

Εφαρμογή Arduino για Απομακρυσμένο έλεγχο θύρας μέσω Ασύρματου δικτύου Wi-Fi.



**ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ**  
**ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ**  
**ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ**  
**ΥΠΟΛΟΓΙΣΤΩΝ**

**ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ**

Εφαρμογή Arduino  
για  
Απομακρυσμένο έλεγχο θύρας  
μέσω  
Ασύρματου δικτύου Wi-Fi

Κόντη Έλντι

Εισηγήτρια: Αναστασία Ν. Βελώνη, Λέκτορας Εφαρμογών

Εφαρμογή Arduino για Απομακρυσμένο έλεγχο θύρας μέσω Ασύρματου δικτύου Wi-Fi.

## **ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ**

Εφαρμογή Arduino για Απομακρυσμένο έλεγχο θύρας μέσω Ασύρματου δικτύου Wi-Fi.

Κόντη Έλντι

A.M. 39347

**Εισηγήτρια:**

Δρ. Βελώνη Αναστασία

**Εξεταστική Επιτροπή:**

1. ....**A. Βελώνη**.....

2. ....**Σ. Φατούρος**.....

3. ....**I. Έλληνας**.....

**Ημερομηνία εξέτασης .....**

Κόντη Έλντι

Εφαρμογή Arduino για Απομακρυσμένο έλεγχο θύρας μέσω Ασύρματου δικτύου Wi-Fi.

Εφαρμογή Arduino για Απομακρυσμένο έλεγχο θύρας μέσω Ασύρματου δικτύου Wi-Fi.

## ΕΥΧΑΡΙΣΤΙΕΣ

Στο σημείο αυτό θα ήθελα να ευχαριστήσω την οικογένεια μου που στάθηκε στο πλευρό μου όλον αυτόν τον καιρό ηθικά καθώς και οικονομικά για την ολοκλήρωση αυτής της πτυχιακής εργασίας. Επίσης θα ήθελα να ευχαριστήσω και την εισηγήτρια μου, Δρ. Βελώνη Αναστασία για την υποστήριξη και καθοδήγησή της στην παρούσα πτυχιακή εργασία.

Εφαρμογή Arduino για Απομακρυσμένο έλεγχο θύρας μέσω Ασύρματου δικτύου Wi-Fi.

## ΠΕΡΙΛΗΨΗ

Στην παρούσα διπλωματική εργασία σχεδιάσθηκε κάτοψη οικίας με γκαραζόπορτα και δημιουργήθηκε σε μακέτα μαζί με τον μηχανισμό της ηλεκτρικής του εγκατάστασης. Αναπτύχθηκε σύστημα ελέγχου, χρησιμοποιώντας την πλατφόρμα Arduino, των ηλεκτρικών στοιχείων του στα πρότυπα του «έξυπνου σπιτιού» καθώς και εφαρμογή για λειτουργικό σύστημα Android για τον απομακρυσμένο έλεγχο της εγκατάστασης.

## ABSTRACT

At this diploma thesis an electric installation was developed and applied in a model garage door. Using Arduino the electrical functions were automated according to “smart home” standards. Moreover an Android application was developed for controlling the installation remotely.

### **Επιστημονική Περιοχή:**

Μικροελεγκτές και προγραμματισμός ηλεκτρονικών υπολογιστών

### **Λέξεις Κλειδιά:**

μικροελεγκτής, Arduino, διαδικτυακός έλεγχος, έξυπνη οικία, Android

Εφαρμογή Arduino για Απομακρυσμένο έλεγχο θύρας μέσω Ασύρματου δικτύου Wi-Fi.



## ΠΕΡΙΕΧΟΜΕΝΑ

<a href="#">ΚΑΤΑΛΟΓΟΣ ΕΙΚΟΝΩΝ</a>	7
<a href="#">ΚΑΤΑΛΟΓΟΣ ΠΙΝΑΚΩΝ</a>	9
<a href="#">ΣΥΝΤΟΜΟΓΡΑΦΙΕΣ</a>	11
<a href="#">1. ΕΙΣΑΓΩΓΗ</a>	13
<a href="#">1.1 Σκοπός της Πτυχιακής</a>	13
<a href="#">1.2 Internet of Things</a>	13
<a href="#">1.3 Πλάνο Λειτουργίας Εφαρμογής</a>	14
<a href="#">2. ΣΧΕΔΙΑΣΗ ΜΑΚΕΤΑΣ</a>	15
<a href="#">2.1 Αρχιτεκτονικό Σχέδιο</a>	15
<a href="#">2.2 Υλοποίηση Μακέτας</a>	15
<a href="#">2.3 Ηλεκτρολογική Εγκατάσταση</a>	16
<a href="#">3. ΠΛΗΡΟΦΟΡΙΕΣ ARDUINO</a>	19
<a href="#">3.1 Τι είναι το Arduino</a>	19
<a href="#">3.2 Πλακέτα</a>	19
<a href="#">3.3 Λογισμικό Ανάπτυξης</a>	20
<a href="#">3.4 Arduino UNO</a>	21
<a href="#">3.5 ESP8266 Wi-Fi Module</a>	22
<a href="#">4. ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ARDUINO</a>	25
<a href="#">4.1 Αρχές Προγραμματισμού</a>	25
<a href="#">4.2 Αναβάθμιση Firmware ESP8266</a>	25
<a href="#">4.3 Εντολές Επικοινωνίας Wi-Fi</a>	26
<a href="#">4.4 Εντολές Λειτουργίας Γκαραζόπορτας</a>	27
<a href="#">4.5 Ιστοσελίδα Εφαρμογής</a>	28
<a href="#">5. ΠΛΗΡΟΦΟΡΙΕΣ ANDROID</a>	31
<a href="#">5.1 Τι είναι το Android</a>	31
<a href="#">5.2 Περιβάλλον Android</a>	31
<a href="#">5.3 Λογισμικό Android</a>	32
<a href="#">6. ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΗΣ ANDROID</a>	35
<a href="#">6.1 Αρχές Προγραμματισμού</a>	35
<a href="#">6.2 Η εφαρμογή</a>	37
<a href="#">6.3 Κύρια Δραστηριότητα</a>	37
<a href="#">6.4 Έλεγχος Λειτουργίας Γκαραζόπορτας</a>	39
<a href="#">7. ΣΥΜΠΕΡΑΣΜΑΤΑ</a>	41

<a href="#">7.1 Διαπιστώσεις</a>	41
<a href="#">7.2 Εξέλιξη</a>	41
<a href="#">8. ΒΙΒΛΙΟΓΡΑΦΙΑ</a>	43
<a href="#">9. ΠΑΡΑΡΤΗΜΑ</a>	45
<a href="#">1. Μακέτα</a>	45
<a href="#">2. Κώδικας Arduino</a>	46
<a href="#">3. Κώδικας Android</a>	51

## ΚΑΤΑΛΟΓΟΣ ΕΙΚΟΝΩΝ

<a href="#"><u>Figure 1 Μακέτα εγκατάστασης</u></a>	15
<a href="#"><u>Figure 2 Metal Gearmotor 25mm - 100rpm</u></a>	16
<a href="#"><u>Figure 3 L293D Motor Driver</u></a>	16
<a href="#"><u>Figure 4 ESP8266 Wi-Fi Module</u></a>	17
<a href="#"><u>Figure 5 Voltage regulator LD1117 - 3.3V</u></a>	17
<a href="#"><u>Figure 6 Στιγμιότυπο λογισμικού Arduino</u></a>	20
<a href="#"><u>Figure 7 Arduino Uno</u></a>	21
<a href="#"><u>Figure 8 ESP8266 Wi-Fi Module</u></a>	22
<a href="#"><u>Figure 9 ESP8266 Flash Software</u></a>	25
<a href="#"><u>Figure 10 Ακίδες του ESP8266 Wi-Fi module</u></a>	26
<a href="#"><u>Figure 11 Στιγμιότυπο οθόνης Android Studio</u></a>	33
<a href="#"><u>Figure 12 Εφαρμογή Android (Αρχική – Ανοιχτή)</u></a>	39
<a href="#"><u>Figure 13 Εφαρμογή Android (Κλειστή - Σταματημένη)</u></a>	40
<a href="#"><u>Figure 14 Κάτοψη εγκατάστασης</u></a>	45

Εφαρμογή Arduino για Απομακρυσμένο έλεγχο θύρας μέσω Ασύρματου δικτύου Wi-Fi.

## ΚΑΤΑΛΟΓΟΣ ΠΙΝΑΚΩΝ

Table 1 Χαρακτηριστικά επεξεργαστή Arduino Uno

Table 2 Χαρακτηριστικά του ESP8266 Wi-Fi module

Εφαρμογή Arduino για Απομακρυσμένο έλεγχο θύρας μέσω Ασύρματου δικτύου Wi-Fi.

## ΣΥΝΤΟΜΟΓΡΑΦΙΕΣ

IoT - Internet of Things

RS - Recommended Standard

TTL - Transistor Transistor Logic

USB - Universal Serial Bus

I/O - Input Output

IDE - Integrated Development Environment

CPU - Central Processing Unit

ICSP - In-Circuit Serial Programming

EEPROM - Electrically Erasable Programmable Read-Only Memory

EPROM - Erasable Programmable Read-Only Memory

SRAM - Static Random Access Memory

SOK - System On a Chip

TCP - Transmission Control Protocol

IP - Internet Protocol

GPIO - General Purpose Input/Output

PCB - Printed Circuit Board

P2P - Peer to Peer

Balun - Balanced Unbalanced (signals)

LNA - Low Noise Amplifier

PLL - Phase Locked Loop

DCXO - Digital Controlled Crystal Oscillators

SDIO - Secure Digital Input Output

SPI - Serial Peripheral Interface

STBC - Space-Time Block Code

MIMO - Multiple-Input and Multiple-Output

A-MPDU - Aggregate MAC Protocol Data Unit

A-MSDU - Aggregate MAC Service Data Unit

DTIM - Delivery Traffic Indication Map

Εφαρμογή Arduino για Απομακρυσμένο έλεγχο θύρας μέσω Ασύρματου δικτύου Wi-Fi.

SDK - Software Development Kit

XML - eXtensible Markup Language

API - Application Programming Interface

UI - User Interface

USART - Universal Synchronous/Asynchronous Receiver/Transmitter

UART - Universal Asynchronous Receiver/Transmitter

GUI - Graphical User Interface

HTML - HyperText Markup Language

CSS - Cascading Style Sheets



## 1. ΕΙΣΑΓΩΓΗ

### 1.1 Σκοπός της Πτυχιακής

Σκοπός της πτυχιακής μου εργασίας είναι η δημιουργία μιας εύχρηστης εφαρμογής η οποία θα βελτιώσει την καθημερινότητα του ανθρώπου μετατρέποντας το σπίτι του σε ένα πλήρως διαδικτυακά ελεγχόμενο σπίτι. Ο χρήστης θα μπορεί οποιαδήποτε στιγμή, από οποιοδήποτε μέρος, μέσω οποιασδήποτε συσκευής με πρόσβαση στο διαδίκτυο να αποκτήσει πρόσβαση στον έλεγχο των συσκευών του σπιτιού του.

Για να παρουσιαστεί κατά πόσο είναι δυνατή μία τέτοια σύνδεση και η λειτουργικότητά της, επιλέχθηκε να δημιουργηθεί μακέτα μίας γκαραζόπορτας μαζί με την ηλεκτρολογική της εγκατάσταση της οποίας ο έλεγχος θα επιτευχθεί με την χρήση της πλακέτας ανοικτού κώδικα Arduino μέσω εφαρμογής Android.

