

ΒΙΒΛΙΟΘΗΚΗ
ΠΑΝΕΠΙΣΤΗΜΙΟΥ ΙΩΑΝΝΙΝΩΝ



026000265546



(145)
ΜΠΛΕ

**ΑΣΥΡΜΑΤΟ ΣΥΣΤΗΜΑ
ΑΥΤΟΜΑΤΙΣΜΟΥ ΚΑΙ ΤΗΛΕΜΕΤΡΙΑΣ
ΕΡΕΥΝΗΤΙΚΟΥ ΕΡΓΑΣΤΗΡΙΟΥ**

ΜΕΤΑΠΤΥΧΙΑΚΗ ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

**ΕΥΣΤΡΑΤΙΟΣ ΤΖΟΒΑΡΑΣ
Ηλεκτρονικός Μηχανικός**

**Επίβλεψη: Αναπληρωτής Καθηγητής Ν. Μάνθος
Εργαστήριο Φυσικής Υψηλών Ενεργειών**

**ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ
ΣΤΙΣ ΣΥΓΧΡΟΝΕΣ ΗΛΕΚΤΡΟΝΙΚΕΣ ΤΕΧΝΟΛΟΓΙΕΣ**

**ΤΜΗΜΑ ΦΥΣΙΚΗΣ
ΠΑΝΕΠΙΣΤΗΜΙΟ ΙΩΑΝΝΙΝΩΝ**

ΙΩΑΝΝΙΝΑ ΜΑΡΤΙΟΣ 2007



Περίληψη

Στην παρούσα εργασία, παρουσιάζεται η ανάπτυξη ενός συστήματος αυτοματισμού και τηλεμετρίας, βασισμένο σε ασύρματη επικοινωνία.

Το σύστημα βασίζεται στην ασύρματη δικτύωση Wi - Fi και στο πρωτόκολλο 802.11. Αποτελείται από έναν κόμβο πρόσβασης, τρεις ασύρματους σταθμούς διασύνδεσης Wi - Fi καθώς και τρεις σταθμούς προσάρτησης αισθητηρίων. Τέλος, περιλαμβάνει μια ασύρματη κάμερα που μπορεί να λειτουργήσει στα πρότυπα 802.11b/g του πρωτοκόλλου 802.11.

Ο κόμβος πρόσβασης είναι υπεύθυνος για τη δημιουργία του ασύρματου δικτύου Wi - Fi και της διευθυνσιοδότησης των σταθμών που είναι συνδεδεμένοι. Οι τρεις ασύρματοι σταθμοί διασύνδεσης, συνδέουν τους τρεις σταθμούς προσάρτησης αισθητηρίων με τον κόμβο πρόσβασης, μέσω ενσύρματης σειριακής επικοινωνίας (RS-232). Στους σταθμούς προσάρτησης αισθητηρίων μπορούν να προσαρτηθούν αισθητήρια αναλογικά, ψηφιακά, αισθητήρια τύπου I²C καθώς και ενεργοποιητές. Τέλος, η κάμερα του συστήματος προσφέρει και οπτική επιτήρηση του χώρου που θέλουμε να ελέγξουμε, πέρα από την επιτήρηση που μας προσφέρουν τα συνδεδεμένα αισθητήρια.

Ο έλεγχος του συστήματος είναι φιλικός προς τον χρήστη και πραγματοποιείται εύκολα μέσω μιας σελίδας υπερκειμένου (HTML). Στη σελίδα αυτή, υπάρχουν πλαίσια για τα αισθητήρια τύπου I²C, για τα αναλογικά αλλά και για τα ψηφιακά. Υπάρχει ακόμη ένα πλαίσιο, το οποίο δείχνει την εικόνα της κάμερας, προσφέροντας έτσι αδιάκοπη οπτική επιτήρηση.

Το σύστημα έχει σαν στόχο την επιτήρηση ενός χώρου με δυνατότητα παρέμβασης. Και όλα αυτά γρήγορα και εύκολα μέσω της σελίδας υπερκειμένου και από οποιοδήποτε σημείο του πλανήτη, αρκεί να υπάρχει διαθέσιμη πρόσβαση στον Παγκόσμιο Ιστό (Διαδίκτυο).



Abstract

The current thesis demonstrates the development of an automation and telemetry system.

The system is based on the wireless network Wi – Fi and the protocol 802.11. It consists of an access point, three wirelesses Wi – Fi modules and also from three stations of sensor's attachment and a wireless camera. The last one is able to operate using the 802.11b/g standards.

The access point is responsible for the creation of the wireless network Wi – Fi and for the addressing of the connected nodes. The three wireless Wi – Fi modules, interconnect the three stations of sensor's attachment with the access point, via a wire serial communication (RS-232) port. Analog, digital and I²C sensors and in addition actuators can be connected to the stations of sensor's attachment. Finally, the system's camera offers a visual surveillance of the area that must be controlled, beyond the surveillance offered by the attached sensors.

The system's user interface is based on an HTML page, which includes frames for the I²C, analog and digital sensors. Finally, the camera's view is displayed in a separate additional frame, offering a continuous visual inspection.

The system's aim is the surveillance of an area with the capability of user's intervention. This can be easily and quickly achieved through a web page, from anywhere provided that there is an available access to the World Wide Web.

Ευχαριστίες

Στο σημείο αυτό, θα ήθελα να ευχαριστήσω τους ανθρώπους που συνέβαλαν σημαντικά με τη βοήθειά τους, στην πραγματοποίηση αυτής της εργασίας.

Τον επιβλέποντα της διπλωματικής μου εργασίας Αν Καθηγητή κ. Νικόλαο Μάνθο, μέλος του Εργαστηρίου Φυσικής Υψηλών Ενεργειών του Πανεπιστημίου Ιωαννίνων, για την αμέριστη βοήθεια, καθοδήγηση και τις καίριες του παρατηρήσεις, καθ' όλη τη διάρκεια εκπόνησης της εργασίας μου.

Τα μέλη της εξεταστικής επιτροπής, κ. Ιωάννη Παπαδόπουλο και κ. Ιωάννη Ευαγγέλου, για τη βοήθεια που μου προσέφεραν και προπαντός για την εμπιστοσύνη που μου έδειξαν.

Τους φίλους και συναδέρφους Ασημάκη Ασημίδη, Βάϊο Πάτρα, Στέφανο Κρολβό και Ευστάθιο Μπλέτσα για τη βοήθεια, το πραγματικό ενδιαφέρον και τις πολύτιμες συμβουλές που μου παρείχαν σε όλη τη διάρκεια αυτής της εργασίας.

Επίσης θα ήθελα να ευχαριστήσω ιδιαίτερω τους πολύ καλούς φίλους και συναδέλφους Γεώργιο Σιδηρόπουλο και Αλέξανδρο Παπακώστα, χάρη στη συνεργασία, στην αδιάκοπη και πολύτιμη βοήθεια καθώς και στις σημαντικότερες συμβουλές των οποίων, στάθηκε δυνατή η ολοκλήρωση της παρούσας εργασίας.

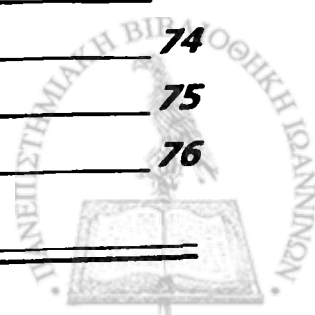
Τέλος, θα ήθελα να ευχαριστήσω τη Θεανώ, το Βασίλη, το Νίκο, το Γιάννη, το Λάμπρο, τους γονείς μου Μιχάλη και Παρασκευή και την υπόλοιπη οικογένειά μου, για την ηθική αλλά και υλική υποστήριξη, καθ' όλη τη διάρκεια της παρούσας εργασίας και όχι μόνο!



