
    while True:
        print("Trying to connect...")
        try:
            client.connect(host)
            break
        except Exception as e:
            print(f"Connection failed. Error {e}")

while not client.connected_flag: #wait in loop
    print("Waiting for client connected flag")
    time.sleep(1)

if client.bad_connection_flag:
    while client.bad.connected_flag:
        print("Trying to connect. bad connected flag")
        try:
            client.connect(host)
        except:
            print("Can't connect")

```

```

        print(f"Connected Flag:{client.connected_flag}")
        return schedule,client
####MQTTT

def checkDatabase(command):
    mydb = mysql.connector.connect(
        host="localhost",
        database="rfid_system",
        user="root",
        password=""
    )
    mycursor = mydb.cursor()
    mycursor.execute(command)
    return mydb,mycursor

def insertDatabase(command,values):
    mydb = mysql.connector.connect(
        host="localhost",
        database="rfid_system",
        user="root",
        password=""
    )
    mycursor = mydb.cursor()
    mycursor.execute(command,values)
    mydb.commit()
    print(mycursor.rowcount,"record inserted")

def deleteDatabase(command):
    mydb = mysql.connector.connect(
        host="localhost",
        database="rfid_system",

```

```

user="root",
password=""
)
mycursor = mydb.cursor()
mycursor.execute(command)
mydb.commit()
print(mycursor.rowcount,"record deleted")

def checkMessage(msg):
    global rfidNumber
    msg=json.loads(msg)
    rfidNumber=msg['rfidNumber']
    print(f"The rfid Number is: {rfidNumber}")
    print(type(rfidNumber))
    mydb,mycursor=checkDatabase(f"SELECT * FROM attendance WHERE
cardID='{rfidNumber}'")
    print("Searching in database what is this rfid....")
    results=mycursor.fetchone()
    if results:
        rfidNumber=results[0]
        timeCame=results[1]
        client.publish("DoorAccess", "Leaving", qos=2)
        print("Member came out of the building")
        print(f"RfidNumber:{rfidNumber}")
        print(f"TimeCame:{timeCame}")
        print(type(timeCame))
        timeLeft=datetime.datetime.now() #etsi tha to pairnei
kanonika
        # timeLeft=msg['timeLeft'] #manual
        # timeLeft=datetime.datetime.strptime(timeLeft,
datetimeFormat) #ki auto gia manual
        # print(type(timeLeft))

```

```

        mydb,mycursor=checkDatabase(f"SELECT name FROM members WHERE
cardID='{rfidNumber}'")
        print(f"TimeLeft:{timeLeft}")
        results=mycursor.fetchone()
        #print(f"results:{results[0]}")
        msgmqtt={
            "message":f"Goodbye {results[0]} . Exodus:
{timeLeft}"
        }
        msgmqtt= json.dumps(msgmqtt)
        client.publish("screenMessage", msgmqtt, qos=2)

        insertDatabase(f"DELETE FROM attendance WHERE cardID=
%s",(rfidNumber,))
        date=datetime.datetime.now().date()
        date=date.strftime('%Y-%m-%d')
        startTime=date+' 09:00:00.0'
        stopTime=date+' 17:00:00.0'
        print(f"start Time: {startTime}")
        print(f"stopTime {stopTime}")
        print(f"timeCame{timeCame}")
        print(f"timeLeft{timeLeft}")
        if timeCame>datetime.datetime.strptime(startTime,
datetimeFormat):
            print("Your come Late!")
            comeLate=timeCame-datetime.datetime.strptime(startTime,
datetimeFormat)
            print(f"comeLate:{comeLate}")
            comeLate=comeLate.seconds
            comeLate=comeLate/60
            comeLate=round(comeLate)
            print(f"comeLate{comeLate}")

        else:

```

```

        print("You come Early")
        comeLate=datetime.datetime.strptime(startTime,
datetimeFormat)-timeCame
        print(f"comeLate:{comeLate.seconds}")
        comeLate=comeLate.seconds
        comeLate=comeLate/60
        comeLate=-round(comeLate)
        print(f"comeLate{comeLate}")

    if        timeLeft>datetime.datetime.strptime(stopTime,
datetimeFormat):
        print("You leave Late!")
        leftEarly=timeLeft-datetime.datetime.strptime(stopTime,
datetimeFormat)
        print(f"leftEarly:{leftEarly}")
        leftEarly=leftEarly.seconds
        leftEarly=leftEarly/60
        leftEarly=round(leftEarly)
        print(f"leftEarly:{leftEarly}")

    else:
        print("You leave Early")
        leftEarly=datetime.datetime.strptime(stopTime,
datetimeFormat)-timeLeft
        print(f"leftEarly:{leftEarly}")
        leftEarly=leftEarly.seconds
        leftEarly=leftEarly/60
        leftEarly=-round(leftEarly)
        print(f"leftEarly:{leftEarly}")

    #
    print(f"comelate{str(datetime.timedelta(seconds=comeLate.seconds))}"
)

```

```

#
print(f"leftearly{str(datetime.timedelta(seconds=leftEarly.seconds))
}")

    hours=timeLeft-timeCame

    # hours=str(datetime.timedelta(seconds=hours.seconds))

print(f"HOURS:{str(datetime.timedelta(seconds=hours.seconds))}")

    overtime=hours.seconds-28800
    overtime=overtime/60
    overtime=round(overtime)
    print(f"overtime:{overtime}")

    # hours=datetime.timedelta() ##des
    print("hours:%s" % hours)

        insertDatabase(f"INSERT          INTO          dailyreport
(cardID,arrival,exodus,comeLate,leftEarly,hoursWorked,overtime)
VALUES(%s,%s,%s,%s,%s,%s,%s)",(rfidNumber,timeCame,timeLeft,comeLate
,leftEarly,hours,overtime))

else:
    print("Not found ====> Member came in the building")
    date=time.strftime('%Y-%m-%d %H:%M:%S') #kanonika etsi tha to
pairnei
    # date=msg['timeCame'] #gia manual
    mydb,mycursor=checkDatabase(f"SELECT name FROM members WHERE
cardID='{rfidNumber}'")
    results=mycursor.fetchone()
    if results:
        client.publish("DoorAccess", "Authorized", qos=2)
        print(f"results:{results[0]}")
        msgmqtt={
            "message":f"Welcome {results[0]} .Have a nice
day. Arrival: {date}"
        }
        msgmqtt= json.dumps(msgmqtt)

```

```

        client.publish("screenMessage", msgmqtt, qos=2)
        insertDatabase(f"INSERT INTO attendance (cardID,arrival)
VALUES (%s,%s)",(rfidNumber,date))
    else:
        client.publish("DoorAccess", "NotAuthorized", qos=2)
        print("Unknown rfid number. Entrance not authorized")

def dailyCheck():
    print("====Time for daily Report====")
    msg=''
    mydb,mycursor=checkDatabase(f"SELECT * FROM dailyreport WHERE 1")
    print("Searching in daily report what is this rfid....")
    results=mycursor.fetchall()
    # sendMail(results)
    for result in results:
        mydb,mycursor=checkDatabase(f"SELECT      name,surname      FROM
members WHERE cardID='{result[0]}'")
        personal=mycursor.fetchone()
        print(f"personal {personal}")
        print(f"Name:{personal[0]}  Surname:{personal[1]}  CardID:
{result[0]}  Arrival: {result[1]} Exodus: {result[2]} \n ")
        insertDatabase(f"INSERT          INTO          history
(cardID,arrival,exodus,comeLate,leftEarly,hoursWorked,overtime)
VALUES(%s,%s,%s,%s,%s,%s,%s)",(result[0],result[1],result[2],result[
3],result[4],result[5],result[6]))
        msg=msg+(f"Name:{personal[0]}  Surname:{personal[1]}  CardID:
{result[0]}  Arrival: {result[1]} Exodus: {result[2]} \n")
        deleteDatabase(f"DELETE FROM dailyreport WHERE 1")
        deleteDatabase(f"DELETE FROM attendance WHERE 1")
    print("Message")
    print(msg)
    if len(msg)==0:
        print("Message is empty, mail will not be sent")
    else:

```

```
sendMail(msg)
```

```
def sendMail(msg):
    sender_email = "zagkoulisnikos512034@gmail.com"
    receiver_email = "zagkoulisnikos512034@gmail.com"
    password = '6976397888Aa'
    message = MIMEMultipart("alternative")
    message["Subject"] = "Daily Report"
    message["From"] = sender_email
    message["To"] = receiver_email
    # Create the plain-text and HTML version of your message
    text = """\
---Daily Report---
Intelligent Entertainment"""\
    # html = """\
# <html>
# <body>
#     <p>---Daily Report---

#     </p>
# </body>
# </html>
# """\
    # Turn these into plain/html MIMEText objects
    part1 = MIMEText(msg, "plain")
    # part2 = MIMEText(html, "html")
    # Add HTML/plain-text parts to MIMEMultipart message
    # The email client will try to render the last part first
    message.attach(part1)
    #message.attach(part2)
    # Create secure connection with server and send email
```

```

context = ssl.create_default_context()
with smtplib.SMTP_SSL("smtp.gmail.com", 465, context=context) as
server:
    server.login(sender_email, password)
    server.sendmail(
        sender_email, receiver_email, message.as_string()
    )

```

portFinder.py (the program that scans the ports looking for Arduino)

```

from logging import Logger
from os import read
import re
import paho.mqtt.client as mqtt #import the client1
import time
import serial.tools.list_ports as port_list #port finder
import serial
from serial import SerialException
from serial.tools import list_ports
import configparser
import asyncio
from threading import Timer

port="Unknown"
async def findCom():
    while True:
        try:
            ports = list(port_list.comports())
            for port in ports:
                if port.manufacturer=="FTDI":
                    print(f"Arduino detected on com:{port.name}")

```