### 1.2 Internet of Things

Η πτυχιακή εργασία εκπονήθηκε ως μέρος της γενικότερης φιλοσοφίας του "Internet of Things(IoT).

Το IoT είναι το δίκτυο των φυσικών αντικειμένων ή συσκευών με ενσωματωμένα ηλεκτρονικά, λογισμικό, αισθητήρες, και συνδεσιμότητα έτσι ώστε τα αντικείμενα να ανταλλάσσουν δεδομένα με τον κατασκευαστή ή και άλλες συνδεδεμένες συσκευές. Το Internet of Things επιτρέπει σε αντικείμενα να ανιχνεύονται και να ελέγχονται εξ αποστάσεως χρησιμοποιώντας υπάρχουσες υποδομές του δικτύου.

Επίσης δημιουργείται η δυνατότητα για πιο άμεση επικοινωνία μεταξύ του φυσικού κόσμου και κάποιων computer-based συστημάτων παγκοσμίως και οδηγεί σε βελτίωση της αποτελεσματικότητας, της ακρίβειας και του οικονομικού οφέλους.

### 1.3 Πλάνο Λειτουργίας Εφαρμογής

Η εφαρμογή αποτελείται από 3 βασικά σημεία λειτουργίας:

- Την πλακέτα εφαρμογών Arduino.
- Μια διαδικτυακή ιστοσελίδα.
- Μια εφαρμογή Android που διαχειρίζεται τις λειτουργίες της διαδικτυακής ιστοσελίδας.

Το πλάνο λειτουργίας της εφαρμογής είναι το ακόλουθο:

- Η οικιακή συσκευή που θέλουμε να ελέγξουμε βρίσκεται συνδεδεμένη με την πλακέτα Arduino.
- Η συσκευή ελέγχεται από την πλακέτα ESP8266 Wi-Fi module που βρίσκεται συνδεδεμένη στην πλακέτα Arduino.
- Στην παραπάνω πλακέτα που λειτουργεί ως διακομιστής βρίσκεται εγκατεστημένη μια διαδικτυακή ιστοσελίδα.
- Το πρόγραμμά μας θα στέλνει στην πλακέτα κάποιες τιμές και θα ενεργοποιεί τον κινητήρα της γκαραζόπορτας αναλόγως την περίπτωση.
- Το πρόγραμμά μας θα δέχεται από την πλακέτα ESP8266 πληροφορίες σε σχέση με την κατάσταση της γκαραζόπορτας η οποία θα εμφανίζεται στην ιστοσελίδα.
- Η διαδικτυακή μας ιστοσελίδα θα λειτουργήσει ως χειριστήριο επιτρέποντας στον χρήστη τον έλεγχο της συσκευής του καθώς και την ενημέρωσή του για την κατάσταση που βρίσκεται.

## 2. ΣΧΕΔΙΑΣΗ ΜΑΚΕΤΑΣ

### 2.1 Αρχιτεκτονικό Σχέδιο

Σχεδιάσθηκε ισόγειο κτίριο επιφάνειας 30,24 τ.μ. με εξωτερικό χώρο γκαράζ και πλησίον η είσοδος της κατοικίας. Το σχέδιο αναπτύχθηκε στο πρόγραμμα Autocad και η ολοκληρωμένη κάτοψη υπάρχει στο παράρτημα.

### 2.2 Υλοποίηση Μακέτας

Για τη μακέτα χρησιμοποιήθηκε λευκή γυαλιστερή ινοσανίδα μεσαίας πυκνότητας πάχους 15 mm. Η κλίμακα που επιλέχθηκε είναι 1:20 και η μακέτα έχει διαστάσεις 27 x 28 x 16 cm. Για τη διευκόλυνση της τοποθέτησης του κινητήρα, δόθηκε ανύψωση της μακέτας με στηρίγματα 7 cm. Οι τοίχοι έγιναν με διογκωμένη πολυστερίνη (φελιζόλ) πάχους 1 cm για καλύτερη αισθητική.

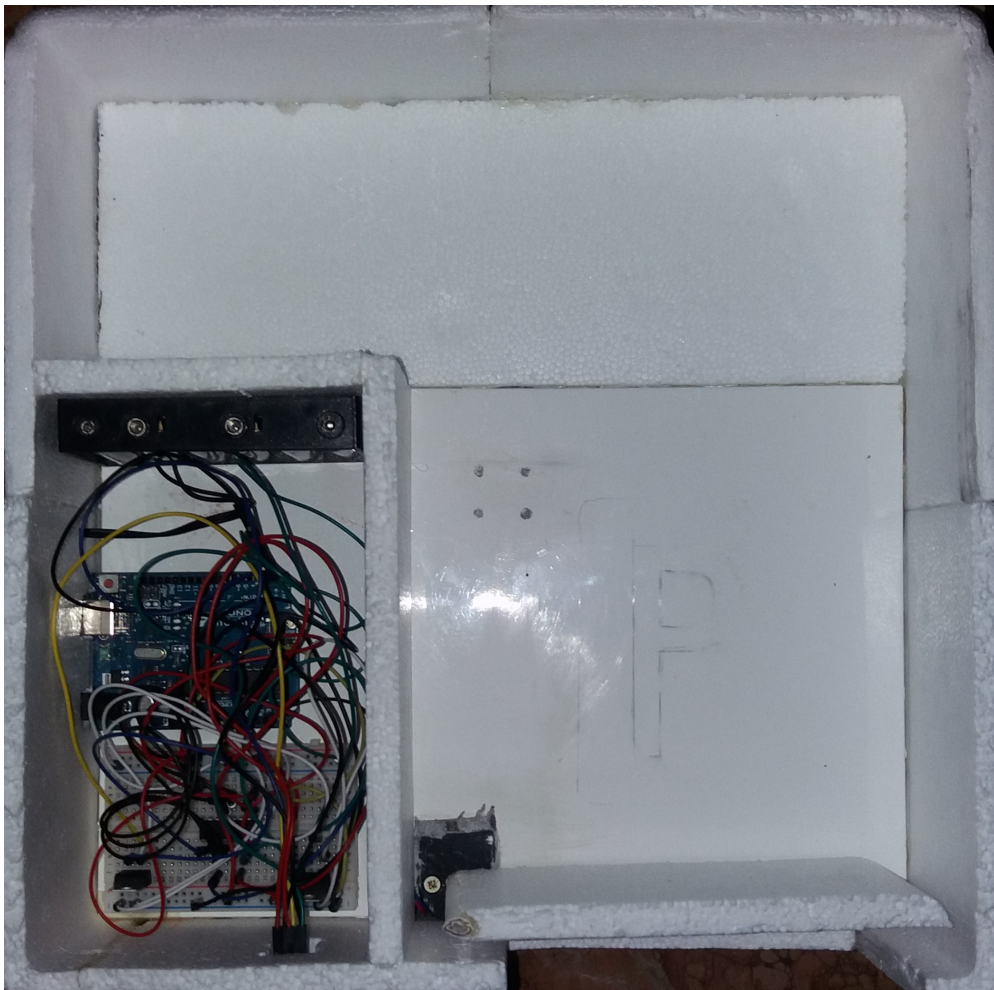


Figure 1 Μακέτα εγκατάστασης

### 2.3 Ηλεκτρολογική Εγκατάσταση

Για την ηλεκτρολογική εγκατάσταση επιλέχθηκε τάση λειτουργίας στα 5 V που παρέχεται από τον ελεγκτή και εναλλακτικά για λόγους φορητότητας χρησιμοποιήθηκε τροφοδοσία μέσω μπαταριών συνολικής τάσης 9 V.

Η καλωδίωση έγινε με καλώδια ηλεκτρονικών jumper wire 1.6 mm και για την γκαραζόπορτα χρησιμοποιήθηκε κινητήρας metal gearmotor 25mm, 100 rpm, 9-12V πλήρους περιστροφής.



Figure 2 Metal Gearmotor 25mm - 100rpm

Για τον έλεγχο του κινητήρα χρειάστηκε να χρησιμοποιηθεί το ολοκληρωμένο L293D ως οδηγός για τον κινητήρα.

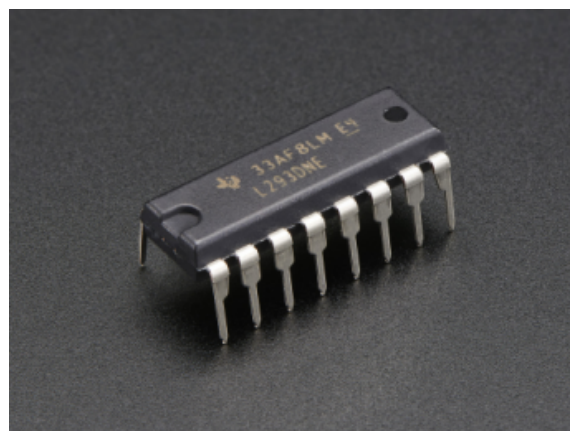


Figure 3 L293D Motor Driver

Εφαρμογή Arduino για Απομακρυσμένο έλεγχο θύρας μέσω Ασύρματου δικτύου Wi-Fi.

Επίσης στη συνδεσμολογία ένα από τα κυριότερα εξαρτήματα που χρησιμοποιήθηκε είναι η πλακέτα ESP8266 Wi-Fi Module η οποία δίνει της εντολές και μέσω του motor driver ενεργοποιείται ο κινητήρας.

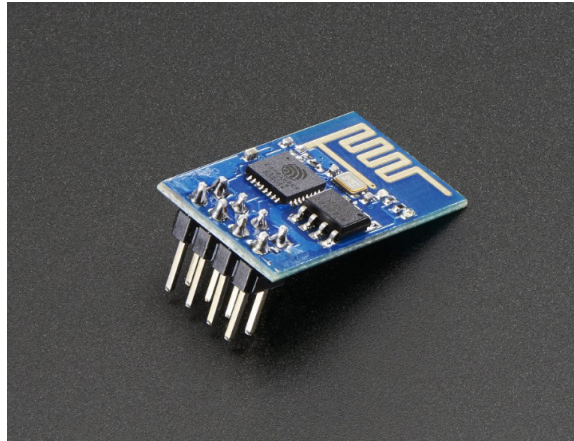


Figure 4 ESP8266 Wi-Fi Module

Και τέλος επειδή το ESP8266 χρειάζεται χαμηλή τάση τροφοδοσίας της τάξης των 3.3V χρησιμοποιήθηκαν δύο ρυθμιστές τάσεις για τη σύνδεση του με το motor driver για τον έλεγχο της περιστροφής του κινητήρα στις αντίστοιχες περιπτώσεις.



Figure 5 Voltage regulator LD1117 - 3.3V

Εφαρμογή Arduino για Απομακρυσμένο έλεγχο θύρας μέσω Ασύρματου δικτύου Wi-Fi.

## 3. ΠΛΗΡΟΦΟΡΙΕΣ ARDUINO

### 3.1 Τι είναι το Arduino

Το Arduino είναι μια απλή μητρική πλακέτα ανοικτού κώδικα με ενσωματωμένο μικροελεγκτή και εισόδους/εξόδους, η οποία μπορεί να προγραμματιστεί με τη γλώσσα Wiring (ουσιαστικά πρόκειται για τη γλώσσα προγραμματισμού C++ και ένα σύνολο από βιβλιοθήκες, υλοποιημένες επίσης στην C++ ).