ΠΕΡΙΕΧΟΜΕΝΑ

<i>Περίληψη</i>	1
<i>Abstract</i>	2
<i>Ευχαριστίες</i>	3
ΠΕΡΙΕΧΟΜΕΝΑ	4
<i>Εισαγωγή</i>	7
ΚΕΦΑΛΑΙΟ 1^ο.	9
<i>Εισαγωγή στις ασύρματες επικοινωνίες Wi-Fi και οι εφαρμογές τους</i>	9
ΚΕΦΑΛΑΙΟ 2^ο.	17
<i>Το ασύρματο σύστημα αυτοματισμού και τηλεμετρίας ερευνητικού εργαστηρίου</i>	17
2.1 Ο κόμβος πρόσβασης (Access Point).	17
2.1.1 Η ασφάλεια του συστήματος	18
2.1.2 Οι βασικές ρυθμίσεις	19
2.1.3 Οι ρυθμίσεις ασύρματου δικτύου	19
2.1.4 Οι ρυθμίσεις τοπικού δικτύου	20
2.1.5 Οι ρυθμίσεις προώθησης/ενεργοποίησης θυρών	21
2.2 Οι ασύρματοι σταθμοί διασύνδεσης (wireless modules)	22
2.2.1 Ρυθμίσεις ενσύρματου δικτύου	25
2.2.2 Ρυθμίσεις ασφάλειας	25
2.2.3 Ρυθμίσεις σειριακής θύρας 1 και 2	25
2.2.4 Ρυθμίσεις ασυρμάτου δικτύου	26
2.3 Η κάμερα του συστήματος (IP camera)	28
2.3.1 Snapshot.	29
2.3.2 Client Settings.	30
2.3.3 Configuration	30
2.4 Ο σταθμός προσάρτησης αισθητηρίων.	32
2.5 Ο μικροελεγκτής του συστήματος	35
2.5.1 Οι θύρες γενικά	36
2.5.2 Ο μετατροπέας αναλογικού σε ψηφιακό	37
2.5.3 Ο δίαυλος I²C	38
2.5.4 Η σύγχρονη/ασύγχρονη σειριακή επικοινωνία	40
2.6 Τα αισθητήρια του συστήματος	42

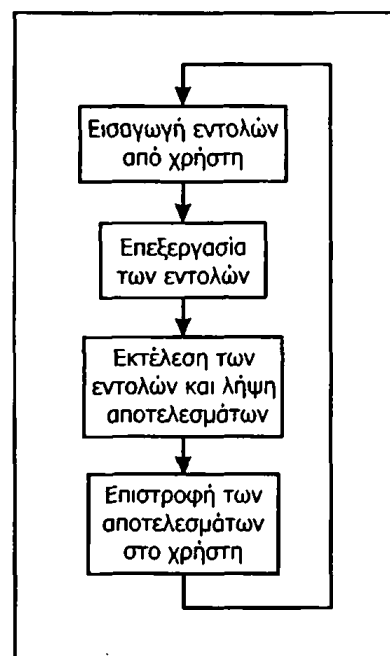
ΚΕΦΆΛΑΙΟ 3^ο.	45
<i>Το λογισμικό του συστήματος.</i>	45
<i>3.1 Το λογισμικό της εφαρμογής σε γλώσσα προγραμματισμού Assembly.</i>	45
<i>3.2 Το λογισμικό του μικροελεγκτή σε γλώσσα Assembly</i>	47
<i>3.2.1 Το λογισμικό του μικροελεγκτή για το δίαυλο I²C</i>	48
<i>3.2.1 Το λογισμικό του μικροελεγκτή για τον μετατροπέα ADC</i>	49
<i>3.2.3 Το λογισμικό του μικροελεγκτή για τη θύρα GPIO</i>	51
<i>3.2.4 Το λογισμικό του μικροελεγκτή για τον έλεγχο σκανδαλισμού της κάμερας</i>	52
<i>3.2.5 Το λογισμικό του μικροελεγκτή για τη σύγχρονη/ασύγχρονη σειριακή επικοινωνία</i>	52
<i>3.2.6 Το λογισμικό του μικροελεγκτή για τους χαρακτήρες ASCII</i>	53
<i>3.2.7 Το λογισμικό του μικροελεγκτή για τον WDT</i>	53
<i>3.3 Το λογισμικό της εφαρμογής σε HTML.</i>	54
<i>3.4 Η γλώσσα προγραμματισμού C και το πρότυπο CGI.</i>	58
<i>3.4.1 Εισαγωγή</i>	58
<i>3.4.2 Χρήση του προτύπου CGI της γλώσσας C, στο λογισμικό του συστήματος.</i>	59
<i>3.5 Ο εξυπηρετητής του συστήματος.</i>	62
<i>3.5.1 Ο εξυπηρετητής διαδικτύου.</i>	63
<i>3.5.2 Εφαρμογή εικονικής θύρας.</i>	65
ΚΕΦΆΛΑΙΟ 4^ο.	67
<i>Λειτουργία και αποτίμηση συστήματος.</i>	67
<i>4.1 Η λειτουργία των τμημάτων του υλικού.</i>	67
<i>4.1.1 Ο κόμβος πρόσβασης</i>	68
<i>4.1.2 Ο σταθμός ασύρματης διασύνδεσης</i>	68
<i>4.1.3 Η κάμερα IP</i>	69
<i>4.1.4 Ο σταθμός διασύνδεσης αισθητηρίων</i>	69
<i>4.1.4 Ο διακομιστής του συστήματος</i>	70
<i>4.2 Η λειτουργία των τμημάτων του λογισμικού.</i>	71
<i>4.2.1 Ο κώδικας HTML</i>	71
<i>4.2.1.1 Πλαίσιο I²C</i>	74
<i>4.2.1.2 Πλαίσιο ADC</i>	75
<i>4.2.1.3 Πλαίσιο GPI/O</i>	76



4.2.1.4 Πλαίσιο σκανδαλισμού της κάμερας	76
4.2.2 Ο κώδικας του προτύπου CGI	76
4.2.3 Ο κώδικας της Assembly	80
4.3 Αποτίμηση του συστήματος.	83
4.4 Μελλοντική εξέλιξη του συστήματος	84
Αναφορές - Βιβλιογραφία	86
A.1 Ο κώδικας της γλώσσας HTML.	88
A.1.1 Κεντρική ιστοσελίδα	88
A.1.2 Πλαίσιο τίτλου	88
A.1.3 Πλαίσιο διασυνδέσεων	89
A.1.4 Πλαίσιο κάμερας	92
A.1.5 Πλαίσιο διαύλου F^2C	93
A.1.6 Πλαίσιο μετατροπέα ADC	94
A.1.6 Πλαίσιο θύρας GPI/O	95
A.1.7 Πλαίσιο ελέγχου σκανδαλισμού της κάμερας A	97
A.1.8 Πλαίσιο ελέγχου σκανδαλισμού της κάμερας B	98
A.2 Ο κώδικας της γλώσσας C με χρήση του προτύπου CGI.	98
A.2.1 Ο κώδικας του διαύλου F^2C	98
A.2.2 Ο κώδικας του μετατροπέα ADC	103
A.2.3 Ο κώδικας της θύρας GPI/O	107
A.2.4 Ο κώδικας ελέγχου της κάμερας	112
A.3 Ο κώδικας της γλώσσας Assembly.	114
ΠΑΡΑΡΤΗΜΑ Β	126
B.1 Ο σταθμός προσάρτησης αισθητηρίων και ενεργοποιητών.	126