```

        return port.name
    Arduino = next(list_ports.grep("Nano"))
    print(f"Arduino detected on com: {Arduino.name} ")
    return Arduino.name
except Exception as e:
    try:
        Arduino = next(list_ports.grep("CH340"))
        print(f"Arduino detected on com: {Arduino.name} ")
        return Arduino.name
    except Exception as e:
        try:
            Arduino = next(list_ports.grep("ACM0"))
            Arduino.name='/dev/'+Arduino.name
            print(f"Arduino detected on com: {Arduino.name}
")

            return Arduino.name
        except Exception as e:
            try:
                Arduino = next(list_ports.grep("USB0"))
                Arduino.name='/dev/'+Arduino.name
                print(f"Arduino detected on com:
{Arduino.name} ")

                return Arduino.name
            except Exception as e:
                try:
                    Arduino = next(list_ports.grep("USB1"))
                    Arduino.name='/dev/'+Arduino.name
                    print(f"Arduino detected on com:
{Arduino.name} ")

                    return Arduino.name
                except:
                    try:

```

```

        Arduino
    next(list_ports.grep("ACM1"))
        Arduino.name='/dev/'+Arduino.name
        print(f"Arduino detected on com:
{Arduino.name} ")
        return Arduino.name
    except:
        await asyncio.sleep(1)

async def makeConnection(port):
    global ser
    while True:
        try:
            ser=serial.Serial(port,baudrate=9600,timeout=0.1)
            print("-----Serial communication started-----")
            return ser
        except Exception as e:
            time.sleep(1)
            print("Trying to connect...")
            port= await findCom()
            try:
                ser=serial.Serial(port,baudrate=9600,timeout=1)
                print("Success...Arduino is now connected")
                return ser
            except Exception as e:
                await asyncio.sleep(1)

```

Arduino Codes

Mqtt.ino (the code for ESP32)

```
#include <ArduinoJson.h>
#include <WiFi.h>
#include <PubSubClient.h>

const char* ssid = "-----";
const char* password = "-----";
const char* mqttServer = "-----";
const int mqttPort = 1883;
const char* mqttUser = "mqtt";
const char* mqttPassword = "-----";
int a=0;
WiFiClient espClient;
PubSubClient client(espClient);

void setup()
{

  Serial.begin(9600);
  delay(3000);
  Serial2.begin(9600);
  delay(500);
  Serial.println();

  WiFi.begin(ssid, password);

  while (WiFi.status() != WL_CONNECTED)
  {
    delay(500);
    Serial.println("Connecting to WiFi..");
  }
}
```

```

Serial.println("Connected to the WiFi network");

client.setServer(mqttServer, mqttPort);
client.setCallback(callback);

while (!client.connected())
{
  Serial.println("Connecting to MQTT...");
  if (client.connect("ESP32Client", mqttUser, mqttPassword ))
  {
    client.subscribe("DoorAccess");
    Serial.println("connected");
  }
  else
  {
    Serial.print("failed with state ");
    Serial.print(client.state());
    delay(2000);
  }
}

void loop()
{
  if (Serial2.available())
  {
    String rfid = Serial2.readString();
    Serial.println(rfid);
    sendMqttMessage(rfid);
  }
  client.loop();
}

```

```
}
```

```
void sendMqttMessage(String rfidNumber)
{
    StaticJsonDocument<300> jb;
    jb["rfidNumber"] = rfidNumber;
    char JSONmessageBuffer[100];
    serializeJson(jb,JSONmessageBuffer);
    Serial.println("Sending message to MQTT topic..");
    Serial.println(JSONmessageBuffer);

    if (client.publish("DoorRfid", JSONmessageBuffer) == true)
    {
        Serial.println("Success");
    }
    else
    {
        Serial.println("Error");
    }
}
```

```
void callback(char* topic, byte* message, unsigned int length) {
    Serial.print("Message arrived on topic: ");
    Serial.print(topic);
    Serial.print(". Message: ");
    String messageTemp;

    for (int i = 0; i < length; i++) {
        Serial.print((char)message[i]);
        messageTemp += (char)message[i];
    }
}
```

```

}
Serial.println();
int a = 0;

if (String(topic) == "DoorAccess") {
  Serial.println("Door Access");
  if(messageTemp == "Authorized"){
    Serial.println("Authorized");
    a=1;
    Serial2.print(a);
  }
  else if(messageTemp == "NotAuthorized")
  {
    a=2;
    Serial2.print(a);
    Serial.println("Not Authorized");
  }
  else if(messageTemp == "Leaving")
  {
    a=3;
    Serial2.print(a);
    Serial.println("Leaving");
  }
  else
  {
    a=0;
    Serial2.print(a);
    Serial.println("Unknown message");
  }
}
}
}

```

readUID.ino (the code for Arduino scanning the ID of wristbands)

```
/*
 * -----
 * -----
 * Example sketch/program showing how to read new NUID from a PICC to
 serial.
 * -----
 * -----
 * This is a MFRC522 library example; for further details and other
 examples see: https://github.com/miguelbalboa/rfid
 *
 * Example sketch/program showing how to the read data from a PICC
 (that is: a RFID Tag or Card) using a MFRC522 based RFID
 * Reader on the Arduino SPI interface.
 *
 * When the Arduino and the MFRC522 module are connected (see the pin
 layout below), load this sketch into Arduino IDE
 * then verify/compile and upload it. To see the output: use Tools,
 Serial Monitor of the IDE (hit Ctrl+Shft+M). When
 * you present a PICC (that is: a RFID Tag or Card) at reading distance
 of the MFRC522 Reader/PCD, the serial output
 * will show the type, and the NUID if a new card has been detected.
 Note: you may see "Timeout in communication" messages
 * when removing the PICC from reading distance too early.
 *
 * @license Released into the public domain.
 *
 * Typical pin layout used:
 * -----
 *
 *          MFRC522      Arduino      Arduino      Arduino      Arduino
 Arduino
 *          Reader/PCD   Uno/101        Mega          Nano v3
 Leonardo/Micro  Pro Micro
 * Signal      Pin          Pin          Pin          Pin          Pin
 Pin
```

```

* -----
-----
* RST/Reset  RST      9          5          D9          RESET/ICSP-
5    RST
* SPI SS      SDA(SS)   10          53          D10          10
10
* SPI MOSI    MOSI      11 / ICSP-4  51          D11          ICSP-4
16
* SPI MISO    MISO      12 / ICSP-1  50          D12          ICSP-1
14
* SPI SCK     SCK       13 / ICSP-3  52          D13          ICSP-3
15
*/

```

```
#include <SPI.h>
```

```
#include <MFRC522.h>
```

```
#define SS_PIN 10
```

```
#define RST_PIN 9
```

```
MFRC522 rfid(SS_PIN, RST_PIN); // Instance of the class
```

```
MFRC522::MIFARE_Key key;
```

```
#define buzzer 2
```

```
// Init array that will store new NUID
```

```
//byte nuidPICC[4];
```

```
void setup()
```

```
{
```

```
  Serial.begin(9600);
```

```
  SPI.begin(); // Init SPI bus
```

```
  rfid.PCD_Init(); // Init MFRC522
```

```
  pinMode(buzzer,OUTPUT);
```

```
  for (byte i = 0; i < 6; i++)
```

```

    {
        key.keyByte[i] = 0xFF;
    }
}

void loop() {

    // Reset the loop if no new card present on the sensor/reader. This
    // saves the entire process when idle.
    if ( ! rfid.PICC_IsNewCardPresent())
        return;

    // Verify if the NUID has been readed
    if ( ! rfid.PICC_ReadCardSerial())
        return;

    MFRC522::PICC_Type piccType = rfid.PICC_GetType(rfid.uid.sak);

    // Check is the PICC of Classic MIFARE type
    if (piccType != MFRC522::PICC_TYPE_MIFARE_MINI &&
        piccType != MFRC522::PICC_TYPE_MIFARE_1K &&
        piccType != MFRC522::PICC_TYPE_MIFARE_4K)
    {
        return;
    }

    tone(buzzer, 1000); // Send 1KHz sound signal...
    delay(500);        // ...for 1 sec
    noTone(buzzer);   // Stop sound...
    printHex(rfid.uid.uidByte, rfid.uid.size);
    Serial.println();
}

```

```

    rfid.PICC_HaltA();
    rfid.PCD_StopCrypto1();
}

/**
 * Helper routine to dump a byte array as hex values to Serial.
 */
void printHex(byte *buffer, byte bufferSize)
{
    for (byte i = 0; i < bufferSize; i++)
    {
        Serial.print(buffer[i] < 0x10 ? " 0" : "");
        Serial.print(buffer[i], HEX);
    }
}

/**
 * Helper routine to dump a byte array as dec values to Serial.
 */
void printDec(byte *buffer, byte bufferSize)
{
    for (byte i = 0; i < bufferSize; i++)
    {
        Serial.print(buffer[i] < 0x10 ? " 0" : " ");
        Serial.print(buffer[i], DEC);
    }
}

```

CSS codes

Index.css (css code for index page)

```
/* ===== Scrollbar CSS ===== */
/* Firefox */
* {
  scrollbar-width: auto;
  scrollbar-color: #86aff0 white;
}

/* Chrome, Edge, and Safari */
::-webkit-scrollbar {
  width: 16px;
}

::-webkit-scrollbar-track {
  background: white;
}

::-webkit-scrollbar-thumb {
  background-color: #86aff0;
  border-radius: 10px;
  border: 3px solid #ffffff;
}

body {
  margin: 0;
  padding: 0;
  width:100vw;
  height: 100vh;
  background-color:#86aff0;
  font-family: "Lucida Console", Courier, monospace;
```

```
}
```

```
.content {  
  z-index: 1;  
  display: grid;  
  display: grid;  
  grid-template-columns: 0.2fr 0.9fr 1.9fr 1.4fr 0.6fr;  
  grid-template-rows: 0.6fr 3.1fr 0.3fr 0.2fr;  
  gap: 0px 0px;  
  grid-template-areas:  
    ". . message message ."  
    ". sidebar main main ."  
    ". welcome info clock clock"  
    ". . . . .";  
  width: 100%;  
  height: 100%;  
}
```

```
.message  
{  
  grid-area: message;  
  margin: auto;  
  font-size: 30px;  
}
```

```
.sidebar{  
  grid-area: sidebar;  
  border-radius: 5px;  
  padding: 20px;  
  /* background-color: #212529; */
```

```

background-color:#f2f8ff;
/* background: url("sidebar.jpg"); */
color:aliceblue;
-webkit-box-shadow: -3px 6px 17px 2px rgba(0,0,0,0.71);
box-shadow: -3px 6px 17px 2px rgba(0,0,0,0.71);
}

.welcome
{
  grid-area: welcome;
  margin: auto;
  /* margin-bottom: 40px; */
  color:white;
}

.clock {
  grid-area: clock;
  color:white ;
  font-size: 30px;
  /* font-family: 'Open Sans','TwemojiMozilla'; */
  /* font-family: 'Open Sans','TwemojiMozilla'; */
  font-family: "Lucida Console", Courier, monospace;

  letter-spacing: 5px;
  margin:auto;
  word-wrap: break-word;
}

.main{
  grid-area: main;
  text-align: center;
  /* margin-bottom: 40px; */
  background-color:white;

```