Το Arduino μπορεί να χρησιμοποιηθεί για την ανάπτυξη ανεξάρτητων διαδραστικών αντικειμένων αλλά και να συνδεθεί με υπολογιστή μέσω προγραμμάτων σε Processing, Max/MSP, Pure Data, SuperCollider. Οι περισσότερες εκδόσεις του Arduino μπορούν να αγοραστούν προ-συναρμολογημένες και το διάγραμμα και οι πληροφορίες για το υλικό είναι ελεύθερα διαθέσιμα για αυτούς που θέλουν να συναρμολογήσουν το Arduino μόνοι τους.

### 3.2 Πλακέτα

Μία πλακέτα Arduino αποτελείται από ένα μικροελεγκτή Atmel AVR (ATmega328 και ATmega168 στις νεότερες εκδόσεις, ATmega8 στις παλαιότερες) και συμπληρωματικά εξαρτήματα για την διευκόλυνση του χρήστη στον προγραμματισμό και την ενσωμάτωση του σε άλλα κυκλώματα.

Όλες οι πλακέτες περιλαμβάνουν ένα γραμμικό ρυθμιστή τάσης 5V και έναν κρυσταλλικό ταλαντωτή 16MHz (ή κεραμικό αντηχητή σε κάποιες παραλλαγές). Ο μικροελεγκτής είναι από κατασκευής προγραμματισμένος με ένα bootloader, έτσι ώστε να μην χρειάζεται εξωτερικός προγραμματιστής.

Σε εννοιολογικό επίπεδο, στην χρήση του Arduino software stack, όλα τα boards προγραμματίζονται με μία RS-232 σειριακή σύνδεση, αλλά ο τρόπος που επιτυγχάνεται αυτό διαφέρει σε κάθε hardware εκδοχή. Οι σειριακές πλάκες Arduino περιέχουν ένα απλό level shifter κύκλωμα για την μετατροπή του σήματος επιπέδου RS-232 σε TTL. Τα σημερινά Arduino προγραμματίζονται μέσω USB· αυτό καθίσταται δυνατό μέσω της εφαρμογής προσαρμογών chip USB-to-Serial όπως το FTDI FT232. Κάποιες παραλλαγές, όπως το Arduino mini και το ανεπίσημο Boarduino, χρησιμοποιούν ένα αφαιρούμενο USB-to-Serial καλώδιο ή board, Bluetooth ή άλλες μεθόδους.

Ο πίνακας Arduino εκθέτει τα περισσότερα microcontroller I/O pins για χρήση από άλλα κυκλώματα. Τα Diecimila, Duemilanove και το τρέχον Uno παρέχουν 14 ψηφιακά I/O pins, έξι από τα οποία μπορούν να παράγουν pulse-width διαμορφωμένα σήματα, και 6 αναλογικά δεδομένα. Αυτά τα pins βρίσκονται στην κορυφή του πίνακα μέσω female headers 0.1 ιντσών (2,2mm).

### 3.3 Λογισμικό Ανάπτυξης

Το ολοκληρωμένο περιβάλλον ανάπτυξης (IDE) του Arduino είναι μία εφαρμογή γραμμένη σε Java, που λειτουργεί σε πολλές πλατφόρμες, και προέρχεται από το IDE για τη γλώσσα προγραμματισμού Processing και το σχέδιο Wiring. Έχει σχεδιαστεί για να εισαγάγει τον προγραμματισμό στους νέους που δεν είναι εξοικειωμένοι με την ανάπτυξη λογισμικού. Περιλαμβάνει ένα πρόγραμμα επεξεργασίας κώδικα με χαρακτηριστικά όπως είναι η επισήμανση σύνταξης και ο συνδυασμός αγκύλων και είναι επίσης σε θέση να μεταγλωττίζει και να φορτώνει προγράμματα στην πλακέτα με ένα μόνο κλικ. Δεν υπάρχει συνήθως καμία ανάγκη να επεξεργαστείτε αρχεία make ή να τρέξετε προγράμματα σε ένα περιβάλλον γραμμής εντολών. Ένα πρόγραμμα ή κώδικας που γράφτηκε για Arduino ονομάζεται σκίτσο (sketch).

Τα Arduino προγράμματα είναι γραμμένα σε C ή C++. Το Arduino IDE έρχεται με μια βιβλιοθήκη λογισμικού που ονομάζεται "Wiring" από το πρωτότυπο σχέδιο Wiring γεγονός που καθιστά πολλές κοινές λειτουργίες εισόδου/εξόδου πολύ πιο εύκολες. Οι χρήστες πρέπει μόνο να ορίσουν δύο λειτουργίες για να κάνουν ένα πρόγραμμα κυκλικής εκτέλεσης:

-setup(): μία συνάρτηση που τρέχει μία φορά στην αρχή του προγράμματος η οποία αρχικοποιεί τις ρυθμίσεις

-loop(): μία συνάρτηση η οποία καλείται συνέχεια μέχρι η πλακέτα να απενεργοποιηθεί

Το IDE του Arduino χρησιμοποιεί το GNU toolchain και το AVR Libc για να μεταγλωττίζει προγράμματα και το avrdude για να φορτώνει προγράμματα στην πλακέτα.

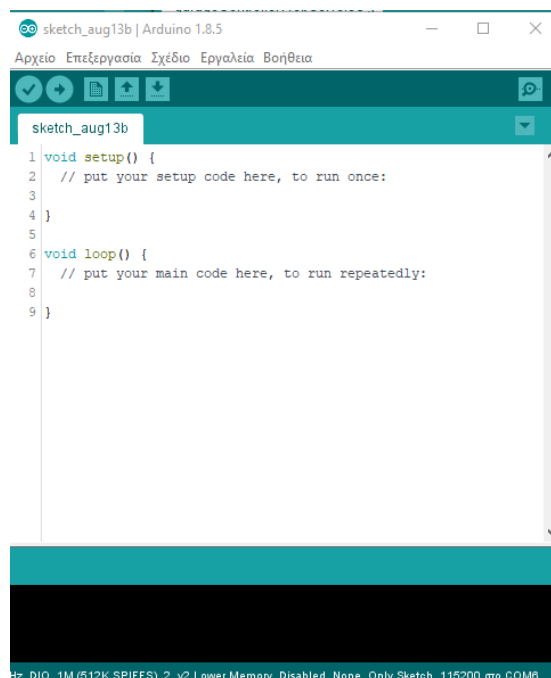


Figure 6 Στιγμιότυπο λογισμικού Arduino



### 3.4 Arduino UNO

Για την παρούσα πτυχιακή εργασία επιλέχθηκε η πλακέτα Arduino Uno το οποίο φέρει τον επεξεργαστή ATmega328P, έχει 14 ψηφιακές εισόδους/εξόδους και 6 αναλογικές, τάση λειτουργίας στα 5 V, υποδοχή για RS-232 σειριακή σύνδεση, είσοδο εξωτερικής τροφοδοσίας, ISCP header, κουμπί reset και λαμπάκι led συνδεδεμένο με την ψηφιακή είσοδο 13.

Table 1 Χαρακτηριστικά επεξεργαστή Arduino Uno

CPU Clock	16 MHz
EEPROM	1Kbyte
SRAM	2 Kbytes
Flash memory	32 Kbytes

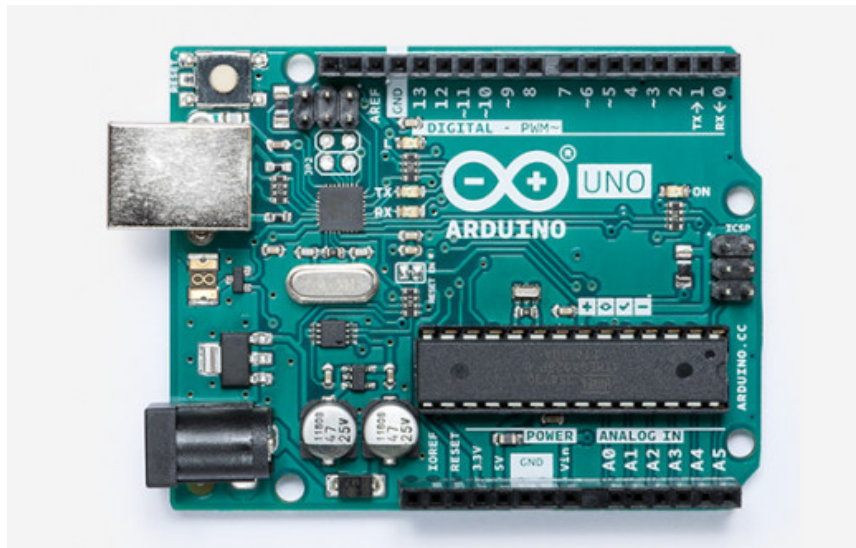


Figure 7 Arduino Uno

### 3.5 ESP8266 Wi-Fi Module

Για να δοθεί η δυνατότητα απομακρυσμένου ελέγχου στον ελεγκτή χρησιμοποιήθηκε το Wi-Fi module ESP8266. Ουσιαστικά πρόκειται για μια πλακέτα που προστίθεται στην κυρίως πλακέτα Arduino Uno και επεκτείνει τις δυνατότητες της. Η συγκεκριμένη πλακέτα δίνει τη δυνατότητα επικοινωνίας μέσω Wi-Fi από οποιαδήποτε ασύρματο δίκτυο.

Το ESP8266 WiFi Module πρόκειται για ένα αυτόνομο ολοκληρωμένο κύκλωμα SOC με ενσωματωμένη στοίβα πρωτοκόλων TCP/IP που δίνει σε κάθε μικροελεγκτή δυνατότητα πρόσβασης σε ασύρματο δίκτυο. Το ESP8266 μπορεί είτε να φιλοξενήσει κάποια εφαρμογή, είτε να μεταφορτώσει όλες τις λειτουργίες δικτύου από κάποιον άλλον επεξεργαστή. Κάθε πλακέτα ESP8266 βγαίνει προγραμματισμένη με ένα σετ εντολών υλικολογισμικού(AT), πράγμα το οποίο παρέχει ευκολία σύνδεσης στο Arduino δίνοντας άμεση πρόσβαση σε ασύρματο δίκτυο, καθιστώντας το εξαιρετικά αποδοτικό.

Αυτή η συσκευή διαθέτει επαρκή ικανότητα επεξεργασίας και αποθήκευσης που επιτρέπει την ενσωμάτωσή της σε αισθητήρες και άλλες συσκευές εφαρμογών μέσω των ακίδων GPIO με ελάχιστη ανάγκη προγραμματισμού και μεταφόρτωσης κατά τη διάρκεια της εκτέλεσης. Ο υψηλός βαθμός ολοκλήρωσης της πλακέτας επιτρέπει τη χρήση της χωρίς την ανάγκη επιπλέον εξωτερικών κυκλωμάτων, συμπεριλαμβανομένης της μικρής επιφάνειας PCB που καταλαμβάνει.

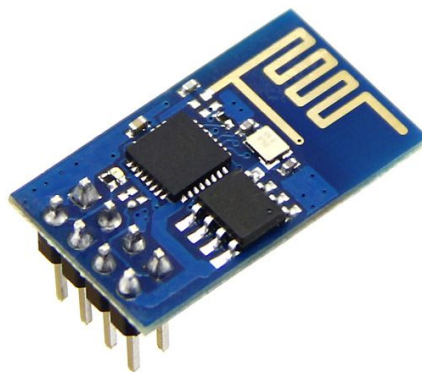


Figure 8 ESP8266 Wi-Fi Module

Σημείωση: Το ESP8266 χρειάζεται χαμηλή τροφοδοσία 3V.