αυτόν. Τα αισθητήρια και οι ενεργοποιητές συνδέονται στο σταθμό προσάρτησης αισθητηρίων, και μέσω αυτού επικοινωνούν με το υπόλοιπο σύστημα ασύρματα. Την ασύρματη επικοινωνία του συστήματος, διαχειρίζεται ο κόμβος πρόσβασης, ο οποίος συνδέεται ασύρματα με τους σταθμούς προσάρτησης αισθητηρίων και ενσύρματα με το διακομιστή του συστήματος. Είναι υπεύθυνος για τη δημιουργία του ασυρμάτου δικτύου και για τη διασύνδεσή του με το εξωτερικό περιβάλλον, όπως το διαδίκτυο. Στον κόμβο πρόσβασης είναι προσαρτημένη ασύρματα και η κάμερα του συστήματος, η οποία προσφέρει την οπτική επιτήρηση του χώρου που το σύστημα καλύπτει. Τέλος, τη διαχείριση του ενσύρματου δικτύου, αναλαμβάνει ο διακομιστής του συστήματος. Είναι υπεύθυνος για τη λειτουργία του λογισμικού του συστήματος που έρχεται σε άμεση επαφή με το χρήστη (σχήμα Α).

Το λογισμικό του συστήματος αποτελείται από τρία σκέλη. Το πρώτο σκέλος είναι η ιστοσελίδα του συστήματος με το οποίο ο χρήστης επικοινωνεί άμεσα. Μέσω αυτής, ο χρήστης εισάγει τις εντολές του στο σύστημα και λαμβάνει τα αποτελέσματα αυτών. Η επεξεργασία των εντολών γίνεται κυρίως στο δεύτερο σκέλος του λογισμικού του συστήματος από αρχεία του προτύπου CGI. Οι εντολές αφού επεξεργαστούν αποστέλλονται στο τρίτο σκέλος του λογισμικού του συστήματος, όπου εκτελούνται οι επεξεργασμένες πλέον εντολές. Τα αποτελέσματα τους επιστρέφουν στο δεύτερο σκέλος, όπου τα αρχεία CGI ενημερώνουν το πρώτο σκέλος του λογισμικού του συστήματος, δηλαδή την ιστοσελίδα. Έτσι, ο χρήστης αλληλεπιδρά με το σύστημα και άρα με τον χώρο κάλυψής του, απλά μέσω μιας ιστοσελίδας. Συνεπώς η λειτουργία του λογισμικού του συστήματος ακολουθεί μια διαδρομή η οποία επαναλαμβάνεται κάθε φορά που ο χρήστης θέλει να επικοινωνήσει με το σύστημα (σχήμα Β).



Σχήμα Β: Η διαδρομή του λογισμικού

ΚΕΦΑΛΑΙΟ 1^ο.

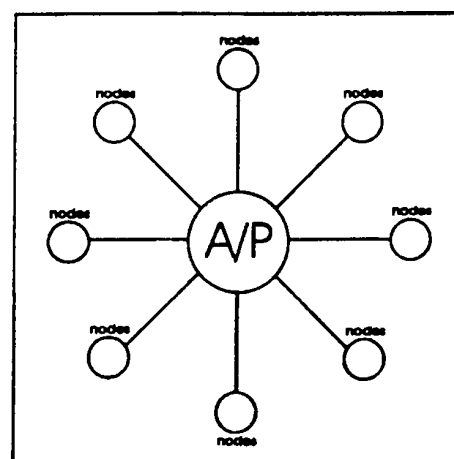
Εισαγωγή στις ασύρματες επικοινωνίες Wi-Fi και οι εφαρμογές τους

Ένα από τα πιο σημαντικά υψηλής συχνότητας ασύρματα δίκτυα, είναι το Wi – Fi. Η ονομασία αυτή είναι αδειοδοτημένη για τον μη κερδοσκοπικό οργανισμό Wi – Fi Alliance και περιγράφει την τεχνολογία της Ασύρματης Τοπικής Δικτύωσης (WLAN)¹. Η ονομασία προέρχεται από τα αρχικά **Wireless Fidelity** (ασύρματη πιστότητα). Το είδος της δικτύωσης αυτής βασίζεται στις προδιαγραφές του πρωτοκόλλου IEEE² 802.11, του πρώτου πρωτοτύπου που αναπτύχθηκε για ασύρματη δικτύωση [1,14,26].



Σχήμα 1: Το Λογότυπο του Wi-Fi

Το Wi – Fi αναπτύχθηκε αρχικά για να ικανοποιήσει την ανάγκη δημιουργίας ασύρματων τοπικών δικτύων κινητών ηλεκτρονικών συσκευών. Γρήγορα, όμως, λόγω της χρησιμότητας και της χρηστικότητάς του, επεκτάθηκε σε σωρεία ηλεκτρονικών συσκευών, όπως φορητών υπολογιστών, προσωπικών ψηφιακών βοηθών (Personal Digital Assistant), συσκευές παιχνιδιών, ηλεκτρονικών οικιακής ψυχαγωγίας (DVD players, ψηφιακές τηλεοράσεις, ηχοσυστήματα), ασύρματα τηλέφωνα, συστήματα τηλεμετρίας και ασφάλειας, ακόμη και σε αυτοκίνητα για χρήση σε εθνικές αρτηρίες.



Το Wi – Fi είναι ασύρματο δίκτυο τύπου αστέρα. Καλείται *Σύνολο Βασικών Υπηρεσιών*, Basic Service Set (BSS), και αποτελείται από ένα κεντρικό κόμβο και πολλούς σταθμούς [1,14]. Ο κεντρικός κόμβος ενός δικτύου Wi – Fi, είναι ο

Σχήμα 2: Δίκτυο Αστέρα

¹ Wireless local area network(WLAN)

² Institution of Electrical and Electronic Engineers

κόμβος πρόσβασης (Access Point) και τα όρια της περιοχής που καλύπτεται από το BSS, είναι η *Περιοχή Βασικών Υπηρεσιών*, Basic Service Area (BSA)³ [1,14]. Η BSA μπορεί να είναι ένα δωμάτιο, αλλά και μία περιοχή μέχρι αρκετά τετραγωνικά χιλιόμετρα με αλληλοκαλυπτόμενα σημεία πρόσβασης. Ο κόμβος πρόσβασης είναι η συσκευή που λειτουργεί ως η βάση του δικτύου και είναι υπεύθυνη για την διευθυνσιοδότηση των συσκευών που είναι συνδεδεμένες σε αυτήν. Ακόμη είναι υπεύθυνη για την διασύνδεση του δικτύου με άλλα όμοια ή με ενσύρματα δίκτυα, όπως τοπικά ή το Διαδίκτυο. Το σύστημα που ενώνει τους κόμβους πρόσβασης του ίδιου δικτύου μεταξύ τους και που τους επιτρέπει την ανταλλαγή πακέτων, καλείται Σύστημα Διανομής (Distribution System).