```
border-radius: 5px;
-webkit-box-shadow: 5px 5px 5px 5px rgba(0,0,0,0.45);
box-shadow: 5px 5px 5px 5px rgba(0,0,0,0.45);
overflow: scroll;
}
```

```
.members
{
text-align: center;
margin-bottom: 40px;
}
```

```
.register
{
text-align: center;
}
```

```
.logout
{
text-align: center;
}
```

```
.p-2
{
color: #4e73df;
}
```

```
.button {
background-color: transparent;
color: #86aff0 ;
margin-top: 10px;
padding: 15px 20px;
```

```

border: none;
cursor: pointer;
width: 100%;
opacity: 0.9;

}
.delete{
background-color: transparent;
color: black ;
/* margin-top: 10px; */
/* padding: 15px 20px; */
border: none;
cursor: pointer;
/* width: 100%; */
opacity: 0.9;
margin-right: 0px;
padding: 0px;

}
.delete:hover
{
color:white;
}
.deleteIcon
{
align-items: center;
}

.info{
background-color: transparent;
color: black ;
/* margin-top: 10px; */

```

```
/* padding: 15px 20px; */
border: none;
cursor: pointer;
/* width: 100%; */
opacity: 0.9;
margin-right: 0px;
padding: 0px;

}
.info:hover
{
    color:white;
}

.button:hover {
    color: black;
}
.buttonMain
{
    background-color: transparent;
    color: black ;
    margin-top: 10px;
    padding: 15px 20px;
    border: none;
    cursor: pointer;
    width: 100%;
```

```

    opacity: 0.9;

}
.buttonMain:hover
{
    color: black;

}
#next{
    margin-top: 10px;
}

/* LOADER */
.loader-wrapper {
    z-index: 100;
    width: 100%;
    height: 100%;
    position: absolute;
    top: 0;
    left: 0;
    background-color: #86aff0 ;
    display: flex;
    justify-content: center;
    align-items: center;
}
.loader {
    display: inline-block;
    width: 30px;
    height: 30px;
    position: relative;
    border: 4px solid #212529;
    animation: loader 2s infinite ease;

```

```
}  
.loader-inner {  
  vertical-align: top;  
  display: inline-block;  
  width: 100%;  
  background-color: #212529;  
  animation: loader-inner 2s infinite ease-in;  
}  
.home  
{  
  width: 30px;  
  height:30px;  
  margin: auto;  
}  
.members  
{  
  width: 30px;  
  height:30px;  
  margin: auto;  
}  
  
.register  
{  
  width: 30px;  
  height:30px;  
  margin: auto;  
}  
.logout  
{  
  width: 30px;  
  height:30px;  
  margin: auto;
```

```
}  
#output  
{  
  border-radius: 50%;  
  border: 5px solid #86aff0;  
  
}  
#output:hover  
{  
  transform: scale(1.5);  
}  
  
#last  
{  
  margin-bottom: 30px;  
}  
.picture  
{  
  margin-left: 40px;  
  float: left;  
  margin-bottom: 10px;  
  
}  
.picture:hover  
{  
  transform: scale(1.5);  
}  
  
.flex-container {  
  display: flex;  
  /* background-color: DodgerBlue; */  
  flex-wrap: wrap;
```

```
color: white;
flex-direction: row;
justify-content: space-around ;
align-items:center;
}
```

```
.flex-container > button {
background-color: #86aff0;
/* display: flex; */
margin: 10px;
padding: 20px;
font-size: 10px;
width: 200px;
height: 200px;
border-radius: 10%;
/* margin: auto; */
margin-top: 10px;
color: white;
}
```

```
.allmembersNumber
{
font-size: 40px;
padding-top: 40px;
}
```

```
.insideNumber
```

```
{
  font-size: 40px;
  padding-top: 40px;
}
```

```
.flex-member{
  background-color: white;
  display: flex;
  flex-direction: row;
  justify-content: space-between;
  align-items:center;
}
```

```
.list-group-item:hover
{
  background-color: #86aff0;
}
.flex-member:hover
{
  background-color: #86aff0;
  color: white;
}
```

```
.flex-member > div {
  margin: 5px;
  padding: 5px;
```

```

    /* font-size: 15px; */
}

@keyframes loader {

    0% { transform: rotate(0deg);}
    25% { transform: rotate(180deg);}
    50% { transform: rotate(180deg);}
    75% { transform: rotate(360deg);}
    100% { transform: rotate(360deg);}
}

@keyframes loader-inner {

    0% { height: 0%;}
    25% { height: 0%;}
    50% { height: 100%;}
    75% { height: 100%;}
    100% { height: 0%;}
}

/* LOADER */

/* The Modal (background) */
.modal {
    display: none; /* Hidden by default */
    position: fixed; /* Stay in place */
    z-index: 1; /* Sit on top */
    padding-top: 100px; /* Location of the box */
    left: 0;

```

```
top: 0;
width: 100%; /* Full width */
height: 100%; /* Full height */
overflow: auto; /* Enable scroll if needed */
background-color: rgb(0,0,0); /* Fallback color */
background-color: rgba(0,0,0,0.4); /* Black w/ opacity */
}
```

```
/* Modal Content */
.modal-content {
background-color: #fefefe;
margin: auto;
padding: 20px;
border: 1px solid #888;
width: 30%;
}
```

```
/* The Close Button */
.close {
color: #aaaaaa;
float: right;
font-size: 28px;
font-weight: bold;
}
```

```
.close:hover,
.close:focus {
color: #000;
text-decoration: none;
cursor: pointer;
}
```

Login.css (the css file for login page)

```
/* css for login page */
body
{
  font-family: "Lucida Console", Courier, monospace;
  background-color:#86aff0;
  width: 100%;
}
.container
{
  display: grid;
  grid-template-columns: 1fr 1fr 1fr;
  grid-template-rows: 1fr 1fr 0.3fr 0.7fr;
  gap: 0px 0px;
  grid-template-areas:
    "clock rfid ."
    ". loginBox ."
    ". message ."
    "movingText movingText start";
  width: 85vw;
  height: 85vh;
}
.loginBox
{
  grid-area: loginBox;
  border-radius: 5%;
  background-color:#ffffff ;
  -webkit-box-shadow: 5px 5px 18px 5px #000000;
  box-shadow: 5px 5px 18px 5px #000000;
}

.input-container
```

```

{
  display: flex;
  width: 80%;
  margin-bottom: 15px;
}
.rfid
{
  grid-area: rfid;
  margin: auto;
}
.icon
{
  padding: 15px;
  width: 0%;
  background: #86aff0 ;
  color: white;
  min-width: 50px;
  margin-top: 10px;
  text-align: center;
}
.input-field
{
  width: 100%;
  margin-top: 10px;
  padding: 10px;
  outline: none;
}
.input-field:focus
{
  border: 2px solid #86aff0 ;
}
.input-field:focus

```

```
{
border: 2px solid #86aff0 ;
}

.message
{
  grid-area: message;
  margin: auto;
}

/* Button for submit */
.btn
{
background-color: #86aff0 ;
color: rgb(0, 0, 0);
margin-top: 20px;
padding: 15px 20px;
border: none;
cursor: pointer;
width: 100%;
opacity: 0.9;
}

.btn:hover
{
color: white;
}

/* Button for submit */

.clock
{
grid-area: clock;
color:#212529;
```

```
font-size: 30px;
font-family: "Lucida Console", Courier, monospace;
```

```
letter-spacing: 5px;
margin:auto;
word-wrap: break-word;
}
```

```
.movingText
{
grid-area:movingText;
align-self: end;
color:#212529;
font-family: "Lucida Console", Courier, monospace;
animation: move 3s;
-webkit-animation: move 3s;
margin-left: 10%;
}
```

```
@keyframes move
{
from
{
margin-left: 100%;
width: 300%;
}
```

```
to
{
margin-left: 10%;
width: 100%;
}
```

```

}

@-webkit-keyframes move
{
from
{
margin-left: 100%;
width: 300%;
}

to
{
margin-left: 10%;
width: 100%;
}
}

```

Javascript files

Index.js (javascript file for index page)

```

$(window).on("load",function()
{
$.loader-wrapper).fadeOut("slow");
});

// ===== MQTT =====
// the ip of the device
var ip = location.host;
// Create a client instance
client = new Paho.MQTT.Client(ip, Number(9001),"clientId");

// set callback handlers

```

```

client.onConnectionLost = onConnectionLost;
client.onMessageArrived = onMessageArrived;

// connect the client
client.connect({onSuccess:onConnect,
               userName:"mqtt",
               password:"zagkoulis",
               mqttVersion: 3

});

// called when the client connects
function onConnect() {
// Once a connection has been made, make a subscription and send a
message.
console.log("onConnect");
client.subscribe("rfidScanner");
}

// called when the client loses its connection
function onConnectionLost(responseObject) {
if (responseObject.errorCode !== 0) {
console.log("onConnectionLost:"+responseObject.errorMessage);
}
}

// called when a message arrives
function onMessageArrived(message) {
console.log("onMessageArrived:"+message.payloadString);
let rf=JSON.parse(message.payloadString);
$("#rfidNumber").text(rf.rfidNumber);
}

```

```

}

// ===== MQTT =====

// ===== ROLOI =====
function showTime()
{
    var date = new Date();
    var h = date.getHours(); // 0 - 23
    var m = date.getMinutes(); // 0 - 59
    var s = date.getSeconds(); // 0 - 59
    var session = "AM";

    if(h == 0)
    {
        h = 12;
    }

    if(h > 12)
    {
        h = h - 12;
        session = "PM";
    }

    h = (h < 10) ? "0" + h : h;
    m = (m < 10) ? "0" + m : m;
    s = (s < 10) ? "0" + s : s;

    var time = h + ":" + m + ":" + s + " " + session;
    document.getElementById("MyClockDisplay").innerText = time;
    document.getElementById("MyClockDisplay").textContent = time;
}

```

```

        setTimeout(showTime, 1000);
    }
    showTime();
    // ===== ROLOI =====

    // This function checks if the user has logged in, if yes, the members
    // are showed up.
    function main()
    {

        let user = getCookie("username");
        if (user != "")
        {
            // console.log("Logged in");
            var div = document.getElementById('welcome');
            div.innerHTML = "Welcome " + user;
            $( "#main" ).load( "main.html #flex-container" );
            membersDetails(); //members inside all members etc
        }
        else
        {
            console.log("Not logged in");
            window.location.href="login.html";
        }
    }