Table 2 Χαρακτηριστικά του ESP8266 Wi-Fi module

802.11 b/g/n
Wi-Fi Direct (P2P), soft-AP
Ενσωματωμένη στοίβα πρωτοκόλλων TCP/IP
Ενσωματωμένος διακόπτης TR, balun, LNA, ενισχυτής ισχύος και δίκτυο αντιστοίχισης
Ενσωματωμένα PLL, ρυθμιστές, DCXO και μονάδες διαχείρισης ισχύος
+19,5dBm ισχύς εξόδου σε λειτουργία 802.11b
Ρεύμα διαρροής <10uA
Ενσωματωμένο CPU 32-bit χαμηλής ισχύος ικανό για χρήση ως επεξεργαστής εφαρμογών
SDIO 1.1 / 2.0, SPI, UART
STBC, 1 × 1 MIMO, 2 × 1 MIMO
Συσσωμάτωση A-MPDU & A-MSDU και διάστημα προστασίας 0,4ms
Χρόνος ετοιμότητας για μετάδοση πακέτων σε <2ms
Κατανάλωση ισχύος σε αναμονή <1.0mW (DTIM3)

Εφαρμογή Arduino για Απομακρυσμένο έλεγχο θύρας μέσω Ασύρματου δικτύου Wi-Fi.

## 4. ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ARDUINO

### 4.1 Αρχές Προγραμματισμού

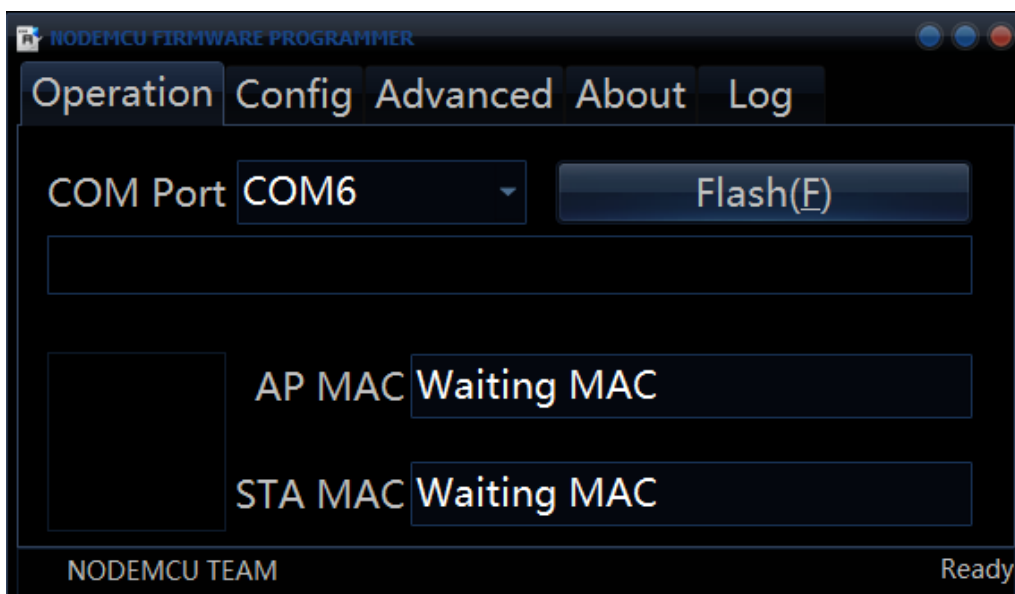
Όπως γράφτηκε παραπάνω η γλώσσα προγραμματισμού για το Arduino είναι μια διαφορετική υλοποίηση της γλώσσας Wiring. Κάποιες βιβλιοθήκες που χρησιμοποιήθηκαν είναι η <ESP8266WiFi.h> και η <ESP8266WebServer.h>. Αρχικά σε οποιοδήποτε sketch δημιουργηθεί απαιτείται να υπάρχουν δύο συναρτήσεις, η setup() και η loop(). Η πρώτη εκτελείται μία φορά στην αρχή, ουσιαστικά περιέχει κάποιες αναγκαίες αρχικοποιήσεις, ενώ η loop εκτελείται συνεχόμενα κατά τη διάρκεια της λειτουργίας του Arduino. Στην αρχιτεκτονική της συγκεκριμένης εργασίας χρησιμοποιήθηκε και η σειριακή σύνδεση του Arduino με προσωπικό υπολογιστή και το serial monitor του IDE χρησιμοποιήθηκε για λόγους debugging αλλά και για τον έλεγχο της ορθής λειτουργίας του αυτοματισμού. Παρ' όλα αυτά δεν είναι αναγκαία η ύπαρξη του προσωπικού υπολογιστή για τη λειτουργία της εγκατάστασης.

### 4.2 Αναβάθμιση Firmware ESP8266

Αρχικά θα χρειαστεί αναβάθμιση του Firmware για τη σωστή λειτουργία του ESP8266 Module.

Βήμα 1: Βρίσκουμε και κατεβάζουμε το ESP8266 Flash Software που βρίσκεται διαθέσιμο στο παρακάτω link.

<https://github.com/nodemcu/nodemcu-flasher>



Εφαρμογή Arduino για Απομακρυσμένο έλεγχο θύρας μέσω Ασύρματου δικτύου Wi-Fi.

Figure 9 ESP8266 Flash Software

Βήμα 2: Βρίσκουμε τα αρχεία του ESP8266 Firmware από το παρακάτω link.

[https://github.com/espressif/ESP8266\\_AT](https://github.com/espressif/ESP8266_AT)

Βήμα 3: Συνδέουμε το ESP σε Flash Mode.

Για να επιτευχθεί αυτό θα πρέπει το GPIO 0 να είναι συνδεδεμένο στο GND.

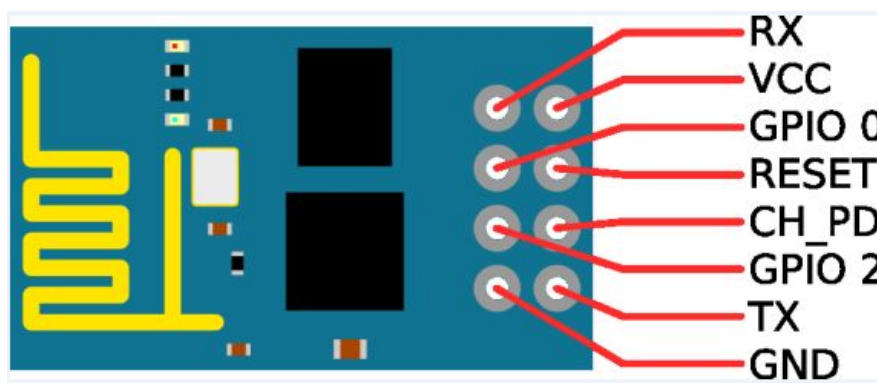


Figure 10 Ακίδες του ESP8266 Wi-Fi module

Στη συνέχεια συνδέουμε την πλακέτα του Arduino μέσω της σειριακής θύρας και ακολουθούμε τα βήματα στο παράθυρο του λογισμικού.

### 4.3 Εντολές Επικοινωνίας Wi-Fi

Ο τρόπος που ορίστηκε για να δέχεται απομακρυσμένες εντολές το Arduino είναι να στηθεί ουσιαστικά ένας web server.

Αρχικά χρειάζονται δύο μεταβλητές για να μπορέσει η πλακέτα ESP8266 να συνδεθεί σε δίκτυο, το όνομα του ασύρματου δικτύου και ο κωδικός του, όπως φαίνεται παρακάτω.

```
char* ssid = "HOME";  
char* password = "*****";
```

Στον server δόθηκε η τοπική IP 192.168.1.103. Με τις εντολές που ακολουθούν ξεκινάει τη λειτουργία του ο web server. Αυτό γίνεται μέσα στην συνάρτηση setup

που όπως αναφέρθηκε παραπάνω εκτελείται μία φορά κατά την ενεργοποίηση του Arduino.

```
WiFi.begin(ssid,password);
Serial.begin(115200);
while(WiFi.status()!=WL_CONNECTED)
{
  Serial.print(".");
  delay(500);
}
Serial.println("");
Serial.print("IP Address: ");
Serial.println(WiFi.localIP());

server.on("/",[](){server.send_P(200,"text/html", webpage);});
server.on("/openstate",getOpenState);
server.on("/closestate",getCloseState);
server.on("/stopstate",getStopState);
server.begin();
}
```

Μέσα στη συνάρτηση loop που εκτελείται συνέχεια περιμένουμε να συνδεθεί κάποιος client στο server μας και η συγκεκριμένη συνάρτηση χειρίζεται τις απαιτήσεις στον διακομιστή.

```
server.handleClient();
```

#### 4.4 Εντολές Λειτουργίας Γκαραζόπορτας

Στις συναρτήσεις void openDoor, void closeDoor και void stopDoor υλοποιήθηκε ο έλεγχος του κινητήρα της γκαραζόπορτας.

Η θέση της γκαραζόπορτας αναπαρίσταται στον κώδικα με τη μεταβλητή door-state η οποία παίρνει τιμές OPEN, CLOSE και STOP Για να υλοποιήσουμε το άνοιγμα της γκαραζόπορτας χρησιμοποιείται η εντολή digital.write(output2, HIGH); digital.write(output0,LOW); και με μια καθυστέρηση 1 δευτερολέπτου καλείται η συνάρτηση stopDoor όπου σταματάει το άνοιγμα της γκαραζόπορτας. Αντίστοιχα για το κλείσιμο της γκαραζόπορτας με τις ίδιες εντολές αλλάζοντας τις τιμές των ψηφιακών εξόδων από HIGH σε LOW και αντίστροφα και με μια καθυστέρηση 1,2 δευτερολέπτων σταματάει το κλείσιμο της γκαραζόπορτας. Και τέλος για για το σταμάτημα της γκαραζόπορτας θέτουμε στο LOW τις τιμές και των δύο μεταβλητών όπως φαίνεται παρακάτω:

```
void openDoor()
{
  digitalWrite(output2,HIGH);
  digitalWrite(output0,LOW);
  delay(1000); // time in ms
  stopDoor();
}
```

```
void closeDoor()
{
  digitalWrite(output2,LOW);
  digitalWrite(output0,HIGH);
  delay(1200);
  stopDoor();
}

void stopDoor()
{
  digitalWrite(output2,LOW);
  digitalWrite(output0,LOW);
}
```

Μετά το διάβασμα της εντολής, συντάσσεται η κατάσταση της γκαραζόπορτας όπου καλείται αντίστοιχα και η εκάστοτε συνάρτηση για την λειτουργία της:

```
void getOpenState()
{
  openDoor();
  String door_state = "OPEN";
  server.send(200,"text/plain", door_state);
}

void getCloseState()
{
  closeDoor();
  String door_state = "CLOSE";
  server.send(200,"text/plain", door_state);
}

void getStopState()
{
  stopDoor();
  String door_state = "STOP";
  server.send(200,"text/plain", door_state);
}
```