Ένα ασύρματο δίκτυο Wi – Fi περιέχει έναν ή περισσότερους κόμβους πρόσβασης, καθώς και έναν ή περισσότερους σταθμούς (clients). Ο κόμβος πρόσβασης είναι υπεύθυνος για την εκπομπή της ταυτότητας του δικτύου, Service Set Identifier (SSID) [1,14]. Η εκπομπή της ταυτότητας αυτής γίνεται σε πακέτα που ονομάζονται δέσμες ραδιοκυμάτων (beacons) και εκπέμπονται κάθε 100ms με ταχύτητα 1Mbps που είναι και η ελάχιστη δυνατή σε ένα δίκτυο Wi – Fi [14,26,33]. Επίσης, σημαντικό ρόλο στη λειτουργία του δικτύου έχει και το λογισμικό του υλικού (firmware) της κάθε συσκευής που συνδέεται σε αυτό. Για παράδειγμα, έστω ότι ένα δίκτυο απαρτίζεται από δύο κόμβους πρόσβασης του ίδιου SSID. Μια συσκευή που θα προσπαθήσει να συνδεθεί, θα προτιμήσει τη σύνδεση με τον κόμβο πρόσβασης του οποίου το σήμα είναι ισχυρότερο. Αυτή η επιλογή, δίνει άλλωστε στο χρήστη τη δυνατότητα φορητότητας μέσα στη BSA, αλλάζοντας κόμβους πρόσβασης ανάλογα με την ισχύ του σήματος.

Το πρωτόκολλο του δικτύου Wi – Fi 802.11 χωρίζεται σε δυο βασικά πρότυπα. Το 802.11a και το 802.11b [14,26]. Τα πιο κοινά ασύρματα δίκτυα Wi- Fi λειτουργούν στη μη αδειοδοτημένη περιοχή συχνοτήτων ISM (Industrial, Scientific and Medical) των 2.4GHz και στην UNII (Unlicensed National Information Infrastructure) των 5GHZ [14,26]. Εκτός όμως από τα

³ Όταν πρόκειται για BSA δημόσιας χρήσης, ονομάζονται hotspots

δύο βασικά πρότυπα, έχουν δημιουργηθεί κάποια υποπρότυπα, που αποτελούν εξελίξεις του πρωτοκόλλου. Αναλυτικότερα:

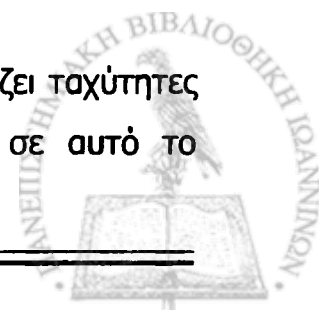
i. **IEEE 802.11a:** Χρησιμοποιεί τη ζώνη των 5 Ghz. Αυτή η περιοχή έχει εύρος 300 MHz και χωρίζεται σε δυο υποπεριοχές. Η κατώτερη επεκτείνεται από 5.15 έως 5.35 MHz και η ανώτερη από 5.725 έως 5.825 MHz. Η ταχύτητα μεταφοράς δεδομένων που μπορεί να επιτευχθεί είναι μικρότερη από 54Mbps με τυπική τιμή τα 25 Mbps. Το είδος της διαμόρφωσης που χρησιμοποιείται σε κάθε περίπτωση εξαρτάται από τον ρυθμό μετάδοσης των δεδομένων. Για ταχύτητες μεταφοράς δεδομένων από 6 έως 9 Mbps η διαμόρφωση που χρησιμοποιείται είναι η Binary Phase Shift Keying (BPSK). Για ταχύτητες μεταφοράς δεδομένων από 12 έως 18 Mbps η διαμόρφωση είναι η Quadrature Phase Shift Keying (QPSK). Για ταχύτητες μεταφοράς δεδομένων από 24 έως 36 Mbps, η διαμόρφωση είναι η 16 Quadrature Amplitude Modulation (16-QAM). Για ταχύτητες μεταφοράς δεδομένων από 48 έως 54 Mbps η διαμόρφωση είναι 64-QAM [14,26,33].

ii. **IEEE 802.11b:** Χρησιμοποιεί τη ζώνη των 2.412 έως 2.484 GHz. Η ταχύτητα μεταφοράς δεδομένων που μπορεί να επιτευχθεί είναι μικρότερη από 11 Mbps με τυπική τιμή τα 6.5 MHz. Κάθε κανάλι έχει εύρος 22 MHz και διάκενο 5 MHz μεταξύ των καναλιών. Η διαμόρφωση που χρησιμοποιείται είναι Direct Sequence Spread Spectrum (DSSS) με Complementary Code Keying (CCK). Η τυπική του εμβέλεια σε κλειστό χώρο, φτάνει τα 50 μέτρα με ταχύτητα 11 Mbps και τα 90 μέτρα με ταχύτητα 1 Mbps. Σε εξωτερικό χώρο με κεραιές υψηλής απολαβής το πρότυπο αυτό, μπορεί να χρησιμοποιηθεί σε συνδέσεις σημείου – σημείου με εμβέλεια που ξεπερνά τα 8 χιλιόμετρα [14,26,33].

iii. **IEEE 802.11e:** Παρέχει εγγυήσεις για ποιότητα υπηρεσίας (Quality of Service - QoS) [14,26,33].

iv. **IEEE 802.11f:** Παρέχει φορητότητα των σταθμών μέσα σε ένα δίκτυο IP (Intra - network Handover) [14,26,33].

v. **IEEE 802.11g:** Επεκτείνει το 802.11b ώστε να προσεγγίζει ταχύτητες που αγγίζουν τα 54Mbps με τυπική τιμή τα 25 MHz. Και σε αυτό το

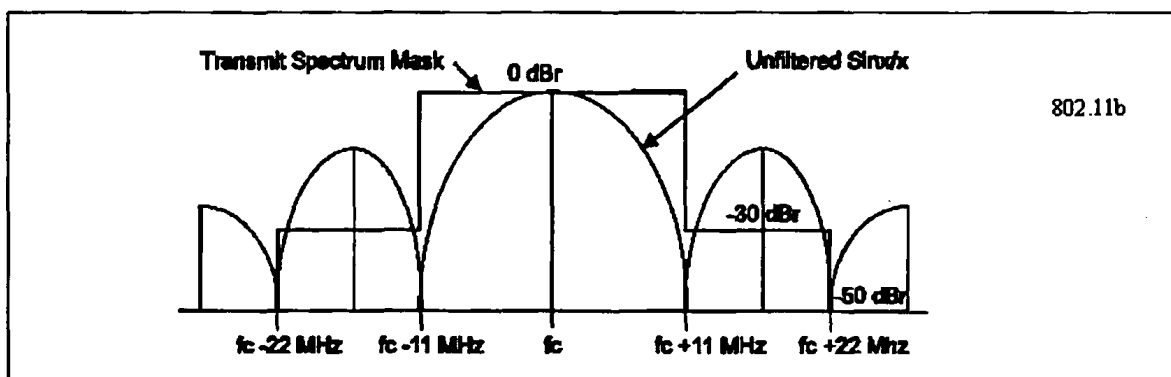


υποπρότυπο, κάθε κανάλι έχει εύρος 22 MHz και διάκενο 5 MHz μεταξύ των καναλιών. Η διαμόρφωση που χρησιμοποιείται για ταχύτητες 6, 9, 12, 18, 24, 36, 48 και 54 Mbps είναι η Orthogonal Frequency Division Multiplexing (OFDM). Για ταχύτητες 5.5 και 11 Mbps χρησιμοποιείται διαμόρφωση Complementary Code Keying (CCK) και για ταχύτητες 1 και 2 Mbps, χρησιμοποιείται διαμόρφωση DBPSK/DQPSK+DSSS⁴. Πρέπει να σημειωθεί ότι η μέγιστη ταχύτητα του 802.11g είναι αρκετά μεγαλύτερη από το 802.11b, αλλά η εμβέλεια στην οποία ο σταθμός μπορεί να πετύχει τη μέγιστη ταχύτητα είναι κατά πολύ μικρότερη. Η εμβέλεια στην ταχύτητα των 25 Mbps είναι περίπου 30 μέτρα [14,26,33].

vi. **IEEE 802.11i:** Πρότυπο το οποίο ικανοποιεί ή παρέχει θέματα ασφάλειας στα WLANs [14,26,33].

vii. **IEEE 802.11h:** Η συγκεκριμένη ομάδα κανόνων, εισάγει στο 802.11a την δυνατότητα καλύτερου ελέγχου των συγκρούσεων των πακέτων δεδομένων [14,26,33].