    // this functions returns the saved cookie
    function getCookie(cname)
    {
        let name = cname + "=";
        // console.log("cookie");
    }

```

```

console.log(document.cookie);
let ca = document.cookie.split(';');
for(let i = 0; i < ca.length; i++)
{
    let c = ca[i];
    while (c.charAt(0) == ' ')
    {
        c = c.substring(1);
    }
    if (c.indexOf(name) == 0)
    {
        return c.substring(name.length, c.length);
    }
}
return "";
}

```

// This function show the numbers of all members of the company and the members that are inside the building

```

function membersDetails()
{
    // console.log("membersDetails");
    $.post("./php/allMembers.php",
    function(data)
    { // getting the response and parsing and validating it
        if (data>=0)
        { //on success response
            // console.log("All members"); @debug
            // console.log(data);
            // $('#allmembers').val()="ALL MEMBERS:";

```

```

        $("#allmembersNumber").text(data);
    }
    else
    { // on invalid credentials
        // console.log("Something went wrong!"); @debug
        // console.log(data); @debug
    }
});

$.post("./php/insideMembers.php",
function(data)
{ // getting the response and parsing and validating it
    if (data>=0)
    { //on success response
        // console.log("Inside"); @debug
        // console.log(data); @debug
        $("#insideNumber").text(data);
    }
    else
    { // on invalid credentials
        // console.log("Something went wrong!"); @debug
        // console.log(data); @debug
    }
});
}

// not used function
function home()
{
    window.location.href = "index.html";
}
// this function redirects to login page when you press the logout

```

```

function logout()
{
    window.location.href = "login.html";
}
// this function shows the members list
function members()
{
    $("#main").load("main.html #membersForm");
    membersList();
}
// this function loads the register form
function register()
{
    $("#main").load( "main.html #registerForm");
}

function membersList()
{
    // console.log("membersList");
    var i=0;
    $.getJSON('./php/membersList.php', function(data)
    {
        $.each(data, function()
        {
            var name=data[i][1];
            // console.log(name);
            var surname=data[i][2];
            // console.log(surname);
            var job=data[i][3];
            // console.log(job);
            var email=data[i][4];

```

```

// console.log(email);
var picture=data[i][7];
// console.log(picture);

i++;

var iDiv = document.createElement('div');
iDiv.id = 'list-group-item';
iDiv.className = 'list-group-item';

document.getElementById('membersForm').appendChild(iDiv);

// Now create and append to iDiv
var innerDiv = document.createElement('div');
innerDiv.className = 'flex-member';

// The variable iDiv is still good... Just append to it.
iDiv.appendChild(innerDiv);

var img=new Image();
img.src='uploads/'+picture;
img.id="output";
img.setAttribute("height", "77");
img.setAttribute("width", "120");

innerDiv.appendChild(img);

var divName=document.createElement('div');
divName.textContent=name;
divName.setAttribute("id", "div"+i);
innerDiv.appendChild(divName);

```

```

var divSurname=document.createElement('div');
divSurname.textContent=surname;
innerDiv.appendChild(divSurname);
var divJob=document.createElement('div');
divJob.textContent=job;
innerDiv.appendChild(divJob);

var divinfo=document.createElement('div');
var info = document.createElement('button');
info.innerHTML = 'Info';
info.type="submit";
info.className="info";
info.onclick = function()
infoMember(divName.textContent,divSurname.textContent); };
divinfo.appendChild(info);
var img=new Image();
img.src='./pictures/info.png';
img.className="infoIcon";
img.setAttribute("height", "30");
img.setAttribute("width", "30");
divinfo.appendChild(img);
innerDiv.appendChild(divinfo);

var div=document.createElement('div');
var del = document.createElement('button');
del.innerHTML = 'Remove';
del.type="submit";
del.className="delete";
del.onclick = function()
deleteMember(divName,divSurname); };
div.appendChild(del);
var img=new Image();
img.src='./pictures/delete.jpg';

```

```

        img.className="deleteIcon";
        img.setAttribute("height", "30");
        img.setAttribute("width", "30");
        div.appendChild(img);
        innerDiv.appendChild(div);

    });
});
}

// this function shows the info of a member
function infoMember(divName,divSurname)
{
    // console.log("info");
    // console.log(divName);
    // console.log(divSurname);
    $("#main").load("main.html #reportrange");
    setTimeout(function() { MemberName(divName,divSurname);}, 10)
}

// this function deletes the members
function deleteMember(divName,divSurname)
{
    // console.log("deleteMember pressed");           @debug
    // console.log("name:"+divName.textContent);      @debug
    // console.log("surname"+divSurname.textContent); @debug
    var result = confirm ("Are you sure?");
    if (result)
    {
        $.post("../php/deleteMember.php",
        {
            name: divName.textContent,

```

```

        surname: divSurname.textContent
    },
    function(data)
    { // getting the response and parsing and validating it
        if (data=="ok")
        { //on success response
            // console.log("Delete successful");
            $("#message").text("Delete Successful!");
            members();
            var messageGone=setTimeout(clearMessage, 3000);
        }
        else
        { // on invalid credentials
            // console.log("Something went wrong!");
            // console.log(data);
            $("#message").text(data);
            var messageGone=setTimeout(clearMessage, 3000);
        }
    });
}
else
{
// console.log ("delete cancelled");
}
}

// this function registers a new member
function newMember()
{
// console.log("Register new member...");
var picture=$('#file').val();
picture=picture.replace(/^C:\\fakepath\\\/i, '');

```

```

var name=document.getElementById('name').value;
var surname=document.getElementById('surname').value;
var job=document.getElementById('job').value;
var email=document.getElementById('email').value;
var password=document.getElementById('password').value;
var
confirmpassword=document.getElementById('confirmpassword').value;
var rfidNumber=$('#rfidNumber').text();

    if ($('#name').val() && $('#surname').val() && $('#job').val() &&
$('#email').val() && $('#password').val() &&
$('#confirmpassword').val()&& $('#rfidNumber').text())
    {
        // console.log("Everything is ok");
        $.post("../php/register.php",
        {
            name: name,
            surname: surname,
            job: job,
            email:email,
            password:password,
            rfidNumber:rfidNumber,
            picture:picture
        },
        function(data)
        { // getting the response and parsing and validating it
            if (data=="ok")
            { //on success response
                // console.log("Register successful");
                $("#message").text("Register Successful!");
                var messageGone=setTimeout(clearMessage, 3000);
                setTimeout(uploadFile,1000);
            }
        }
    }

```

```

    }
    else
    { // on invalid credentials
    // alert.innerHTML="invalid username or password";
    // console.log("Something went wrong!");
    // console.log(data);
    $("#message").text(data);
    var messageGone=setTimeout(clearMessage, 3000);
    }
    });
}
else
{
    $("#message").text("Please complete all the required
fields!");
    var messageGone=setTimeout(clearMessage, 3000);
}
}

```

```

// This function uploads the picture
function uploadFile()
{
    var files = document.getElementById("file").files;
    if(files.length > 0 )
    {
        var formData = new FormData();
        formData.append("file", files[0]);
        var xhttp = new XMLHttpRequest();
        // Set POST method and ajax file path
        xhttp.open("POST", "./php/pictures.php", true);
        // call on request changes state
    }
}

```

```

xhttp.onreadystatechange = function()
{
    if (this.readyState == 4 && this.status == 200)
    {
        var response = this.responseText;
        if(response == "Picture uploaded")
        {
            $("#message").text("Picture uploaded");
            var messageGone=setTimeout(clearMessage, 3000);
        }
        else
        {
            $("#message").text(response);
            var messageGone=setTimeout(clearMessage, 3000);
        }
    }
};

// Send request with data
xhttp.send(formData);
}
else
{
    $("#message").text("Please select an image");
    var messageGone=setTimeout(clearMessage, 3000);
}
}

// this function clear the message box
function clearMessage()
{
    $("#message").text("");
}

```

```

}

// this function loads the picture
var loadFile = function(event)
{
    var output = document.getElementById('output');
    output.src = URL.createObjectURL(event.target.files[0]);
    output.onload = function()
    {
        URL.revokeObjectURL(output.src) // free memory
    }
};

// this function show the members inside the building in the popup
window
function openPopup()
{
    var i=0;
    $.getJSON('./php/memberPopup.php', function(data)
    {
        $.each(data, function()
        {
            // console.log(data);
            // console.log(data[i]["name"]);
            // console.log(data[i]["surname"]);
            // console.log(data[i]["time"]);

            var modal = document.getElementById("myModal");
            modal.style.display = "block";
            var inside = document.createElement('span');

```

```

        inside.id = 'insideMember';
        inside.className = 'insideMember';
        inside.textContent=data[i]["name"]+
"+data[i]["surname"] +" "+ data[i]["time"] ;
        inside.style.color = "black";
        document.getElementById('modal-
content').appendChild(inside);
        i++;
    });
});
}

```

// this function deletes the div that came from popup window

function closePopup()

```

{
    var modal = document.getElementById("myModal");
    modal.style.display = "none";
    var                divsToRemove                =
document.getElementsByClassName("insideMember");
    for (var i = divsToRemove.length-1; i >= 0; i--)
    {
        divsToRemove[i].remove();
    }
}

```

// this function show the name and surname in the popup box

function MemberName(divName,divSurname)

```

{

var iDiv = document.createElement('div');
iDiv.id = 'nameSurname';
iDiv.className = 'nameSurname';

```

```

iDiv.textContent="Name:"+divName+" Surname:"+divSurname;
document.getElementById('main').appendChild(iDiv);

}

// this function delete a history log of a member
function deleteHistory(divId)
{
    // console.log("deleteHistory pressed");
    var Id =
divId.textContent.substring(divId.textContent.indexOf(':') + 1,
divId.textContent.length);
    // console.log("id:"+Id);
    var result = confirm ("Are you sure?");
    if (result)
    {
        // console.log("delete in progress");
        $.post("./php/deleteHistory.php",
        {
            id: Id
        },

        function(data)
        { // getting the response and parsing and validating it
            if (data=="ok")
            { //on success response
                // console.log("Delete successful");
                $("#message").text("Delete Successful!");
                var ret =
document.getElementById("nameSurname").textContent;
                ret=ret.replace("Name:", "");
                ret=ret.replace("Surname:", "");
                const myArr = ret.split(" ");

```