## 4.5 Ιστοσελίδα Εφαρμογής

Μέσα στον κώδικα του Arduino έχει γραφτεί και ένα τμήμα κώδικα html για τη διαδικτυακή σελίδα της εφαρμογής. Όπως φαίνεται παρακάτω έχουν δημιουργηθεί τρία κουμπιά, το καθένα από τα οποία κάνει κάποιες λειτουργίες μόλις πατηθούν.

```
<h1>Garage Motor Controller</h1>
<p></p><p></p>
<p> Door Status: <span id="door-state">__</span> </p>
<p><button id="button1" class = "button" onclick="myFunctionOpen();changeColor1();"> OPEN
</button></p>
<p><button id="button2" class = "button" onclick="myFunctionClose();changeColor2();"> CLOSE
</button></p>
```



```
<p><button id="button3" class = "button" onclick="myFunctionStop();changeColor3();"> STOP  
</button></p>
```

Με το πάτημα του κουμπιού OPEN καλείται η συνάρτηση changeColor1 η οποία χρωματίζει το κουμπί πράσινο κάνοντας τα άλλα δύο κουμπιά στο αρχικό γκρι χρώμα όπως φαίνεται παρακάτω.

```
function changeColor1(){  
  document.getElementById("button1").style.backgroundColor = "#43a047";  
  document.getElementById("button2").style.backgroundColor = "#77878A";  
  document.getElementById("button3").style.backgroundColor = "#77878A";  
};
```

Αντίστοιχα με το πάτημα του κουμπιού CLOSE καλείται η συνάρτηση changeColor2 η οποία χρωματίζει το κουμπί κόκκινο κάνοντας τα άλλα δύο κουμπιά στο αρχικό γκρι χρώμα.

```
function changeColor2(){  
  document.getElementById("button2").style.backgroundColor = "#e53935";  
  document.getElementById("button1").style.backgroundColor = "#77878A";  
  document.getElementById("button3").style.backgroundColor = "#77878A";  
};
```

Και τέλος με το πάτημα του κουμπιού STOP καλείται η συνάρτηση changeColor3 η οποία χρωματίζει το κουμπί κίτρινο κάνοντας τα άλλα δύο κουμπιά στο αρχικό γκρι χρώμα.

```
function changeColor3(){  
  document.getElementById("button3").style.backgroundColor = "#ffeb3b";  
  document.getElementById("button1").style.backgroundColor = "#77878A";  
  document.getElementById("button2").style.backgroundColor = "#77878A";  
};
```

Άλλη μια συνάρτηση που καλείται τη στιγμή που θα πατηθεί το εκάστοτε κουμπί είναι αυτή που θα στείλει και θα αλλάξει την κατάσταση της γκαραζόπορτας με μια μέθοδο απαντώντας στο web server. Όπως βλέπουμε παρακάτω για το κουμπί OPEN η κατάσταση της θύρας αλλάζει σε ανοιχτή.

```
function myFunctionOpen()  
{  
  console.log("button was clicked!");  
  var xhr = new XMLHttpRequest();  
  var url = "/openstate";  
  xhr.onreadystatechange = function() {  
    if (this.readyState == 4 && this.status == 200) {  
      document.getElementById("door-state").innerHTML = this.responseText;  
    }  
  };  
  xhr.open("GET", url, true);  
  xhr.send();  
};
```

Αντίστοιχα με το πάτημα του κουμπιού CLOSE η παρακάτω συνάρτηση καλείται στέλνοντας την κατάσταση της γκαραζόπορτας που έχει κλείσει.

```
function myFunctionClose(){
  console.log("button was clicked!");
  var xhr = new XMLHttpRequest();
  var url = "/closestate";
  xhr.onreadystatechange = function() {
    if (this.readyState == 4 && this.status == 200) {
      document.getElementById("door-state").innerHTML = this.responseText;
    }
  };
  xhr.open("GET", url, true);
  xhr.send();
};
```

Και τέλος με το πάτημα του κουμπιού STOP στέλνεται η κατάσταση της γκαραζόπορτας η οποία έχει σταματήσει.

```
function myFunctionStop(){
  console.log("button was clicked!");
  var xhr = new XMLHttpRequest();
  var url = "/stopstate";
  xhr.onreadystatechange = function() {
    if (this.readyState == 4 && this.status == 200) {
      document.getElementById("door-state").innerHTML = this.responseText;
    }
  };
  xhr.open("GET", url, true);
  xhr.send();
};
```

Με τα παρακάτω Event Listener εκτελούνται οι αντίστοιχες συναρτήσεις κάθε φορά που ανανεώνεται ο φυλλομετρητής. Η προσθήκη αυτή έγινε για τσεκάρισμα της λειτουργίας του κινητήρα της γκαραζόπορτας κατά την αρχική είσοδο στη σελίδα της εφαρμογής.

```
document.addEventListener('DOMContentLoaded', myFunctionOpen, false);
document.addEventListener('DOMContentLoaded', myFunctionClose, false);
document.addEventListener('DOMContentLoaded', myFunctionStop, false);
```

## 5. ΠΛΗΡΟΦΟΡΙΕΣ ANDROID

### 5.1 Τι είναι το Android

Το Android είναι λειτουργικό σύστημα για συσκευές κινητής τηλεφωνίας το οποίο τρέχει τον πυρήνα του λειτουργικού Linux. Αρχικά αναπτύχθηκε από την Google και αργότερα από την Open Handset Alliance. Επιτρέπει στους κατασκευαστές λογισμικού να συνθέτουν κώδικα με την χρήση της γλώσσας προγραμματισμού Java, ελέγχοντας την συσκευή μέσω βιβλιοθηκών λογισμικού ανεπτυγμένων από την Google.

Η πρώτη παρουσίαση της πλατφόρμας Android έγινε στις 5 Νοεμβρίου 2007, παράλληλα με την ανακοίνωση της ίδρυσης του οργανισμού Open Handset Alliance, μιας κοινοπραξίας 48 τηλεπικοινωνιακών εταιριών, εταιριών λογισμικού καθώς και κατασκευής hardware, οι οποίες είναι αφιερωμένες στην ανάπτυξη και εξέλιξη ανοιχτών προτύπων στις συσκευές κινητής τηλεφωνίας. Η Google δημοσίευσε το μεγαλύτερο μέρος του κώδικα του Android υπό τους όρους της Apache License, μιας ελεύθερης άδειας λογισμικού. Το λογότυπο για το λειτουργικό σύστημα Android είναι ένα ρομπότ σε χρώμα πράσινου μήλου και σχεδιάστηκε από τη γραφίστρια Irina Blok.

### 5.2 Περιβάλλον Android

Το προεπιλεγμένο περιβάλλον χρήστη του Android βασίζεται στην άμεση διεπαφή, χρησιμοποιώντας εισόδους αφής, που αντιστοιχούν σε πραγματικές δράσεις, όπως το σύρσιμο, το άγγιγμα, το τσίμπημα, και το ανεστραμμένο τσίμπημα. Η απόκριση προς την είσοδο του χρήστη έχει σχεδιαστεί για να είναι άμεση και παρέχει ένα interface αφής, συχνά χρησιμοποιώντας τις δυνατότητες δόνηση της συσκευής να παρέχει απτική ανάδραση στο χρήστη. Εσωτερικό υλικό, όπως επιταχυνσιόμετρα, γυροσκόπια και αισθητήρες εγγύτητας χρησιμοποιούνται από ορισμένες εφαρμογές για να ανταποκριθεί σε πρόσθετες ενέργειες του χρήστη, για παράδειγμα, τη ρύθμιση της οθόνης, από κατακόρυφο σε οριζόντιο προσανατολισμό ανάλογα με το πώς η συσκευή είναι προσανατολισμένη, ή επιτρέποντας στο χρήστη να κατευθύνει ένα όχημα σε ένα παιχνίδι αγώνων με την περιστροφή της συσκευής, προσομοιώνοντας τον έλεγχο του τιμονιού.

Οι Android συσκευές εκκινούν στην αρχική οθόνη, η οποία είναι το κύριο κομμάτι της πλοήγησης και παρέχει πληροφορίες σχετικά με τη συσκευή, η οποία είναι παρόμοια με την επιφάνεια εργασίας που υπάρχει στους υπολογιστές. Οι Android αρχικές οθόνες συνήθως αποτελούνται από εικονίδια εφαρμογών και widgets. Τα εικονίδια εφαρμογών ξεκινάνε τη σχετική εφαρμογή, ενώ τα widgets εμφανίζουν ζωντανά, κάποιο περιεχόμενο αυτόματης ενημέρωσης, όπως η πρόβλεψη του καιρού, εισερχόμενα e-mail του χρήστη, ή ένα live feed ειδήσεων. Η αρχική οθόνη μπορεί να

αποτελείται από αρκετές σελίδες που η χρήστης μπορεί να πλοηγηθεί σέρνοντάς τις πίσω και μπρος, αν και η αρχική οθόνη του Android είναι σε μεγάλο βαθμό προσαρμόσιμη, επιτρέποντας στο χρήστη να επιλέξει την εμφάνιση και την αίσθηση της συσκευής με τις προτιμήσεις τους. Εφαρμογές τρίτων που είναι διαθέσιμες στο Google Play και άλλα καταστήματα εφαρμογών μπορούν να διαφοροποιήσουν ολοκληρωτικά την αρχική οθόνη, ακόμα και να μιμηθούν την εμφάνιση άλλων λειτουργικών συστημάτων, όπως τα Windows Phone. Οι περισσότεροι κατασκευαστές, και μερικοί πάροχοι τηλεπικοινωνιών, προσαρμόζουν την εμφάνιση και την αίσθηση των Android συσκευών τους ώστε να διαφοροποιηθούν από τους ανταγωνιστές τους.

Κατά μήκος της κορυφής της οθόνης υπάρχει μια γραμμή κατάστασης που εμφανίζει πληροφορίες σχετικά με τη συσκευή και την συνδεσιμότητα του. Αυτή η γραμμή κατάστασης μπορεί να "τραβηχτεί" προς τα κάτω για να αποκαλύψει μια οθόνη κατάστασης όπου κάποιες εφαρμογές εμφανίζουν σημαντικές πληροφορίες ή ενημερώσεις, όπως πρόσφατα εισερχόμενα e-mail ή SMS κειμένου, με έναν τρόπο που δεν διακόπτεται άμεσα ή παρενοχλείτε ο χρήστης. Οι ειδοποιήσεις παραμένουν μέχρι να αναγνωστούν ή να απορριφθούν.

### 5.3 Λογισμικό Android

Η ανάπτυξη λογισμικού Android είναι η διαδικασία με την οποία δημιουργούνται νέες εφαρμογές για το λειτουργικό σύστημα Android. Οι εφαρμογές αναπτύσσονται συνήθως στη γλώσσα προγραμματισμού Java, χρησιμοποιώντας το Android Software Development Kit, αλλά και άλλα εργαλεία ανάπτυξης που είναι διαθέσιμα πλέον όπως το Android Studio το οποίο χρησιμοποιήθηκε στην παρούσα εργασία.

Το Android Studio είναι ένα ολοκληρωμένο προγραμματιστικό περιβάλλον (IDE) για ανάπτυξη εφαρμογών στην πλατφόρμα Android. Ανακοινώθηκε στις 16 Μαΐου 2013 στο συνέδριο Google I/O από την Google Product Manager, Katherine Chou. Το Android Studio είναι διαθέσιμο ελεύθερα με την άδεια Apache License 2.0.