Τα κανάλια που χρησιμοποιούνται ποικίλλουν ανάλογα με το πρότυπο και με την γεωγραφική περιοχή. Τα πρότυπα 802.11b και 802.11g διαιρούν το φάσμα των 83.5 MHz (2.412 – 2.484 GHz) σε 14 αλληλοκαλυπτόμενα και εναλλασσόμενα κανάλια, με απόσταση μεταξύ των κεντρικών συχνοτήτων 5



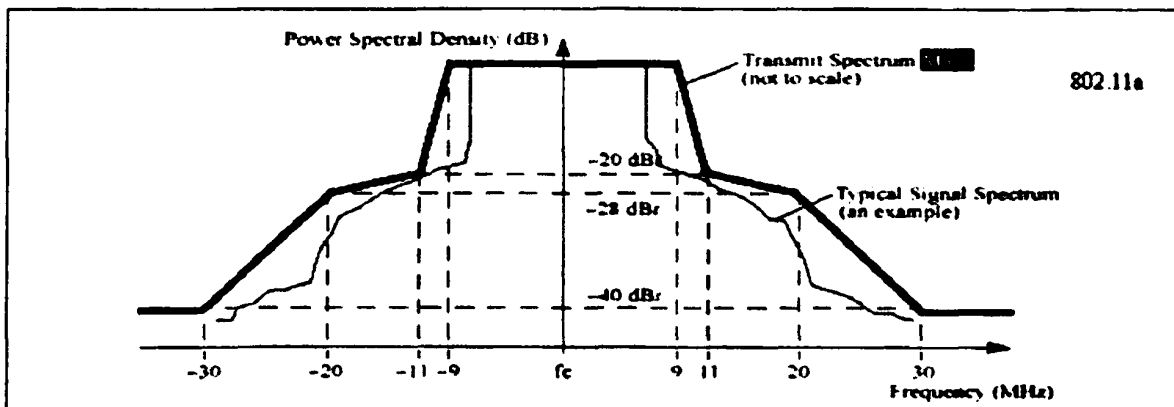
Σχήμα 3: Φάσμα προτύπου 802.11b/g

MHz. Το εύρος του κάθε καναλιού δεν είναι προκαθορισμένο, αλλά είναι προκαθορισμένη η κεντρική του συχνότητα. Όπως φαίνεται και στο σχήμα 3, το φασματικό εύρος για εξασθένιση τουλάχιστον 30dB, είναι ± 11 MHz από τη

⁴ Differentially BPSK, QPSK

κεντρική συχνότητα. Για εξασθένιση τουλάχιστον 50 dB, το φασματικό εύρος είναι ± 22 MHz από τη κεντρική συχνότητα. Στην Ευρωπαϊκή Ζώνη, ο μέγιστος αριθμός καναλιών που έχουν επιτραπεί είναι 13, στις Η.Π.Α. είναι 11^5 ενώ στην Ιαπωνία είναι 14 [14,26].

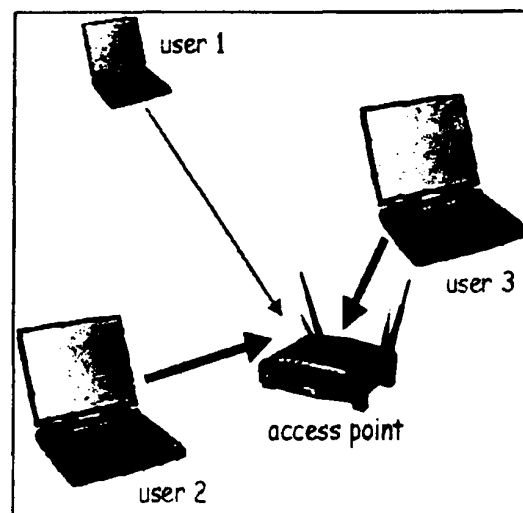
Το πρότυπο 802.11a, έχει 12 μη αλληλοκαλυπτόμενα κανάλια. Το φάσμα που μοιράζονται τα κανάλια, είναι 300 MHz. Από αυτά, τα 8 είναι αφιερωμένα



Σχήμα 4: Φάσμα προτύπου 802.11a

για εσωτερική χρήση, ενώ τα υπόλοιπα 4 για συνδέσεις τύπου κόμβου – κόμβου. Όπως φαίνεται και στο σχήμα 4, για εξασθένιση τουλάχιστον 20dB, το εύρος φάσματος του καναλιού, είναι ± 9 MHz. Για εξασθένιση τουλάχιστον 28 dB, το εύρος φάσματος του καναλιού είναι ± 20 MHz, ενώ για εξασθένιση τουλάχιστον 40 dB το εύρος είναι ± 30 MHz [14,26,33].

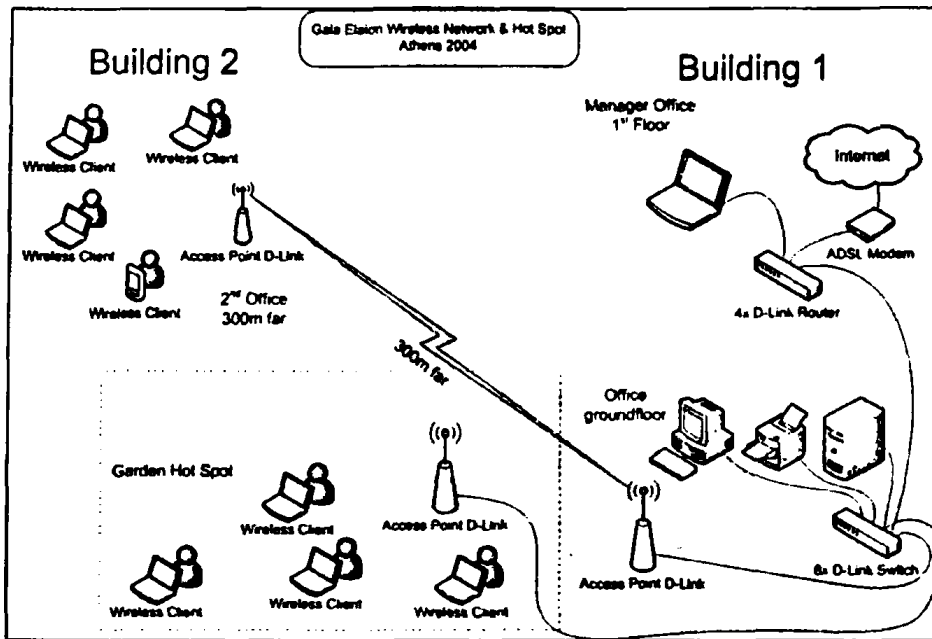
Ο τρόπος με τον οποίο μπορεί να χρησιμοποιηθεί το ασύρματο δίκτυο Wi – Fi είναι είτε ως αυτόνομο τοπικού δικτύου με ένα ή περισσότερους κόμβους πρόσβασης και έναν ή περισσότερους σταθμούς, είτε ως τμήμα ενός ευρύτερου δικτύου που αποτελείται από ενσύρματα αλλά και ασύρματα υποδίκτυα.