```

        infoMember(myArr[0],myArr[1]);
        var messageGone=setTimeout(clearMessage, 3000);
    }
    else
    { // on invalid credentials
        // console.log("Something went wrong!");
        // console.log(data);
        $("#message").text(data);
        var messageGone=setTimeout(clearMessage, 3000);
    }
    });
}
else
{
    // console.log("delete rejected");
}
}

// this function shows the logs of a members for a certain period of
time
function date()
{
    $(function()
    {
        var start = moment().subtract(29, 'days');
        var end = moment();
        function cb(start, end)
        {
            $('#reportrange span').html(start.format('MMMM D, YYYY')
+ ' - ' + end.format('MMMM D, YYYY'));
            // console.log(start);

```

```

        var divsToRemove = document.getElementsByClassName("list-
group-item");
        for (var i = divsToRemove.length-1; i >= 0; i--)
        {
            divsToRemove[i].remove();
        }
        //          var          startDate          =
$('#reportrange').data('daterangepicker').startDate._d;
        // console.log(startDate);
        var          startDate          =
moment($('#reportrange').data('daterangepicker').startDate).toDate()
;

        // console.log(startDate);
        // console.log(startDate.getFullYear());
        // console.log(startDate.getMonth());
        // console.log(startDate.getDate());
        // console.log(startDate.getHours());
        // console.log(startDate.getMinutes());
        // console.log(startDate.getSeconds());
        var          endDate          =
$('#reportrange').data('daterangepicker').endDate._d;
        // console.log(endDate);
        // console.log(endDate.getFullYear());
        // console.log(endDate.getMonth());
        // console.log(endDate.getDate());
        // console.log(endDate.getHours());
        // console.log(endDate.getMinutes());
        // console.log(endDate.getSeconds());

        var          endSQL=endDate.getFullYear()+"-
"+(endDate.getMonth()+1)+"-"+endDate.getDate()          +          "          "          +
endDate.getHours()          +          ":"          +          endDate.getMinutes()          +
":"+endDate.getSeconds();

        var          startSQL=startDate.getFullYear()+"-
"+(startDate.getMonth()+1)+"-"+startDate.getDate()+          "          "          +
startDate.getHours()          +          ":"          +          startDate.getMinutes()          +
":"+startDate.getSeconds();

```

```

// console.log(endSQL);
// console.log(startSQL);
var ret =
document.getElementById("nameSurname").textContent;
ret=ret.replace("Name:", "");
ret=ret.replace("Surname:", "");
const myArr = ret.split(" ");
// console.log(myArr[0]);
var i=0;
$.getJSON('./php/memberHistory.php',
    { name: myArr[0], surname:myArr[1], startDate:
startSQL, endDate:endSQL }, function(data)
    {
$.each(data, function()
    {
        var id = data[i][0];
        var arrival=data[i][2];
        var exodus=data[i][3];
        var comeLate=data[i][4];
        var leftEarly=data[i][5];
        var hoursWorked=data[i][6];
        hoursWorked = hoursWorked.substring(0,
hoursWorked.indexOf('.'));
        var overtime=data[i][7];
        var iDiv = document.createElement('div');
        iDiv.id = 'list-group-item';
        iDiv.className = 'list-group-item';

document.getElementById('main').appendChild(iDiv);
        var innerDiv = document.createElement('div');
        innerDiv.className = 'flex-member';

// The variable iDiv is still good... Just append
to it.

```

```

iDiv.appendChild(innerDiv);

var divId = document.createElement('div');
divId.textContent= "ID:" + id;
divId.setAttribute("id","div"+i);
innerDiv.appendChild(divId);

var divArrival=document.createElement('div');
divArrival.textContent="Arrival: "+arrival;
divArrival.setAttribute("id", "div"+i);
innerDiv.appendChild(divArrival);

var divExodus=document.createElement('div');
divExodus.textContent="Exodus: "+exodus;
divExodus.setAttribute("id", "div"+i);
innerDiv.appendChild(divExodus);

var divcomeLate=document.createElement('div');
divcomeLate.textContent="Arrival      difference:
"+comeLate;

divcomeLate.setAttribute("id", "div"+i);
innerDiv.appendChild(divcomeLate);

var divleftEarly=document.createElement('div');
divleftEarly.textContent="Exodus      difference:
"+leftEarly;

divleftEarly.setAttribute("id", "div"+i);
innerDiv.appendChild(divleftEarly);

var
divHoursWorked=document.createElement('div');

```

```

        divHoursWorked.textContent="Hours          Worked:
"+hoursWorked;

        divHoursWorked.setAttribute("id", "div"+i);
        innerDiv.appendChild(divHoursWorked);

        var divovertime=document.createElement('div');
        divovertime.textContent="Overtime: "+overtime;
        divovertime.setAttribute("id", "div"+i);
        innerDiv.appendChild(divovertime);

        var div=document.createElement('div');
        var del = document.createElement('button');
        del.innerHTML = 'Remove';
        del.type="submit";
        del.className="delete";
        del.onclick = function() { deleteHistory(divId);
};

        div.appendChild(del);
        var img=new Image();
        img.src='./pictures/delete.jpg';
        img.className="deleteIcon";
        img.setAttribute("height", "30");
        img.setAttribute("width", "30");
        div.appendChild(img);
        innerDiv.appendChild(div);

        i++;
    });
});
}

```

```

$('##reportrange').daterangepicker
({
    startDate: start,
    endDate: end,
    ranges:
    {
        'Today': [moment(), moment()],
        'Yesterday': [moment().subtract(1, 'days'),
moment().subtract(1, 'days')],
        'Last 7 Days': [moment().subtract(6, 'days'), moment()],
        'Last 30 Days': [moment().subtract(29, 'days'),
moment()],
        'This Month': [moment().startOf('month'),
moment().endOf('month')],
        'Last Month': [moment().subtract(1,
'month').startOf('month'),
moment().subtract(1,
'month').endOf('month')]
    }
}, cb);
cb(start, end);

});
}

```

Login.js (javascript file for login page)

```
// ===== ROLOI =====  
// the ip of the device  
var ip = location.host;  
  
function showTime()  
{  
    var date = new Date();  
    var h = date.getHours(); // 0 - 23  
    var m = date.getMinutes(); // 0 - 59  
    var s = date.getSeconds(); // 0 - 59  
    var session = "AM";  
  
    if(h == 0)  
    {  
        h = 12;  
    }  
  
    if(h > 12)  
    {  
        h = h - 12;  
        session = "PM";  
    }  
  
    h = (h < 10) ? "0" + h : h;  
    m = (m < 10) ? "0" + m : m;  
    s = (s < 10) ? "0" + s : s;  
  
    var time = h + ":" + m + ":" + s + " " + session;  
    document.getElementById("clockDisplay").innerText = time;  
    document.getElementById("clockDisplay").textContent = time;
```

```

        setTimeout(showTime, 1000);

    }
    showTime();
    // ===== ROLOI =====

function checkLogin()
{
    var username=document.getElementById('username').value;
    var password=document.getElementById('password').value;
    // console.log("username:"+username);
    // console.log("password:"+password);
    // shaking box
    if( !username || !password )
    {
        $("#loginBox").effect( "shake", {times:2}, 200 );
        $("#message").text("Please fill all the required fields!");
        var messageGone=setInterval(clearMessage, 3000);
    }
    else
    {
        $.post("./php/checkLogin.php",
        {
            username: username,
            password: password
        },
        function(data)
        { // getting the response and parsing and validating it
            if (data=="ok")
            { //on success response
                document.cookie= "username" + "=" + username +";";
            }
        }
    }
}

```

```

window.location="http://"+ip+"/site_diplomatikis/index.html";
    // console.log("correct username and password"); debug
purposes
    }
    else
    { // on invalid credentials
        // console.log("wrong username or password"); debug purposes
        $("#loginBox").effect( "shake", {times:2}, 200 );
        $("#message").text("Wrong username or password!");
        var messageGone=setInterval(clearMessage, 3000);
    }
});
}
}

```

```

function clearMessage()
{
    $("#message").text("");
}

```

```

$(document).keypress(function(e)
{
    if(e.which == 13)
    {
        checkLogin();
    }
});

```

```

function startCookie()
{
    document.cookie = "username" + "=" + "" + ";";
}

```

PHP files (files that connect javascript with MYSQL)

Allmembers.php

```
<?php
//This php file shows the number of all members of the company
require('connect.php');
$results = $pdo->query('SELECT * FROM members');

$i=0;
foreach($results as $row)
{
    $i++;
}
echo $i;
?>
```

checkLogin.php

```
<?php
require('connect.php');
// This php file checks if the given username and password are okay

if(isset($_POST['username']) && isset($_POST['password']))
{
    $username = $_POST["username"];
    $password = $_POST["password"];

    $stmt = $pdo->prepare('SELECT * FROM users WHERE username=:username
AND password =:password');

    $criteria=
    [
        'username'=>$username,
        'password'=>$password
```

```

];
$stmt->execute($criteria);
if ($stmt->rowCount() > 0)
{
    echo "ok";
}
else
{
    echo "false";
}
}
?>

```

Connect.php

```

<?php
// This php file connects to the database
$server = "localhost";
$username = "root";
$password = "";
$schema = "rfid_system";

$pdo = new PDO('mysql:dbname=' . $schema . ';host=' . $server,
$username, $password,
[ PDO::ATTR_ERRMODE => PDO::ERRMODE_EXCEPTION]);
$pdo->exec("SET CHARACTER SET 'utf8'");
?>

```

deleteHistory.php

```

<?php
require('connect.php');

// This php file deletes a certain history log of a member

```

```

$id = $_POST["id"];
try
{
    $stmt=$pdo->prepare("DELETE FROM history WHERE id=:id");
    $stmt->bindParam(':id',$id);
    $stmt->execute();
    echo "ok";
}
catch(PDOException $e)
{
    echo "Error: " . $e->getMessage();
}
?>

```

deleteMember.php

```

<?php
require('connect.php');

// This php file deletes a member of the company

$name = $_POST["name"];
$surname = $_POST["surname"];

try
{
    $stmt=$pdo->prepare("DELETE FROM members WHERE name=:name AND
surname=:surname");
    $stmt->bindParam(':name',$name);
    $stmt->bindParam(':surname',$surname);
    $stmt->execute();
    echo "ok";
}
catch(PDOException $e)

```

```
{
    echo "Error: " . $e->getMessage();
}
?>
```

insideMembers.php

```
<?php
require('connect.php');

//This php file shows the number of the members of the company that
are inside the building

$results = $pdo->query('SELECT * FROM attendance');