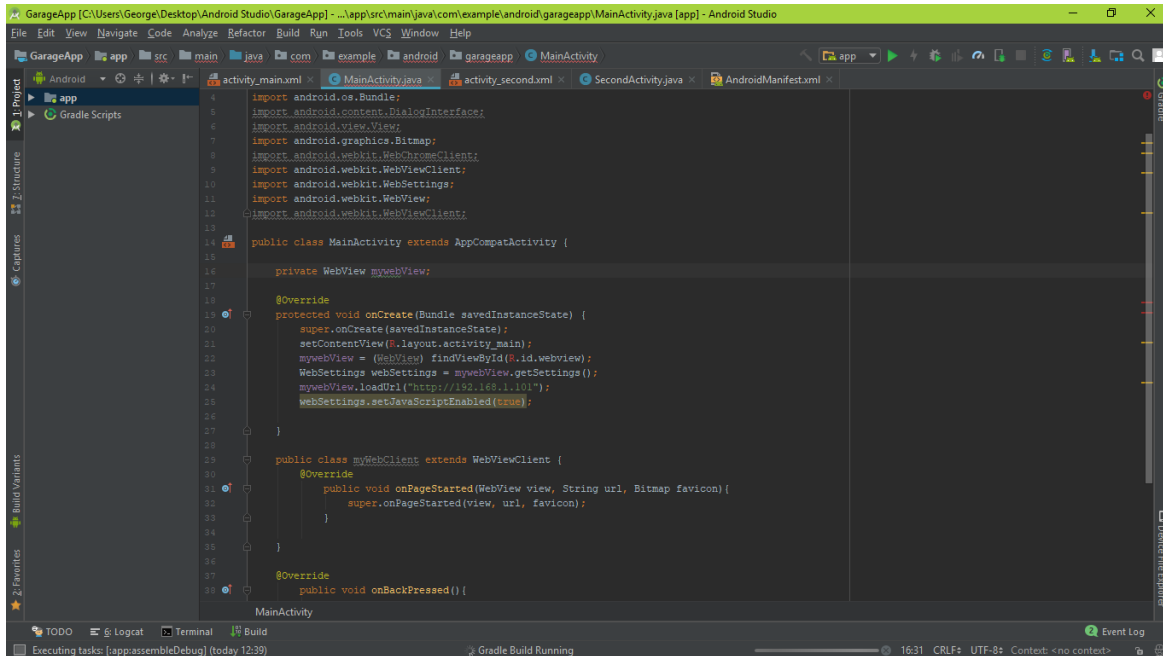
Το Android Studio ήταν διαθέσιμο σε πρώιμο στάδιο για προεπισκόπηση ξεκινώντας από την έκδοση 0.1 τον Μάιο του 2013, έπειτα ξεκίνησε το δοκιμαστικό στάδιο από την έκδοση 0.8 που βγήκε τον Ιούνιο του 2014. Η πρώτη σταθερή έκδοση βγήκε το Δεκέμβριο του 2014, με την έκδοση 1.0.

Βασισμένο στο λογισμικό της JetBrains' IntelliJ IDEA, το Android Studio σχεδιάστηκε αποκλειστικά για προγραμματισμό Android. Είναι διαθέσιμο για Windows, Mac OS X και Linux, και αντικατέστησε τα Eclipse Android Development Tools (ADT) ως το κύριο IDE της Google για ανάπτυξη εφαρμογών Android.

Το Android Studio περιλαμβάνει ένα ολοκληρωμένο σύνολο εργαλείων ανάπτυξης. Αυτά είναι ένα πρόγραμμα εντοπισμού σφαλμάτων, βιβλιοθήκες, έναν εξομοιωτή συσκευής AVD manager, code editor, graphical και text editor, components tree, παράθυρο με properties, κ.α..

Εφαρμογή Arduino για Απομακρυσμένο έλεγχο θύρας μέσω Ασύρματου δικτύου Wi-Fi.

Οι εφαρμογές Android «χτίζονται» σε ένα αρχείο με επέκταση .apk, και αποθηκεύονται στο φάκελο ~/app/build/outputs/apk/debug στο Android Studio. Με το αρχείο .apk, η εφαρμογή μπορεί να διανεμηθεί και να εγκατασταθεί στο ΛΣ Android.



```
4 import android.os.Bundle;
5 import android.content.DialogInterface;
6 import android.view.View;
7 import android.graphics.Bitmap;
8 import android.webkit.WebChromeClient;
9 import android.webkit.WebViewClient;
10 import android.webkit.WebSettings;
11 import android.webkit.WebView;
12 import android.webkit.WebViewClient;
13
14 public class MainActivity extends AppCompatActivity {
15
16     private WebView myWebView;
17
18     @Override
19     protected void onCreate(Bundle savedInstanceState) {
20         super.onCreate(savedInstanceState);
21         setContentView(R.layout.activity_main);
22         myWebView = (WebView) findViewById(R.id.webview);
23         WebSettings webSettings = myWebView.getSettings();
24         myWebView.loadUrl("http://192.168.1.101");
25         webSettings.setJavaScriptEnabled(true);
26     }
27
28     public class myWebClient extends WebViewClient {
29         @Override
30         public void onPageStarted(WebView view, String url, Bitmap favicon) {
31             super.onPageStarted(view, url, favicon);
32         }
33     }
34
35     @Override
36     public void onBackPressed() {
37
38     }
39 }
```

Figure 11 Στιγμιότυπο οθόνης Android Studio

Εφαρμογή Arduino για Απομακρυσμένο έλεγχο θύρας μέσω Ασύρματου δικτύου Wi-Fi.

## 6. ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΗΣ ANDROID

### 6.1 Αρχές Προγραμματισμού

Όπως γράφτηκε και νωρίτερα, η ανάπτυξη εφαρμογής για λειτουργικό σύστημα Android γίνεται συνήθως με τη γλώσσα προγραμματισμού Java. Χρησιμοποιούνται βέβαια συγκεκριμένες βιβλιοθήκες που έχουν δημιουργηθεί για αυτό το σκοπό. Οι βιβλιοθήκες αυτές θα εξηγηθούν εκτενέστερα με τα επιμέρους κομμάτια τους..

Αυτό που πρέπει να τονιστεί εδώ είναι τα απαραίτητα δομικά στοιχεία που αποτελούν τον πηγαίο κώδικα μίας εφαρμογής για λειτουργικό σύστημα Android.

1) Η δραστηριότητα (activity) είναι ένα συστατικό της εφαρμογής που παρέχει μια οθόνη με την οποία οι χρήστες μπορούν να αλληλοεπιδρούν με σκοπό να κάνουν κάτι, όπως να καλέσουν τηλέφωνο, να τραβήξουν μια φωτογραφία, να στείλουν ένα e-mail, ή να δουν ένα χάρτη. Σε κάθε δραστηριότητα δίνεται ένα παράθυρο στο οποίο θα εμφανιστεί η διεπαφή του χρήστη. Το παράθυρο συνήθως γεμίζει την οθόνη, αλλά μπορεί να είναι μικρότερο από την οθόνη και να επικαλύπτει άλλα παράθυρα της εφαρμογής.

Μια εφαρμογή αποτελείται συνήθως από πολλαπλές δραστηριότητες που είναι συνδεδεμένες. Μία δραστηριότητα σε μια εφαρμογή ορίζεται ως η «κύρια» δραστηριότητα(MainActivity), η οποία παρουσιάζεται στο χρήστη κατά την έναρξη της εφαρμογής. Κάθε δραστηριότητα μπορεί στη συνέχεια να αρχίσει μια άλλη δραστηριότητα, προκειμένου να εκτελέσει διάφορες ενέργειες. Κάθε φορά που μια νέα δραστηριότητα ξεκινά, η προηγούμενη δραστηριότητα έχει σταματήσει, αλλά το σύστημα διατηρεί τη δραστηριότητα σε μια στοίβα.

Να σημειωθεί εδώ ότι το γραφικό περιβάλλον κάθε δράσης (activity) περιγράφεται σε ένα xml αρχείο το οποίο ανακαλείται με την εκκίνηση κάθε διαφορετικής δράσης.

2) Κάθε εφαρμογή πρέπει να έχει ένα αρχείο AndroidManifest.xml (με ακριβώς αυτό το όνομα) στον root κατάλογο του. Το αρχείο δήλωσης παρουσιάζει ουσιώδεις πληροφορίες σχετικά με την εφαρμογή στο σύστημα Android, πληροφορίες που το σύστημα πρέπει να έχει για να μπορέσει να τρέξει οποιοδήποτε κώδικα της εφαρμογής. Μεταξύ άλλων, το αρχείο περιέχει τα εξής:

Το όνομα της εφαρμογής ως ένα μοναδικό αναγνωριστικό για την εφαρμογή.

Δηλώνει ποια δικαιώματα έχει η εφαρμογή ώστε να αποκτήσει πρόσβαση σε προστατευμένα μέρη του API και να μπορεί να αλληλοεπιδρά με άλλες εφαρμογές.

Δηλώνει, επίσης, τα δικαιώματα που τρίτοι πρέπει να έχουν προκειμένου να αλληλοεπιδράσουν με τα στοιχεία της εφαρμογής.

Δηλώνει το ελάχιστο επίπεδο του API του Android που απαιτεί η εφαρμογή.

Απαριθμεί τις βιβλιοθήκες τις οποίες χρειάζεται η εφαρμογή για τη λειτουργία της.

Επίσης εκεί δηλώνεται και η κύρια activity της εφαρμογής που τρέχει με το ξεκίνημά της αλλά και όλες οι υπόλοιπες activities. Όπως έχει αναφερθεί και προηγουμένως στο παράρτημα υπάρχει ολόκληρος ο πηγαίος κώδικας τόσο της εφαρμογής android όσο και του Arduino.

3) Η γραφική διεπαφή χρήστη για ένα Android app είναι χτισμένη χρησιμοποιώντας την ιεραρχία View και ViewGroup αντικειμένων. Τα αντικείμενα View είναι συνήθως UI widgets όπως κουμπιά ή πεδία κειμένου και τα ViewGroup αντικείμενα είναι περιέκτες αντικειμένων που καθορίζουν τον τρόπο που εμφανίζονται τα εμπεριεχόμενα αντικείμενα View, όπως σε ένα πλέγμα ή μια κατακόρυφη λίστα.

Το Android παρέχει ένα λεξιλόγιο XML που αντιστοιχεί στις υποκατηγορίες View και ViewGroup ώστε να μπορείτε να οριστεί το User Interface χρησιμοποιώντας την XML.

4) Στο φάκελο app της εφαρμογής υπάρχει ο υποφάκελος res που αναφέρεται στο resources, δηλαδή σε κάποια στοιχεία που μπορούμε να αποθηκεύσουμε και να είναι προσβάσιμα από την εφαρμογή κατά την εφαρμογή του layout μίας δραστηριότητας. Εδώ βρίσκεται ο φάκελος layout που είναι αποθηκευμένα τα xml αρχεία που αναφέρθηκαν προηγουμένως. Επίσης υπάρχει και ο φάκελος values που περιέχει αρχεία όπως το strings.xml μέσα στο οποίο ορίζονται όλα τα αλφαριθμητικά που χρησιμοποιούνται στην εφαρμογή.

5) Η πρόθεση (intent) είναι μια αφηρημένη περιγραφή μιας λειτουργίας που πρέπει να εκτελεστεί. Μπορεί να χρησιμοποιηθεί με την εντολή startActivity να ξεκινήσει μια δραστηριότητα, ή με την εντολή broadcastIntent για να σταλεί σε οποιοδήποτε BroadcastReceiver στοιχείο, και με τις εντολές startService (Intent) ή bindService (Intent, ServiceConnection, int) για να επικοινωνεί με μια υπηρεσία παρασκηνίου.