Σχήμα 5: Δίκτυο με έναν κόμβο πρόσβασης

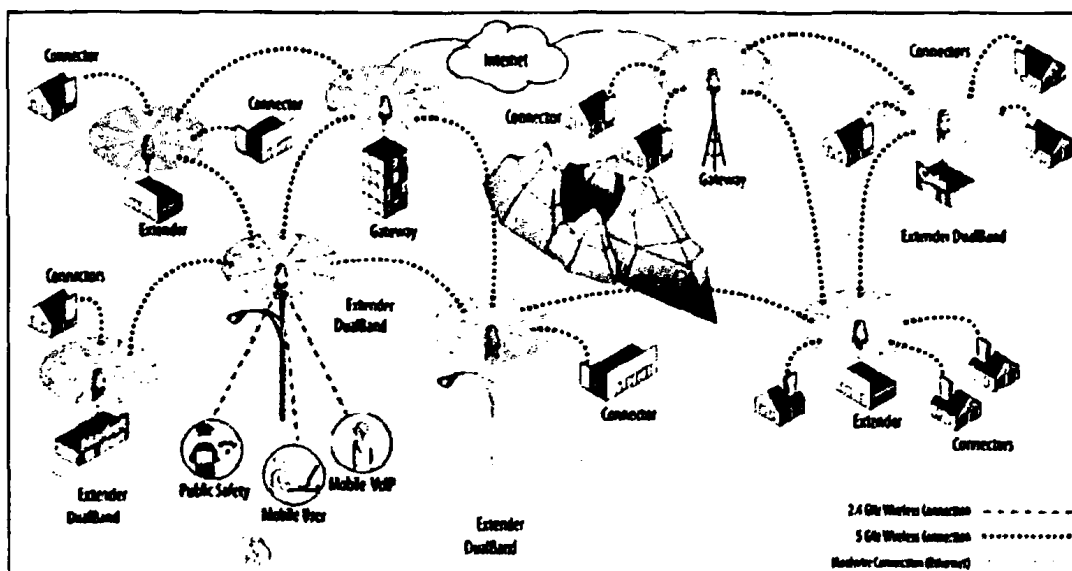
⁵ Βάσει των κανονισμών του οργανισμού FCC

Όπως φαίνεται στο σχήμα 5, η απλούστερη μορφή ενός δικτύου Wi - Fi περιέχει έναν κόμβο πρόσβασης και κάποιον αριθμό σταθμών /χρηστών. Οι



Σχήμα 6: Σύνθετο δίκτυο

σταθμοί μπορεί να είναι φορητοί υπολογιστές, αλλά μπορεί επίσης να είναι και οποιαδήποτε ηλεκτρονική συσκευή που έχει τη δυνατότητα σύνδεσης στο ασύρματο δίκτυο (π.χ. PDA, VoIP τηλέφωνα, κινητά τηλέφωνα). Ο κόμβος πρόσβασης, είναι το αυτός που ρυθμίζει την δημιουργία και λειτουργία του



Σχήμα 7: Ευρύτερο σύνθετο δίκτυο



δικτύου. Ανάλογα με το πρότυπο που χρησιμοποιείται ποικίλλει και η ταχύτητα του δικτύου.

Το ασύρματο δίκτυο Wi – Fi μπορεί να χρησιμοποιηθεί και ως τμήμα ενός ευρύτερου δικτύου. Όπως φαίνεται και στο σχήμα 6, μπορεί να επιτευχθεί κάλυψη ενός μεγάλου χώρου, με συνδυασμό ενσύρματων και ασύρματων δικτύων. Τα ασύρματα δίκτυα Wi – Fi, λόγω του φυσικού τους μέσου και της συχνότητας που χρησιμοποιούν, έχουν κάποιους περιορισμούς στην εμβέλειά τους. Έτσι, μια μεγάλη έκταση όπως η παραπάνω, καλύπτεται κατά τμήματα τα οποία συνδέονται μεταξύ τους ενσύρματα ή ασύρματα.

Ακόμη, με συνδυασμό ενσύρματων και ασύρματων δικτύων, είναι εφικτό να καλυφθούν τεράστιες εκτάσεις όπως μια πόλη. Στο σχήμα 7, παρουσιάζεται η πραγματοποίηση μια τέτοιας κάλυψης. Σε μια τέτοια περίπτωση, οι χρήστες μπορεί να είναι είτε απλοί πολίτες, είτε κρατικοί φορείς, είτε επαγγελματίες. Τέτοια παραδείγματα μπορούμε να βρούμε πλέον σε αρκετές ανεπτυγμένες πόλεις.

Το ασύρματο δίκτυο είναι αρκετά χρήσιμο αλλά εκτός από προτερήματα έχει και μειονεκτήματα. Παρακάτω αναφέρονται τα πιο σημαντικά:

- **Πλεονεκτήματα**

1. Ευκολία στη δημιουργία του δικτύου, αφού δεν απαιτείται καμία προεργασία εγκατάστασης (καλώδια κλπ). Αυτό δίνει τη δυνατότητα δικτύωσης χώρων με χωροταξικές δυσκολίες όπως υπαίθριοι χώροι, ιστορικά μνημεία κ.α.
2. Σημαντική μείωση του κόστους δημιουργίας, καθώς δεν χρειάζονται καλώδια και οι συσκευές ασύρματης δικτύωσης, γίνονται συνεχώς φθηνότερες.
3. Δυνατότητα φορητότητας. Ένας σταθμός /χρήστης μπορεί να μετακινείται στο χώρο κάλυψης και να συνδέεται κάθε φορά με τον κόμβο πρόσβασης του δικτύου, που του παρέχει το ισχυρότερο σήμα. Επίσης είναι δυνατή και η σύνδεση με άλλο δίκτυο, αφού προηγουμένως έχουν γίνει οι απαραίτητες ενέργειες για τη παραχώρηση αυτού του δικαιώματος από τον εξυπηρετητή (server) του δικτύου [14,26,33].



- **Μειονεκτήματα**

1. Καθώς το φυσικό μέσο του δικτύου είναι ο αέρας, είναι εύκολο να παρεμβληθούν στις συχνότητες του δικτύου συσκευές ή συστήματα που λειτουργούν στις ίδιες συχνότητες, όπως ασύρματα τηλέφωνα.
2. Αν και χρησιμοποιούνται τεχνικές κωδικοποίησης και κρυπτογράφησης, η υποκλοπή σε ένα ασύρματο δίκτυο θεωρείται ευκολότερη και εφικτότερη απ' ό τι σε ένα ενσύρματο.
3. Η εμβέλεια ενός ασύρματου δικτύου εξαρτάται εκτός από τη μορφολογία του χώρου κάλυψης και από την ισχύ εκπομπής του κόμβου πρόσβασης. Καθώς όμως η εκπομπή ακτινοβολίας και μάλιστα στις συχνότητες λειτουργίας του Wi – Fi είναι επιβλαβής για την ανθρώπινη υγεία, η ισχύς ακτινοβολίας υπόκειται σε περιορισμούς. Οι περιορισμοί καθορίζονται από κρατικούς οργανισμούς ,συνήθως, και για την Ευρωπαϊκή ζώνη το όριο εκπομπής είναι περιορισμένο στα 20 dBm.
4. Αρκετές φορές λόγω ταυτόχρονης εκπομπής περισσότερων σημείων πρόσβασης στον ίδιο χώρο κάλυψης, παρατηρούνται συγκρούσεις μεταξύ των καναλιών τους. Αυτό είναι κάτι που δεν μπορεί να αποφευχθεί εύκολα, παρά μόνο με έλεγχο των περιοχών κάλυψης των κόμβων πρόσβασης [14,26].