$i=0;
foreach($results as $row)
{
    $i++;
}

echo $i;
?>
```

memberHistory.php

```
<?php
require('connect.php');

// This php file shows the history of a member for a certain period
of time

$name = $_GET["name"];
$surname= $_GET["surname"];
$startDate=$_GET["startDate"];
$endDate=$_GET["endDate"];
```

```

$to_encode = array();

$stmt = $pdo->prepare('SELECT * FROM members WHERE name=:name AND
surname=:surname' );

$stmt->execute(['name'=>$name, 'surname'=>$surname]);

foreach($stmt as $test)
{
    $results = $pdo->prepare('SELECT * FROM history WHERE arrival >=
:startDate AND exodus <=:endDate AND cardID=:cardID' );
    $results-
>execute(['startDate'=>$startDate, 'endDate'=>$endDate, 'cardID'=>$tes
t[6]]);
    foreach($results as $row)
    {
        $to_encode[]=$row;
    }
}

echo json_encode($to_encode);

?>

```

memberPopup.php

```

<?php
require('connect.php');

// This php file shows the popup message of members that are inside
the building

$results = $pdo->query('SELECT * FROM attendance');

$rows=[];
foreach($results as $row)
{

```

```

$stmt = $pdo->prepare('SELECT * FROM members WHERE
cardID=:cardID');
$criteria=
[
'cardID'=>$row['cardID']
];

$stmt->execute($criteria);
foreach($stmt as $member)
$rows[] = [
    "time" => $row["arrival"],
    "name" => $member["name"],
    "surname" => $member["surname"]
];
}
echo json_encode($rows);
?>

```

membersList.php

```

<?php
require('connect.php');

// This php file shows the list of the members

$results = $pdo->query('SELECT * FROM members');

$to_encode = array();
foreach($results as $row)
{
    $to_encode[] = $row;
}
echo json_encode($to_encode);
?>

```

Pictures.php

```
<?php
```

```
// This php file upload the picture when a new employee is registered
```

```
// Check if image file is a actual image or fake image
```

```
if(isset($_FILES['file']['name']))
```

```
{
```

```
    $target_dir = "../uploads/";
```

```
    $target_file = $target_dir . basename($_FILES["file"]["name"]);
```

```
    $uploadOk = 1;
```

```
    $imageFileType = strtolower(pathinfo($target_file,PATHINFO_EXTENSION));
```

```
    $check = getimagesize($_FILES["file"]["tmp_name"]);
```

```
    if($check !== false)
```

```
    {
```

```
        $uploadOk = 1;
```

```
    }
```

```
    else
```

```
    {
```

```
        echo "File is not an image.";
```

```
        $uploadOk = 0;
```

```
        exit;
```

```
    }
```

```
}
```

```
// Check if file already exists
```

```
if (file_exists($target_file))
```

```
{
```

```
    echo "Sorry, file already exists.";
```

```
    $uploadOk = 0;
```

```
    exit;
```

```

}

// Check file size
if ($_FILES["file"]["size"] > 5000000)
{
    echo "Sorry, your file is too large.";
    $uploadOk = 0;
    exit;
}

// Allow certain file formats
if($imageFileType != "jpg" && $imageFileType != "png" &&
$imageFileType != "jpeg"
&& $imageFileType != "gif" )
{
    echo "Sorry, only JPG, JPEG, PNG & GIF files are allowed.";
    $uploadOk = 0;
    exit;
}

// Check if $uploadOk is set to 0 by an error
if ($uploadOk == 0)
{
    echo "Sorry, your file was not uploaded.";
    exit;
}
// if everything is ok, try to upload file
}
else
{
    if (move_uploaded_file($_FILES["file"]["tmp_name"], $target_file))
    {
        //      echo "The file ". htmlspecialchars( basename(
$_FILES["fileToUpload"]["name"])). " has been uploaded.";

```

```

    echo "Picture uploaded";
    exit;
}
else
{
    echo "Sorry, there was an error uploading your file.";
    $uploadOk=0;
    exit;
}
}

?>

```

Register.php

```

<?php
require('connect.php');

// This php registers the new member of the company

$name = $_POST["name"];
$surname = $_POST["surname"];
$job = $_POST["job"];
$email = $_POST["email"];
$password = $_POST["password"];
$rfidNumber = $_POST["rfidNumber"];
$picture=$_POST["picture"];

$stmt = $pdo->prepare('SELECT * FROM members WHERE name=:name AND
surname =:surname');
$criteria=
[
    'name'=>$name,

```

```

        'surname'=>$surname
    ];
    $stmt->execute($criteria);
    if ($stmt->rowCount() > 0)
    {
        echo "User already exists!";
    }
    else
    {
        $stmt = $pdo->prepare('SELECT * FROM members WHERE cardID=:cardID');
        $criteria=
        [
            'cardID'=>$rfidNumber
        ];
        $stmt->execute($criteria);
        if ($stmt->rowCount() > 0)
        {
            echo "Rfid is already in use!";
        }
        else
        {
            try
            {
                $stmt=$pdo->prepare("INSERT INTO members (name,surname
                ,job ,email ,password ,cardID ,picture) VALUES
                (:name,:surname,:job,:email,:password,:rfidNumber,:picture)");
                $stmt->bindParam(':name',$name);
                $stmt->bindParam(':surname',$surname);
                $stmt->bindParam(':job',$job);
                $stmt->bindParam(':email',$email);
                $stmt->bindParam(':password',$password);
            }
        }
    }
}

```

```

        $stmt->bindParam(':rfidNumber',$rfidNumber);
        $stmt->bindParam(':picture',$picture);
        $stmt->execute();
        echo "ok";
    }
    catch(PDOException $e)
    {
        echo "Error: " . $e->getMessage();
    }
}
}
?>

```

HTML files

Main.html (the html file for main page)

```

<div class="flex-container" id="flex-container">
    <button class="allmembers" onclick="members()" id="allmembers">
        ALL MEMBERS:
        <div class="allmembersNumber" id="allmembersNumber">
            </div>
        </button>
        <button class="inside" id="inside" onclick="openPopup()">INSIDE THE
        BUILDING:
            <div class="insideNumber" id="insideNumber">
                </div>
            </button>

    <div id="myModal" class="modal">
        <div class="modal-content" id="modal-content">
            <span class="close" onclick="closePopup()">&times;</span>

```

```

        </div>
    </div>

</div>

<!-- REGISTER FORM -->
<div id="registerForm">
    <div class="form-group">
        <p><input type="file" accept="image/*" name="image" id="file"
onchange="loadFile(event)" style="display: none;"></p>
        <p><label for="file" style="cursor: pointer;">Upload your
image</label></p>
    </div>
    <div class="form-group">
        <div class="picture">
            
        </div>
    </div>
    <div class="form-group">
        <input type="text" class="form-control" id="name"
placeholder="Enter your name">
    </div>
    <div class="form-group">
        <label for="surname">Surname</label>
        <input type="text" class="form-control" id="surname"
placeholder="Enter your surname">
    </div>

    <div class="form-group">
        <label for="exampleFormControlSelect1">Job</label>
        <select class="form-control" id="job">
            <option>Graphic Designer</option>

```

```

        <option>Electronic engineer</option>
        <option>Artist</option>
        <option>Other</option>
    </select>
</div>

<div class="form-group">
    <label for="exampleInputEmail1">Email address</label>
    <input type="email" class="form-control" id="email" aria-
describedby="emailHelp" placeholder="Enter email">
</div>
<div class="form-group">
    <label for="exampleInputPassword1">Password</label>
    <input type="password" class="form-control" id="password"
placeholder="Password">
</div>

<div class="form-group">
    <label for="exampleInputPassword1">Confirm password</label>
    <input type="password" class="form-control" id="confirmpassword"
placeholder="Password">
</div>

<div class="form-check form-check-inline">
    
    <label class="form-check-label" for="inlineCheckbox1">Scan your
wristband</label>
    <div id="rfidNumber"></div>
</div>

<div class="form-group">
    <button type="submit" id="next" onclick="newMember()"
class="buttonMain">Submit</button>
</div>

```

```

</div>

<!-- REGISTER FORM -->

<!-- MEMBERS FORM -->
<div id="membersForm">
  <ul class="list-group">
  </ul>
</div>
<!-- MEMBERS FORM -->

<!-- MEMBERINFO -->

<div id="reportrange" style="background: #fff; cursor: pointer;
padding: 5px 10px; border: 1px solid #ccc; width: 100%">
  <iframe style="display:none" onload="date()"
src="http://www.w3schools.com"> </iframe>
  <i class="fa fa-calendar"></i>&nbsp;
  <span></span> <i class="fa fa-caret-down"></i>
</div>

<!-- MEMBERINFO -->

```

Index.html (the html file for index page)

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <!-- Required meta tags -->
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1">

    <!-- Bootstrap CSS -->
    <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstra
p.min.css"          rel="stylesheet"          integrity="sha384-
EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTWfSpd3yD65VohhpuuCOmLASjC"
crossorigin="anonymous">
    <link                                rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/4.7.0/css/font-awesome.min.css">
    <link rel="stylesheet" href="./css/index.css">
    <script
src="http://ajax.googleapis.com/ajax/libs/jquery/1.9.1/jquery.min.js
"></script>
    <link                                rel="stylesheet"
href="//code.jquery.com/ui/1.12.1/themes/smoothness/jquery-ui.css">
    <link          rel="shortcut          icon"          type="image/png"
href="./pictures/rfid.png">
    <script          src="https://cdnjs.cloudflare.com/ajax/libs/paho-
mqtt/1.0.1/mqttws31.js" type="text/javascript"></script>
    <script                                type="text/javascript"
src="https://cdn.jsdelivr.net/jquery/latest/jquery.min.js"></script>
    <script                                type="text/javascript"
src="https://cdn.jsdelivr.net/momentjs/latest/moment.min.js"></scrip
t>
    <script                                type="text/javascript"
src="https://cdn.jsdelivr.net/npm/daterangepicker/daterangepicker.mi
n.js"></script>
    <link          rel="stylesheet"          type="text/css"
href="https://cdn.jsdelivr.net/npm/daterangepicker/daterangepicker.c
ss" />
```

```

    <title>Rfid sytem</title>
</head>

<body onload="main()">

    <div class="content">
        <div id="MyClockDisplay" class="clock" onload="showTime()">
        </div>

        <div class="message" id="message">
        </div>

        <!-- sidebar -->
        <div class="sidebar">
            <div class="d-flex align-items-center flex-column"
style="height: 100%;">
                <div class="p-2 ">
                    <div class="d-flex flex-row">
                        
                        <button type="submit" class="button"
onclick="main()">Main</button>
                    </div>
                </div>
            </div>

            <div class="p-2">
                <div class="d-flex flex-row">
                    
                    <button type="submit" class="button"
onclick="members()">Members</button>
                </div>
            </div>
        </div>
    </div>