Πιο σημαντική χρήση του Intent είναι κατά την έναρξη των δραστηριοτήτων, όπου μπορεί να θεωρηθεί ως συνδετικός κρίκος μεταξύ των δραστηριοτήτων. Είναι βασικά μια παθητική δομή δεδομένων που κατέχουν μια αφηρημένη περιγραφή της δράσης που πρέπει να εκτελεστεί.



## 6.2 Η εφαρμογή

Η εφαρμογή αυτή θα λειτουργεί ως παράθυρο επικοινωνίας με την υπάρχουσα διαδικτυακή εφαρμογή που είναι εγκατεστημένη στο web server.

Αρχικά στο αρχείο AndroidManifest.xml θα χρειαστεί μια δήλωση που αφορά δικαιωμάτων χρήσης δικτύου όπως φαίνεται παρακάτω:

```
<uses-permission android:name="android.permission.INTERNET"
```

Έπειτα στο αρχείο activity\_main.xml χρησιμοποιήθηκε ένα αντικείμενο WebView το οποίο θα εμφανίζει στην οθόνη την σελίδα της διαδικτυακής εφαρμογής με τις λειτουργίες της.

```
<WebView  
    android:id="@+id/webview"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent"  
    tools:layout_editor_absoluteX="0dp"  
    tools:layout_editor_absoluteY="0dp"/>
```

## 6.3 Κύρια Δραστηριότητα

Στην κύρια δραστηριότητα MainActivity.java στις ήδη υπάρχουσες κλάσεις χρειάστηκε να εισάγουμε κάποιες ακόμη οι οποίες είναι αναγκαίες για την επικοινωνία της εφαρμογής με το διαδίκτυο. Αυτές φαίνονται παρακάτω με έντονη γραμματοσειρά.

```
package com.example.android.garageapp;  
  
import android.support.v7.app.AppCompatActivity;  
import android.os.Bundle;  
import android.content.DialogInterface;  
import android.graphics.Bitmap;  
import android.webkit.WebChromeClient;  
import android.webkit.WebSettings;  
import android.webkit.WebView;  
import android.webkit.WebViewClient;
```

Στη συνέχεια μέσα στην κλάση της κύριας δραστηριότητας δηλώθηκε η μεταβλητή WebView η οποία χρησιμοποιήθηκε για όλες τις παρακάτω παραμετροποιήσεις και με το `loadUrl("http://192.168.1.103");` φορτώνει τη διαδικτυακή σελίδα της εφαρμογής μέσω της διεύθυνσης IP που έχει οριστεί από πριν.

```
public class MainActivity extends AppCompatActivity {
    private WebView mywebView;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        mywebView = (WebView) findViewById(R.id.webview);
        WebSettings webSettings = mywebView.getSettings();
        mywebView.loadUrl("http://192.168.1.103");
        webSettings.setJavaScriptEnabled(true);
    }
}
```

Η παρακάτω κλάση σκοπό έχει την υποστήριξη της σελίδας κατά την αρχική φόρτωση ώστε να εμφανίζει κανονικά τη σελίδα της εφαρμογής, το url και οποιαδήποτε εικονίδια συμπεριλαμβάνονται.

```
public class myWebClient extends WebViewClient {
    @Override
    public void onPageStarted(WebView view, String url, Bitmap favicon){
        super.onPageStarted(view, url, favicon);
    }
}
```

Με τις παρακάτω γραμμές κώδικα αποτρέπεται η διακοπή της σελίδας της εφαρμογής σε περίπτωση που πατηθεί το κουμπί “πίσω” (BACK).

```
@Override
    public void onBackPressed(){
        if (mywebView.canGoBack()) {
            mywebView.goBack();
        }else{
            super.onBackPressed();
        }
    }
}
```

## 6.4 Έλεγχος Λειτουργίας Γκαραζόπορτας

Σε αυτή την οθόνη ο χρήστης μπορεί να επιβλέψει και να ελέγξει την κατάσταση της γκαραζόπορτας.

Η θέση της γκαραζόπορτας αναπαρίσταται από μια κατάσταση που όταν είναι ανοιχτή: Door Status → OPEN, όταν είναι κλειστή: Door Status → CLOSE ενώ όταν είναι σταματημένη: Door Status → STOP.

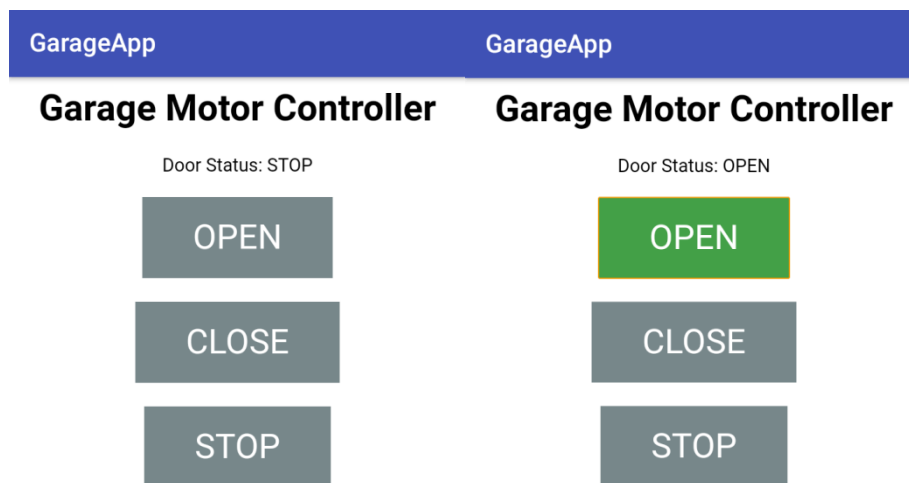


Figure 12 Εφαρμογή Android (Αρχική – Ανοιχτή)

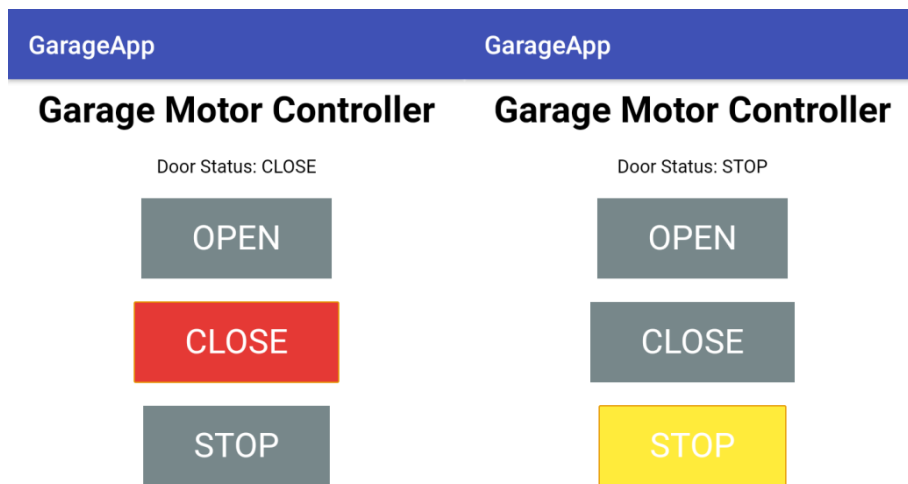


Figure 13 Εφαρμογή Android (Κλειστή - Σταματημένη)

## 7. ΣΥΜΠΕΡΑΣΜΑΤΑ

### 7.1 Διαπιστώσεις

Με την ολοκλήρωση της διπλωματικής εργασίας μπορούμε να καταλήξουμε σε κάποια χρήσιμα συμπεράσματα.

Όπως αναμένονταν αρχικά, δημιουργήθηκε ένα σύστημα επικοινωνίας μεταξύ των δύο opensource πλατφορμών Android και Arduino. Η επικοινωνία γίνεται τοπικά από το ασύρματο δίκτυο όσο και απομακρυσμένα με τη χρήση εξωτερικής IP. Η ταχύτητα της σύνδεσης είναι εξαιρετικά ικανοποιητική για τέτοιου είδους εφαρμογές, το ίδιο και η ποιότητά της.

Προκύπτει ότι είναι δυνατόν, με βάση την παρούσα εργασία, να δημιουργηθεί πληθώρα εφαρμογών με τη χρήση των δύο πλατφορμών. Εφαρμογές που να άπτονται στον ευρύτερο τομέα του Internet of Things. Είναι δυνατόν το κόστος να κρατηθεί πολύ χαμηλά και να δημιουργηθούν ανταγωνιστικές εφαρμογές. Επίσης δεν απαιτούνται εξειδικευμένες γνώσεις προγραμματισμού ή ηλεκτρονικών.

### 7.2 Εξέλιξη

Στηριζόμενοι στο πρωτόκολλο επικοινωνίας της παρούσας διπλωματικής εργασίας, μπορεί να αναπτυχθεί μία οικογένεια συσκευών συνδεδεμένων που θα απλοποιούν τη ρουτίνα του καθημερινού χρήστη.

Το επόμενο βήμα είναι να δημιουργηθεί οικογένεια συσκευών που θα ελέγχονται από Arduino, οι οποίες θα μπορούν να συνδεθούν σε εφαρμογή Android. Ο χρήστης θα μπορεί να παραμετροποιήσει πλήρως το δίκτυό του και να επιλέξει αυτός ποιες συσκευές να συνδέσει στο δίκτυό του. Αυτό θα ήταν ένα βήμα πιο κοντά σε εμπορική εφαρμογή, που απαιτείται ευελιξία, αφού ο κάθε χρήστης θα θέλει ένα δίκτυο συσκευών προσαρμοσμένο στις δικές του ανάγκες.

Εφαρμογή Arduino για Απομακρυσμένο έλεγχο θύρας μέσω Ασύρματου δικτύου Wi-Fi.

## 8. ΒΙΒΛΙΟΓΡΑΦΙΑ

- [1] [www.arduino.cc](http://www.arduino.cc)
- [2] [el.wikipedia.org/wiki/Android](http://el.wikipedia.org/wiki/Android)
- [3] [el.wikipedia.org/wiki/Android\\_Studio](http://el.wikipedia.org/wiki/Android_Studio)
- [4] [developer.android.com/training/index.html](http://developer.android.com/training/index.html)
- [5] [https://en.wikipedia.org/wiki/Internet\\_of\\_things](https://en.wikipedia.org/wiki/Internet_of_things)
- [6] [el.wikipedia.org/wiki/Arduino](http://el.wikipedia.org/wiki/Arduino)
- [7] <https://www.w3schools.com>
- [8] <https://github.com/nodemcu/nodemcu-flasher>
- [9] [https://github.com/espressif/ESP8266\\_AT](https://github.com/espressif/ESP8266_AT)
- [10] <https://nurdspace.nl/ESP8266>

Εφαρμογή Arduino για Απομακρυσμένο έλεγχο θύρας μέσω Ασύρματου δικτύου Wi-Fi.