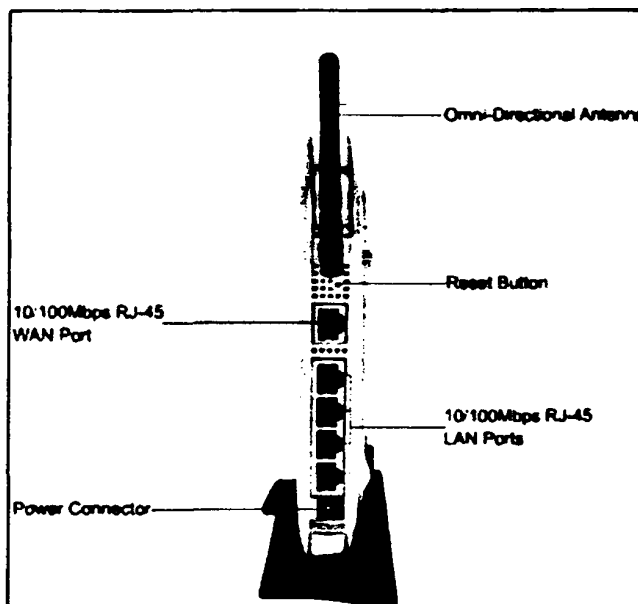
ΚΕΦΑΛΑΙΟ 2^ο.

Το ασύρματο σύστημα αυτοματισμού και τηλεμετρίας ερευνητικού εργαστηρίου

2.1 Ο κόμβος πρόσβασης (Access Point).

Το κύριο μέρος του ασύρματου δικτύου της παρούσας διπλωματικής, είναι ο κόμβος πρόσβασης (A/P). Αποτελεί τη γέφυρα μεταξύ ενσύρματου και ασύρματου κόσμου. Το A/P που χρησιμοποιείται στη συγκεκριμένη εφαρμογή, είναι το WGR614 της εταιρίας Netgear. Μπορεί να λειτουργήσει στα πρότυπα 802.11b και 802.11g, δηλαδή μπορεί να λειτουργήσει σε ταχύτητες μεταφοράς δεδομένων από 1 έως 54 Mbps.

Το A/P της εφαρμογής, (σχ 8) διαθέτει πολυκατευθυντική εύκαμπτη κεραία, κουμπί επανεκκίνησης, θύρες δικτύου 10/100 Mbps τύπου RJ-45 και τροφοδοτείται εξωτερικά. Εκτός από κόμβος πρόσβασης, μπορεί να λειτουργήσει και ως διακλαδωτής δικτύου 4 θυρών, καθιστώντας έτσι δυνατή τη δημιουργία και υποστήριξη ενσύρματου αλλά και ασύρματου δικτύου. Η συσκευή μπορεί να συνδεθεί ενσύρματα με άλλα δίκτυα 10/100 Mbps, μέσω των θυρών WAN [19].



Σχήμα 8: Ο κόμβος πρόσβασης του συστήματος

Ο έλεγχος του κόμβου πρόσβασης, γίνεται μέσω ενός εξυπηρετητή ιστού που λειτουργεί στον κόμβο πρόσβασης. Έτσι η πρόσβαση σε αυτόν, γίνεται από οποιονδήποτε ηλεκτρονικού υπολογιστή (H/Y) ή PDA που είναι συνδεδεμένος στο σύστημα, ενσύρματα ή ασύρματα.

Μέσα από τη σύνδεση αυτή, είναι δυνατό να ελεγχθούν όλες οι παράμετροι του κόμβου πρόσβασης. Οι πιο σημαντικές αναλύονται ακολούθως:

Σχήμα 9: Το πλαίσιο ελέγχου πρόσβασης

2.1.1 Η ασφάλεια του συστήματος

Η πρόσβαση στο σύστημα είναι εφικτή μόνο με χρήση ονόματος χρήστη και κωδικού πρόσβασης (σχήμα 9). Εφόσον εισαχθούν σωστά στον έλεγχο

Σχήμα 10: Οι βασικές ρυθμίσεις του κόμβου πρόσβασης.

κατά τη σύνδεση με την ιστοσελίδα ελέγχου του κόμβου πρόσβασης, ο χρήστης μπορεί να περιηγηθεί στα περιεχόμενα.

2.1.2 Οι βασικές ρυθμίσεις

Στο πεδίο αυτό, δηλώνονται οι ρυθμίσεις για την σύνδεση του κόμβου με την παροχή διαδικτύου (σχήμα 10). Δηλώνονται δηλαδή όλες οι απαραίτητες διευθύνσεις δικτύου IP που χρειάζονται για να επιτευχθεί η σύνδεση του κόμβου πρόσβασης με το διαδίκτυο.

2.1.3 Οι ρυθμίσεις ασύρματου δικτύου

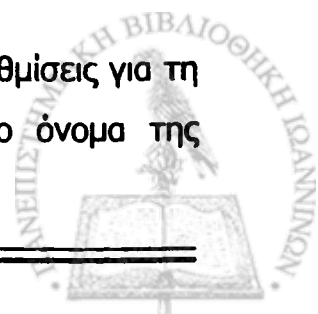
The screenshot shows the 'Wireless Settings' configuration page. On the left is a navigation menu with categories: Setup Wizard, Setup (Basic Settings, Wireless Settings, Content Filtering, Logs, Block Sites, Block Services, Schedule, E-mail), Maintenance (Router Status), Attached Devices, Backup Settings, Set Password, Router Upgrade, Advanced (Wireless Settings, Port Forwarding / Port Triggering, WAN Setup, LAN IP Setup, Dynamic DNS, Static Routes, Remote Management, UPnP), and Web Support (Knowledge Base, Documentation, Logout). The main area is titled 'Wireless Settings' and contains the following fields:

- Wireless Network:** Name (SSID) w_T20j, Region Europe, Channel 11, Mode g and b.
- Security Options:** Radio buttons for Disable, WEP (Wired Equivalent Privacy), and WPA-PSK (W-Fi Protected Access Pre-Shared Key).
- Security Encryption (WEP):** Authentication Type Automatic, Encryption Strength 128bit.
- Security Encryption (WEP) Key:** Passphrase field with a 'Generate' button, and four key input fields (Key 1: 10203040506070809010203040, Key 2, 3, 4).

At the bottom of the configuration area are 'Apply' and 'Cancel' buttons.

Σχήμα 11: Οι ρυθμίσεις του ασυρμάτου δικτύου

Στο πεδίο αυτό, ο χρήστης δηλώνει όλες τις απαραίτητες ρυθμίσεις για τη λειτουργία του ασύρματου δικτύου (σχήμα 11). Δηλώνει το όνομα της



ταυτότητας του δικτύου SSID και την περιοχή λειτουργίας του κόμβου πρόσβασης, ώστε να ρυθμιστούν πόσα κανάλια θα είναι σε λειτουργία. Στη συγκεκριμένη περίπτωση, ο κόμβος πρόσβασης είναι ρυθμισμένος να λειτουργεί στην ευρωπαϊκή ζώνη. Επίσης, στο πεδίο αυτό, δηλώνονται και οι ρυθμίσεις ασφαλείας του ασύρματου δικτύου. Στη περίπτωση της παρούσας διπλωματικής, χρησιμοποιείται κωδικοποίηση δεδομένων WEP⁶ 128-bit με χρήση κλειδιού.