```

```

        <div class="p-2">
            <div class="d-flex flex-row">
                
                <button type="submit" class="button"
onclick="register()">Register</button>
            </div>
        </div>
        <div class="mt-auto p-2">
            <div class="d-flex flex-row">
                
                <button type="submit" class="button"
onclick="logout()">Logout</button>
            </div>
        </div>
    </div>
</div>
<!-- sidebar -->

<!-- MAIN -->
<div class="main" id="main" >
</div>
<!-- MAIN -->

<!-- WELCOME -->
<div id="welcome" class="welcome">Welcome
</div>
<!-- WELCOME -->

<!-- LOADER -->
<div class="loader-wrapper">
    <span class="loader"><span class="loader-
inner"></span></span>

```

```

        </div>

</div>

</body>

<script src="./javascript/index.js"
type="text/javascript"></script>

</html>

```

Login.html (the html file for login page)

```

<!DOCTYPE html>
<html lang="en">
  <head>
    <!-- Required meta tags -->
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1">

    <!-- Bootstrap CSS -->
    <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstra
p.min.css" rel="stylesheet" integrity="sha384-
EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTWfspd3yD65VohhpuuCOmLASjC"
crossorigin="anonymous">
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/4.7.0/css/font-awesome.min.css">
    <link rel="shortcut icon" type="image/png"
href="./pictures/rfid.png">
    <link rel="stylesheet"
href="//code.jquery.com/ui/1.12.1/themes/smoothness/jquery-ui.css">
    <link rel="stylesheet" href="./css/login.css">

```

```

    <script
src="http://ajax.googleapis.com/ajax/libs/jquery/1.9.1/jquery.min.js
"></script>

    <script src="//code.jquery.com/jquery-1.12.4.js"></script>
    <script src="//code.jquery.com/ui/1.12.1/jquery-
ui.js"></script>
    <title>Rfid sytem</title>
</head>

<body onload="startCookie()">

    <div class="container">

        <div id="clockDisplay" class="clock" onload="showTime()">
        </div>

        <div class="rfid">
            
        </div>

        <div class="loginBox" id="loginBox">

            <div class="d-flex flex-column bd-highlight mb-3">
                <div class="p-2 bd-highlight">
                    <div class="d-flex flex-row">
                        <i class="fa fa-user icon"></i>
                        <input class="input-field" id="username"
type="text"placeholder="Username" name="usrnm">
                    </div>
                </div>
            </div>

            <div class="p-2 bd-highlight">
                <div class="d-flex flex-row">

```

```

        <i class="fa fa-key icon"></i>
        <input      class="input-field"      id="password"
type="password" placeholder="Password" name="psw">
    </div>
</div>
<div class="p-2 bd-highlight">
    <button      onclick="checkLogin()"      type="submit"
class="btn">Register</button>
</div>
</div>
</div>

<h1 class="movingText">RFID Based Access Control</h1>
<p class="message" id="message"></div>

</div>

</body>

<script src="./javascript/login.js" type="text/javascript"></script>

</html>

```



```

/* USER CODE BEGIN Includes */
#include "stdio.h"
#include "string.h"
#include "MFRC522.h"
#include "ST7735.h"
#include "GFX_FUNCTIONS.h"
/* USER CODE END Includes */

/* Private typedef -----
-----*/
/* USER CODE BEGIN PTD */

/* USER CODE END PTD */

/* Private define -----
-----*/
/* USER CODE BEGIN PD */
/* USER CODE END PD */

/* Private macro -----
-----*/
/* USER CODE BEGIN PM */

/* USER CODE END PM */

/* Private variables -----
-----*/
SPI_HandleTypeDef hspi1;
SPI_HandleTypeDef hspi2;

UART_HandleTypeDef huart1;
UART_HandleTypeDef huart2;

```

```

/* USER CODE BEGIN PV */
//uint8_t UART_rxBuffer[1] = {0};

/* USER CODE END PV */

/* Private function prototypes -----
-----*/
void SystemClock_Config(void);
static void MX_GPIO_Init(void);
static void MX_USART2_UART_Init(void);
static void MX_SPI1_Init(void);
static void MX_SPI2_Init(void);
static void MX_USART1_UART_Init(void);
/* USER CODE BEGIN PFP */

/* USER CODE END PFP */

/* Private user code -----
-----*/
/* USER CODE BEGIN 0 */
int _write(int file, char *ptr, int len)
{
    /* Implement your write code here, this is used by puts and printf
    for example */
    int i=0;
    for(i=0 ; i<len ; i++)
        ITM_SendChar((*ptr++));
    return len;
}
/* USER CODE END 0 */

/**
 * @brief The application entry point.

```

```

    * @retval int
    */
int main(void)
{
    /* USER CODE BEGIN 1 */

    /* USER CODE END 1 */

    /* MCU Configuration-----
-----*/

    /* Reset of all peripherals, Initializes the Flash interface and the
Systick. */
    HAL_Init();

    /* USER CODE BEGIN Init */
    void writeTFT()
    {
        fillScreen(WHITE);
        ST7735_SetRotation(3);
        ST7735_WriteString(30, 10, "DIPLOMA", Font_11x18,
BLUE,WHITE);
        ST7735_WriteString(30, 50, "RFID", Font_11x18,
BLUE,WHITE);
    }
    /* USER CODE END Init */

    /* Configure the system clock */
    SystemClock_Config();

    /* USER CODE BEGIN SysInit */

    /* USER CODE END SysInit */

```

```

/* Initialize all configured peripherals */
MX_GPIO_Init();
MX_USART2_UART_Init();
MX_SPI1_Init();
MX_SPI2_Init();
MX_USART1_UART_Init();
/* USER CODE BEGIN 2 */
uint8_t rxBuffer[3] = {0,0};
HAL_UART_Receive_IT (&huart2, rxBuffer, 2);

MFRC522_Init();
u_char status, cardstr[MAX_LEN+1];
status = Read_MFRC522(VersionReg);

ST7735_Init(0);
writeTFT();
printf("Start\n");
HAL_Delay(1000);

// LED ON
HAL_GPIO_WritePin(GPIOC, GPIO_PIN_13, GPIO_PIN_SET);
HAL_GPIO_WritePin(GPIOA, GPIO_PIN_8, GPIO_PIN_SET);

HAL_Delay(2000);
// LED OFF
HAL_GPIO_WritePin(GPIOC, GPIO_PIN_13, GPIO_PIN_RESET);
HAL_GPIO_WritePin(GPIOA, GPIO_PIN_8, GPIO_PIN_RESET);
HAL_Delay(2000);
HAL_GPIO_WritePin(GPIOA, GPIO_PIN_8, GPIO_PIN_SET);

/* USER CODE END 2 */

```

```

/* Infinite loop */
/* USER CODE BEGIN WHILE */
while (1)
{

    for (int i = 0; i < 16; i++){
        cardstr[i] = 0;
    }

    status = 0;
    status = MFRC522_Request(PICC_REQIDL, cardstr);
    if(status == MI_OK) {
        status = MFRC522_Anticoll(cardstr);
        if(status == MI_OK) {
            char line[16];
            int size_len = snprintf(line, sizeof(line),
"%X%X%X%X", cardstr[0], cardstr[1], cardstr[2], cardstr[3]);
            HAL_UART_Transmit (&huart2,(uint8_t*) line,
size_len, 10);
            HAL_UART_Receive (&huart2, rxBuffer, 2, 2000);

            printf("UID:%x %x %x %x \n", cardstr[0],
cardstr[1], cardstr[2], cardstr[3]);

            char buffer[20]={0};
            int size_buffer
=sprintf(buffer,sizeof(buffer),"%d kai %d kai
%d",rxBuffer[0],rxBuffer[1],rxBuffer[3]);

            if(rxBuffer[0]==48)
            {

```

```

        ST7735_WriteString(10, 110,"Scan again",
Font_7x10, RED,WHITE);
        HAL_Delay(1000);
    }

    if(rxBuffer[0]==49)
    {
        ST7735_WriteString(10, 110,"Welcome",
Font_7x10, RED,WHITE);
        HAL_GPIO_WritePin(GPIOA, GPIO_PIN_8,
GPIO_PIN_RESET);
        HAL_GPIO_WritePin(GPIOC, GPIO_PIN_13,
GPIO_PIN_SET);
        HAL_Delay(2000);
        HAL_GPIO_WritePin(GPIOC, GPIO_PIN_13,
GPIO_PIN_RESET);
        HAL_GPIO_WritePin(GPIOA, GPIO_PIN_8,
GPIO_PIN_SET);
        HAL_Delay(1000);
    }
    if(rxBuffer[0]==50)
    {
        ST7735_WriteString(10, 110,"Not
Authorized", Font_7x10, RED,WHITE);
        for (int i =0;i<10;i++)
        {
            HAL_GPIO_WritePin(GPIOC, GPIO_PIN_13,
GPIO_PIN_SET);
            HAL_Delay(100);
            HAL_GPIO_WritePin(GPIOC, GPIO_PIN_13,
GPIO_PIN_RESET);
            HAL_Delay(100);
        }
    }

```

```

        }

        if(rxBuffer[0]==51)
        {
            ST7735_WriteString(10,          110,"Leaving",
Font_7x10, RED,WHITE);
            HAL_GPIO_WritePin(GPIOC,      GPIO_PIN_13,
GPIO_PIN_SET);
            HAL_Delay(2000);
            HAL_GPIO_WritePin(GPIOC,      GPIO_PIN_13,
GPIO_PIN_RESET);
            HAL_Delay(1000);
        }
        rxBuffer[0] = 0;
        rxBuffer[1] = 0;
        rxBuffer[2] = 0;
        HAL_Delay(1000);
        writeTFT();
    }

}

/* USER CODE END WHILE */

/* USER CODE BEGIN 3 */
}
void HAL_UART_RxCpltCallback(UART_HandleTypeDef *huart)
{
    HAL_GPIO_WritePin(GPIOC, GPIO_PIN_13, GPIO_PIN_SET);
    HAL_Delay(2000);
    HAL_GPIO_WritePin(GPIOC, GPIO_PIN_13, GPIO_PIN_RESET);
    HAL_UART_Receive_IT(&huart2, rxBuffer, 1);
}