## 9. ΠΑΡΑΡΤΗΜΑ

### 1. Μακέτα

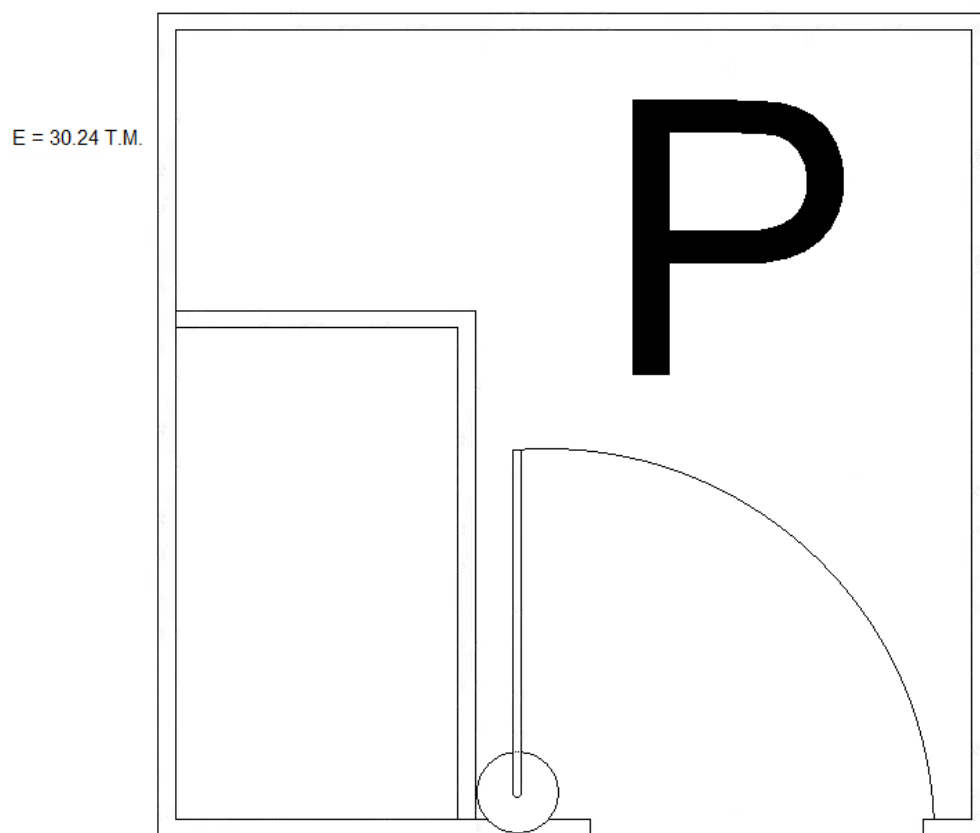


Figure 14 Κάτοψη εγκατάστασης

## 2. Κώδικας Arduino

### [GarageControllerWebServerJS]

```
#include <ESP8266WiFi.h> // βιβλιοθήκη για σύνδεση ESP8266 σε δίκτυο
#include <ESP8266WebServer.h> // βιβλιοθήκη για να στηθεί web server

ESP8266WebServer server; // αρχικοποίηση web server

uint8_t output2 = 2; // δήλωση μεταβλητής για το GPIO2 του ESP8266
uint8_t output0 = 0; // δήλωση μεταβλητής για το GPIO2 του ESP8266

char* ssid = "HOME"; // όνομα του δικτύου σύνδεσης
char* password = "f561c33f1a"; // κωδικός του δικτύου

char webpage[] PROGMEM = R"=====( // πίνακας χαρακτήρων που θα περιέχει
την ιστοσελίδα η οποία αποθηκεύεται στη μνήμη FLASH και είναι γραμμένη ως
Raw string

<html> // αρχή κώδικα ιστοσελίδας
<head>
<style> html {
  font-family: Helvetica;
  display: inline-block;
  margin: 0px auto;
  text-align: center;
}
.button {
  background-color: #77878A; //αρχικό χρώμα κουμπιών
  border: none;
  color: white;
  padding: 16px 40px;
  text-decoration: none;
  font-size: 30px;
  margin: 2px;
  cursor: pointer;
}

</style>
</head>
<body>

<h1>Garage Motor Controller</h1>
<p></p><p></p>
<p> Door Status: <span id="door-state">__</span> </p> //μεταβλητή
για την κατάσταση της γκαραζόπορτας

<p><button id="button1" class = "button"
onclick="myFunctionOpen();changeColor1();"> OPEN </button></p>
<p><button id="button2" class = "button"
onclick="myFunctionClose();changeColor2();"> CLOSE </button></p>
<p><button id="button3" class = "button"
onclick="myFunctionStop();changeColor3();"> STOP </button></p>

</body>
<script> //script συναρτήσεων αλλαγής χρώματος κουμπιών και
συναρτήσεων λειτουργίας γκαραζόπορτας στο πάτημα του εκάστοτε κουμπιού
```

## Εφαρμογή Arduino για Απομακρυσμένο έλεγχο θύρας μέσω Ασύρματου δικτύου Wi-Fi.

```
function changeColor1(){
  document.getElementById("button1").style.backgroundColor = "#43a047";
  document.getElementById("button2").style.backgroundColor = "#77878A";
  document.getElementById("button3").style.backgroundColor = "#77878A";
};

function changeColor2(){
  document.getElementById("button2").style.backgroundColor = "#e53935";
  document.getElementById("button1").style.backgroundColor = "#77878A";
  document.getElementById("button3").style.backgroundColor = "#77878A";
};

function changeColor3(){
  document.getElementById("button3").style.backgroundColor = "#ffeb3b";
  document.getElementById("button1").style.backgroundColor = "#77878A";
  document.getElementById("button2").style.backgroundColor = "#77878A";
};

function myFunctionOpen()
{
  console.log("button was clicked!");
  var xhr = new XMLHttpRequest();
  var url = "/openstate";
  xhr.onreadystatechange = function() {
    if (this.readyState == 4 && this.status == 200) {
      document.getElementById("door-state").innerHTML = this.responseText;
    }
  };
  xhr.open("GET", url, true);
  xhr.send();
};

function myFunctionClose(){
  console.log("button was clicked!");
  var xhr = new XMLHttpRequest();
  var url = "/closestate";
  xhr.onreadystatechange = function() {
    if (this.readyState == 4 && this.status == 200) {
      document.getElementById("door-state").innerHTML = this.responseText;
    }
  };
  xhr.open("GET", url, true);
  xhr.send();
};

function myFunctionStop(){
  console.log("button was clicked!");
  var xhr = new XMLHttpRequest();
  var url = "/stopstate";
  xhr.onreadystatechange = function() {
    if (this.readyState == 4 && this.status == 200) {
      document.getElementById("door-state").innerHTML = this.responseText;
    }
  };
  xhr.open("GET", url, true);
  xhr.send();
};

document.addEventListener('DOMContentLoaded', myFunctionOpen, false);
document.addEventListener('DOMContentLoaded', myFunctionClose, false);
```

## Εφαρμογή Arduino για Απομακρυσμένο έλεγχο θύρας μέσω Ασύρματου δικτύου Wi-Fi.

```
document.addEventListener('DOMContentLoaded', myFunctionStop, false);
</script>
</html>                                     // τέλος κώδικα ιστοσελίδας
)=====";

void setup()
{
  pinMode(output2, OUTPUT);
  pinMode(output0, OUTPUT);

  digitalWrite(output2 , LOW);
  digitalWrite(output0 , LOW);

  WiFi.begin(ssid,password); // σύνδεση στο δίκτυο Wi-Fi
  Serial.begin(115200);      // έναρξη σειριακής επικοινωνίας
  while(WiFi.status() !=WL_CONNECTED) //έλεγχος σύνδεσης στο δίκτυο
  {
    Serial.print(".");
    delay(500);
  }
  Serial.println("");
  Serial.print("IP Address: ");
  Serial.println(WiFi.localIP()); // εμφάνιση IP στη σειριακή

  server.on("/", [] () {server.send_P(200,"text/html", webpage);});
  // η προκαθορισμένη διαδρομή για ανταπόκριση απο την συσκευή ESP8266
  // στέλνοντας την ιστοσελίδα που είναι αποθηκευμένη στη μνήμη Flash
  server.on("/openstate",getOpenState);
  server.on("/closestate",getCloseState);
  server.on("/stopstate",getStopState);
  server.begin();
}

void loop()
{
  server.handleClient();
}

void openDoor() // συνάρτηση για το άνοιγμα της πόρτας
{
  digitalWrite(output2,HIGH);
  digitalWrite(output0,LOW);
  delay(1000); // time in ms
  stopDoor();
}

void closeDoor() // συνάρτηση για το κλείσιμο της πόρτας
{
  digitalWrite(output2,LOW);
  digitalWrite(output0,HIGH);
  delay(900);
  stopDoor();
}

void stopDoor() // συνάρτηση για το σταμάτημα της πόρτας
{
  digitalWrite(output2,LOW);
  digitalWrite(output0,LOW);
}
```

## Εφαρμογή Arduino για Απομακρυσμένο έλεγχο θύρας μέσω Ασύρματου δικτύου Wi-Fi.

```
void getOpenState() // συνάρτηση για την αποστολή της κατάστασης της
πόρτας όταν ανοίξει
{
  openDoor();
  String door_state = "OPEN";
  server.send(200,"text/plain", door_state);
}

void getCloseState() // συνάρτηση για την αποστολή της κατάστασης της
πόρτας όταν κλείσει
{
  closeDoor();
  String door_state = "CLOSE";
  server.send(200,"text/plain", door_state);
}

void getStopState() // συνάρτηση για την αποστολή της κατάστασης της
πόρτας όταν σταματήσει
{
  stopDoor();
  String door_state = "STOP";
  server.send(200,"text/plain", door_state);
}
```

Εφαρμογή Arduino για Απομακρυσμένο έλεγχο θύρας μέσω Ασύρματου δικτύου Wi-Fi.

### 3. Κώδικας Android

[AndroidManifest.xml>]

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.android.garageapp">

    <uses-permission android:name="android.permission.INTERNET" />

    <uses-permission android:name="android.permission.GET_ACCOUNTS" />
    <uses-permission android:name="android.permission.READ_PROFILE" />
    <uses-permission android:name="android.permission.READ_CONTACTS" />

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
        <activity android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER"
            />
            </intent-filter>
        </activity>
        <activity android:name=".SecondActivity"></activity>
    </application>

</manifest>
```

## [MainActivity.java]

```
package com.example.android.garageapp;

import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.content.DialogInterface;
import android.graphics.Bitmap;
import android.webkit.WebChromeClient;
import android.webkit.WebSettings;
import android.webkit.WebView;
import android.webkit.WebViewClient;

public class MainActivity extends AppCompatActivity {

    private WebView mywebView;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        mywebView = (WebView) findViewById(R.id.webview);
        WebSettings webSettings = mywebView.getSettings();
        mywebView.loadUrl("http://192.168.1.103");    <!-- διεύθυνση IP
όπου βρίσκεται η διαδικτυακή σελίδα της εφαρμογής στο ESP8266 -->
        webSettings.setJavaScriptEnabled(true);

    }

    public class myWebClient extends WebViewClient {
        @Override
        public void onPageStarted(WebView view, String url, Bitmap
favicon){
            super.onPageStarted(view, url, favicon);
        }
    }

    @Override
    public void onBackPressed(){
        if (mywebView.canGoBack()) {
            mywebView.goBack();
        }else{
            super.onBackPressed();
        }
    }
}
```



[activity\_main.xml]

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity"
    tools:layout_editor_absoluteY="81dp">

    <WebView
        android:id="@+id/webview"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        tools:layout_editor_absoluteX="0dp"
        tools:layout_editor_absoluteY="0dp"/>

</android.support.constraint.ConstraintLayout>
```

Εφαρμογή Arduino για Απομακρυσμένο έλεγχο θύρας μέσω Ασύρματου δικτύου Wi-Fi.

Εφαρμογή Arduino για Απομακρυσμένο έλεγχο θύρας μέσω Ασύρματου δικτύου Wi-Fi.