```

```

    /* USER CODE END 3 */
}

/**
 * @brief System Clock Configuration
 * @retval None
 */
void SystemClock_Config(void)
{
    RCC_OscInitTypeDef RCC_OscInitStruct = {0};
    RCC_ClkInitTypeDef RCC_ClkInitStruct = {0};

    /** Initializes the RCC Oscillators according to the specified
    parameters
    * in the RCC_OscInitTypeDef structure.
    */
    RCC_OscInitStruct.OscillatorType = RCC_OSCILLATORTYPE_HSE;
    RCC_OscInitStruct.HSEState = RCC_HSE_ON;
    RCC_OscInitStruct.HSEPredivValue = RCC_HSE_PREDIV_DIV1;
    RCC_OscInitStruct.HSIState = RCC_HSI_ON;
    RCC_OscInitStruct.PLL.PLLState = RCC_PLL_ON;
    RCC_OscInitStruct.PLL.PLLSource = RCC_PLLSOURCE_HSE;
    RCC_OscInitStruct.PLL.PLLMUL = RCC_PLL_MUL9;
    if (HAL_RCC_OscConfig(&RCC_OscInitStruct) != HAL_OK)
    {
        Error_Handler();
    }

    /** Initializes the CPU, AHB and APB buses clocks
    */
    RCC_ClkInitStruct.ClockType = RCC_CLOCKTYPE_HCLK|RCC_CLOCKTYPE_SYSCLK

```

```

|RCC_CLOCKTYPE_PCLK1|RCC_CLOCKTYPE_PCLK2;
RCC_ClkInitStruct.SYSCLKSource = RCC_SYSCLKSOURCE_PLLCLK;
RCC_ClkInitStruct.AHBCLKDivider = RCC_SYSCLK_DIV1;
RCC_ClkInitStruct.APB1CLKDivider = RCC_HCLK_DIV2;
RCC_ClkInitStruct.APB2CLKDivider = RCC_HCLK_DIV1;

if (HAL_RCC_ClockConfig(&RCC_ClkInitStruct, FLASH_LATENCY_2) !=
HAL_OK)
{
Error_Handler();
}
}

/**
 * @brief SPI1 Initialization Function
 * @param None
 * @retval None
 */
static void MX_SPI1_Init(void)
{

/* USER CODE BEGIN SPI1_Init 0 */

/* USER CODE END SPI1_Init 0 */

/* USER CODE BEGIN SPI1_Init 1 */

/* USER CODE END SPI1_Init 1 */
/* SPI1 parameter configuration*/
hspi1.Instance = SPI1;
hspi1.Init.Mode = SPI_MODE_MASTER;
hspi1.Init.Direction = SPI_DIRECTION_1LINE;

```

```

hspi1.Init.DataSize = SPI_DATASIZE_8BIT;
hspi1.Init.CLKPolarity = SPI_POLARITY_LOW;
hspi1.Init.CLKPhase = SPI_PHASE_1EDGE;
hspi1.Init.NSS = SPI_NSS_SOFT;
hspi1.Init.BaudRatePrescaler = SPI_BAUDRATEPRESCALER_64;
hspi1.Init.FirstBit = SPI_FIRSTBIT_MSB;
hspi1.Init.TIMode = SPI_TIMODE_DISABLE;
hspi1.Init.CRCCalculation = SPI_CRCCALCULATION_DISABLE;
hspi1.Init.CRCPolynomial = 10;
if (HAL_SPI_Init(&hspi1) != HAL_OK)
{
    Error_Handler();
}
/* USER CODE BEGIN SPI1_Init 2 */

/* USER CODE END SPI1_Init 2 */

}

/**
 * @brief SPI2 Initialization Function
 * @param None
 * @retval None
 */
static void MX_SPI2_Init(void)
{
    /* USER CODE BEGIN SPI2_Init 0 */

    /* USER CODE END SPI2_Init 0 */

    /* USER CODE BEGIN SPI2_Init 1 */

```

```

/* USER CODE END SPI2_Init 1 */
/* SPI2 parameter configuration*/
hspi2.Instance = SPI2;
hspi2.Init.Mode = SPI_MODE_MASTER;
hspi2.Init.Direction = SPI_DIRECTION_2LINES;
hspi2.Init.DataSize = SPI_DATASIZE_8BIT;
hspi2.Init.CLKPolarity = SPI_POLARITY_LOW;
hspi2.Init.CLKPhase = SPI_PHASE_1EDGE;
hspi2.Init.NSS = SPI_NSS_SOFT;
hspi2.Init.BaudRatePrescaler = SPI_BAUDRATEPRESCALER_8;
hspi2.Init.FirstBit = SPI_FIRSTBIT_MSB;
hspi2.Init.TIMode = SPI_TIMODE_DISABLE;
hspi2.Init.CRCCalculation = SPI_CRCCALCULATION_DISABLE;
hspi2.Init.CRCPolynomial = 10;
if (HAL_SPI_Init(&hspi2) != HAL_OK)
{
    Error_Handler();
}
/* USER CODE BEGIN SPI2_Init 2 */

/* USER CODE END SPI2_Init 2 */

}

/**
 * @brief USART1 Initialization Function
 * @param None
 * @retval None
 */
static void MX_USART1_UART_Init(void)
{

```

```

/* USER CODE BEGIN USART1_Init 0 */

/* USER CODE END USART1_Init 0 */

/* USER CODE BEGIN USART1_Init 1 */

/* USER CODE END USART1_Init 1 */
huart1.Instance = USART1;
huart1.Init.BaudRate = 115200;
huart1.Init.WordLength = UART_WORDLENGTH_8B;
huart1.Init.StopBits = UART_STOPBITS_1;
huart1.Init.Parity = UART_PARITY_NONE;
huart1.Init.Mode = UART_MODE_TX_RX;
huart1.Init.HwFlowCtl = UART_HWCONTROL_NONE;
huart1.Init.OverSampling = UART_OVERSAMPLING_16;
if (HAL_UART_Init(&huart1) != HAL_OK)
{
    Error_Handler();
}
/* USER CODE BEGIN USART1_Init 2 */

/* USER CODE END USART1_Init 2 */

}

/**
 * @brief USART2 Initialization Function
 * @param None
 * @retval None
 */
static void MX_USART2_UART_Init(void)

```

```

{

/* USER CODE BEGIN USART2_Init 0 */

/* USER CODE END USART2_Init 0 */

/* USER CODE BEGIN USART2_Init 1 */

/* USER CODE END USART2_Init 1 */
huart2.Instance = USART2;
huart2.Init.BaudRate = 9600;
huart2.Init.WordLength = UART_WORDLENGTH_8B;
huart2.Init.StopBits = UART_STOPBITS_1;
huart2.Init.Parity = UART_PARITY_NONE;
huart2.Init.Mode = UART_MODE_TX_RX;
huart2.Init.HwFlowCtl = UART_HWCONTROL_NONE;
huart2.Init.OverSampling = UART_OVERSAMPLING_16;
if (HAL_UART_Init(&huart2) != HAL_OK)
{
    Error_Handler();
}
/* USER CODE BEGIN USART2_Init 2 */

/* USER CODE END USART2_Init 2 */

}

/**
 * @brief GPIO Initialization Function
 * @param None
 * @retval None
 */

```

```

static void MX_GPIO_Init(void)
{
    GPIO_InitTypeDef GPIO_InitStructure = {0};

    /* GPIO Ports Clock Enable */
    __HAL_RCC_GPIOC_CLK_ENABLE();
    __HAL_RCC_GPIOD_CLK_ENABLE();
    __HAL_RCC_GPIOA_CLK_ENABLE();
    __HAL_RCC_GPIOB_CLK_ENABLE();

    /*Configure GPIO pin Output Level */
    HAL_GPIO_WritePin(GPIOC, GPIO_PIN_13, GPIO_PIN_RESET);

    /*Configure GPIO pin Output Level */
    HAL_GPIO_WritePin(GPIOA,
        GPIO_PIN_1|GPIO_PIN_4|GPIO_PIN_6|GPIO_PIN_8
        |RC522_CS_Pin|RC522_Rst_Pin,
        GPIO_PIN_RESET);

    /*Configure GPIO pin : PC13 */
    GPIO_InitStructure.Pin = GPIO_PIN_13;
    GPIO_InitStructure.Mode = GPIO_MODE_OUTPUT_PP;
    GPIO_InitStructure.Pull = GPIO_NOPULL;
    GPIO_InitStructure.Speed = GPIO_SPEED_FREQ_LOW;
    HAL_GPIO_Init(GPIOC, &GPIO_InitStructure);

    /*Configure GPIO pins : PA1 PA4 PA6 PA8
        RC522_CS_Pin RC522_Rst_Pin */
    GPIO_InitStructure.Pin = GPIO_PIN_1|GPIO_PIN_4|GPIO_PIN_6|GPIO_PIN_8
        |RC522_CS_Pin|RC522_Rst_Pin;
    GPIO_InitStructure.Mode = GPIO_MODE_OUTPUT_PP;
    GPIO_InitStructure.Pull = GPIO_NOPULL;
    GPIO_InitStructure.Speed = GPIO_SPEED_FREQ_LOW;

```

```

    HAL_GPIO_Init(GPIOA, &GPIO_InitStruct);

}

/* USER CODE BEGIN 4 */

/* USER CODE END 4 */

/**
 * @brief This function is executed in case of error occurrence.
 * @retval None
 */
void Error_Handler(void)
{
    /* USER CODE BEGIN Error_Handler_Debug */
    /* User can add his own implementation to report the HAL error return
    state */
    __disable_irq();
    while (1)
    {
    }
    /* USER CODE END Error_Handler_Debug */
}

#ifdef USE_FULL_ASSERT
/**
 * @brief Reports the name of the source file and the source line
    number
 *
 * @param file: where the assert_param error has occurred.
 * @param line: pointer to the source file name
 * @param line: assert_param error line source number
 * @retval None
 */

```

```
void assert_failed(uint8_t *file, uint32_t line)
{
    /* USER CODE BEGIN 6 */
    /* User can add his own implementation to report the file name and
line number,
ex: printf("Wrong parameters value: file %s on line %d\r\n",
file, line) */
    /* USER CODE END 6 */
}
#endif /* USE_FULL_ASSERT */

/***** (C) COPYRIGHT STMicroelectronics *****/
OF FILE*****/
```