2.1.4 Οι ρυθμίσεις τοπικού δικτύου

Στο πεδίο αυτό, ο χρήστης μπορεί να ρυθμίσει τις παραμέτρους του

Setup Wizard

- Setup
- Basic Settings
- Wireless Settings
- Content Filtering
- Logs
- Block Sites
- Block Services
- Schedule
- E-mail
- Maintenance
- Router Status
- Attached Devices
- Backup Settings
- Set Password
- Router Upgrade
- Advanced
- Wireless Settings
- Port Forwarding / Port Triggering
- WAN Setup
- LAN IP Setup
- Dynamic DNS
- Static Routes
- Remote Management
- UPnP
- Web Support
- Knowledge Base
- Documentation
- Logout

LAN IP Setup

LAN TCP/IP Setup

IP Address: 192 . 168 . 1 . 1

IP Subnet Mask: 255 . 255 . 255 . 0

DHCP Direction: None

DHCP Version: Disabled

Use Router as DHCP Server

Starting IP Address: 192 . 168 . 1 . 21

Ending IP Address: 192 . 168 . 1 . 101

Address Reservation

#	IP Address	Device Name	Mac Address

Add Edit Delete

Apply Cancel

Σχήμα 12: Οι ρυθμίσεις τοπικού δικτύου

⁶ Wireless Equivalent Privacy

τοπικού δικτύου του κόμβου πρόσβασης (σχήμα 12). Όπως τη διεύθυνση δικτύου IP του κόμβου πρόσβασης και τις διευθύνσεις δικτύου IP στις οποίες ο κόμβος πρόσβασης θα λειτουργεί ως DHCP⁷ server.

2.1.5 Οι ρυθμίσεις προώθησης/ενεργοποίησης θυρών

Στο πεδίο αυτό, ο χρήστης μπορεί να ρυθμίσει ποια θύρα επικοινωνίας θα χρησιμοποιεί η κάθε υπηρεσία (σχήμα 13). Είναι μια αρκετά σημαντική ρύθμιση για τη λειτουργία του συστήματος τις παρούσας διπλωματικής, καθώς αποφεύγονται συγκρούσεις πακέτων δεδομένων μεταξύ των διαφόρων υπηρεσιών. Έτσι, δηλώνεται συγκεκριμένη θύρα για το πρωτόκολλο HTTP, για τις ασύρματους σταθμούς διασύνδεσης Wi - Fi, για τη κάμερα του συστήματος, αλλά και για την υπηρεσία Telnet.

Port Forwarding / Port Triggering

Please select the service type

Port Forwarding
 Port Triggering

Service Name	Server IP Address
A.M	192 168 1

Add

#	Service Name	Start Port	End Port	Server IP Address
<input type="radio"/> 1	gw21w	4660	4660	192.168.1.11
<input type="radio"/> 2	HTTP	80	80	192.168.1.14
<input type="radio"/> 3	Telnet	23	23	192.168.1.11
<input type="radio"/> 4	WCS2030	8080	8080	192.168.1.10

Edit Service **Delete Service**

Add Custom Service

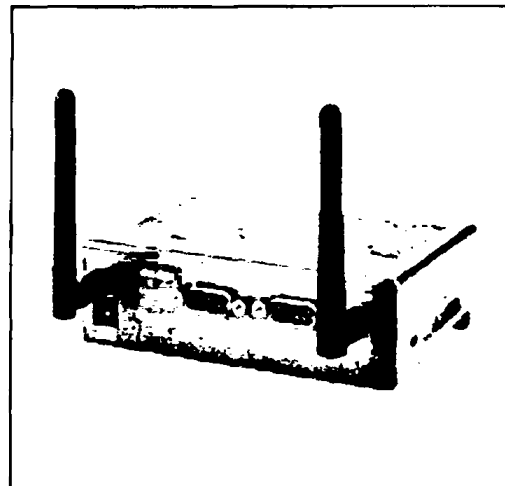
Σχήμα 13: Οι ρυθμίσεις προώθησης/ενεργοποίησης θυρών

⁷ Dynamic Host Configuration Protocol



2.2 Οι ασύρματοι σταθμοί διασύνδεσης (wireless modules)

Ο ασύρματος σταθμός διασύνδεσης Wi - Fi αποτελεί τον υποσταθμό (node) του ασυρμάτου δικτύου (σχήμα 14). Επίσης, είναι και η γέφυρα μεταξύ του ενσύρματου και του ασύρματου τμήματος του συστήματος. Ο ασύρματος σταθμός διασύνδεσης Wi - Fi είναι το μοντέλο GW21W-25 της εταιρίας ATOP.



Σχήμα 14: Ο ασύρματος σταθμός διασύνδεσης Wi - Fi

Ο σταθμός, αποτελείται από δύο τμήματα. Το τμήμα της ενσύρματης επικοινωνίας και το τμήμα της ασύρματης επικοινωνίας. Το τμήμα της ενσύρματης επικοινωνίας, διαθέτει δυο θύρες τύπου D-SUB9 οι οποίες λειτουργούν σε RS-232 ή/και RS-485. Το ασύρματο τμήμα λειτουργεί με το πρότυπο 802.11b και με ταχύτητα μεταφοράς δεδομένων έως 11Mbps. Διαθέτει δυο πολυκατευθυντικές εύκαμπτες κεραιές και μπορεί να χρησιμοποιηθεί σαν αναμεταδότης αυξάνοντας έτσι την περιοχή κάλυψης του κόμβου πρόσβασης.

Η καρδιά του σταθμού είναι ένας μικροεπεξεργαστής των 16-bit που λειτουργεί στα 40 MHz και διαθέτει τρεις μνήμες, flash μνήμη 512 KBytes, EPROM μνήμη 256 KBytes και EEPROM μνήμη 512 Bytes. Η συσκευή διαθέτει ακόμη, watchdog timer και επιτήρηση τάσης με κατώφλι στα 4.75V. Η τροφοδοσία του γίνεται στα 12VDC/350mA [17].

Ο έλεγχος του συστήματος γίνεται με τρεις τρόπους. Ο πρώτος είναι με χρήση Telnet ή μέσω κονσόλας επικοινωνίας RS-232 (π.χ. η εφαρμογή Hyperterminal των windows). Οι αλλαγές στις ρυθμίσεις γίνονται με περιγραφικό τρόπο, δηλαδή γράφοντας στο αντίστοιχο πεδίο την κάθε εντολή. Κάτι τέτοιο προσφέρει αμεσότητα στον έλεγχο της συσκευής διότι απλά χρειάζεται να συνδέσουμε το σταθμό με τη σειριακή θύρα ενός Η/Υ χωρίς την παρουσία κόμβου πρόσβασης. Όμως το περιβάλλον που

χρησιμοποιεί αυτός ο τρόπος ελέγχου των ρυθμίσεων, είναι μη πρακτικός και δύσχρηστος. Ο χρήστης πρέπει να διαθέτει τις βασικές γνώσεις χειρισμού Telnet ώστε να μπορέσει να χειριστεί το αντίστοιχο περιβάλλον. Στο σχήμα 15

```

Telnet - 10.0.50.100
Connect Edit Terminal Help
ABLELink Ethernet-Serial Server
User name:admin
Password:
Login ok

0.Exit 1.Overview 2.Networking 3.Security 4.Com1 5.Com2 6.WLAN

Input choice and enter(0~6): 1

Overview:
Model Name      : GW21W
IP Address      : 10.0.50.100
MAC Address     : 00:60:B3:76:FE:0B
SysName         : CS GW21W
SysLocation     : location
SysContact      : contact
Kernel Version  : 2.20
AP Version      : TerminalSrv ver3.07
Link Status     : SS

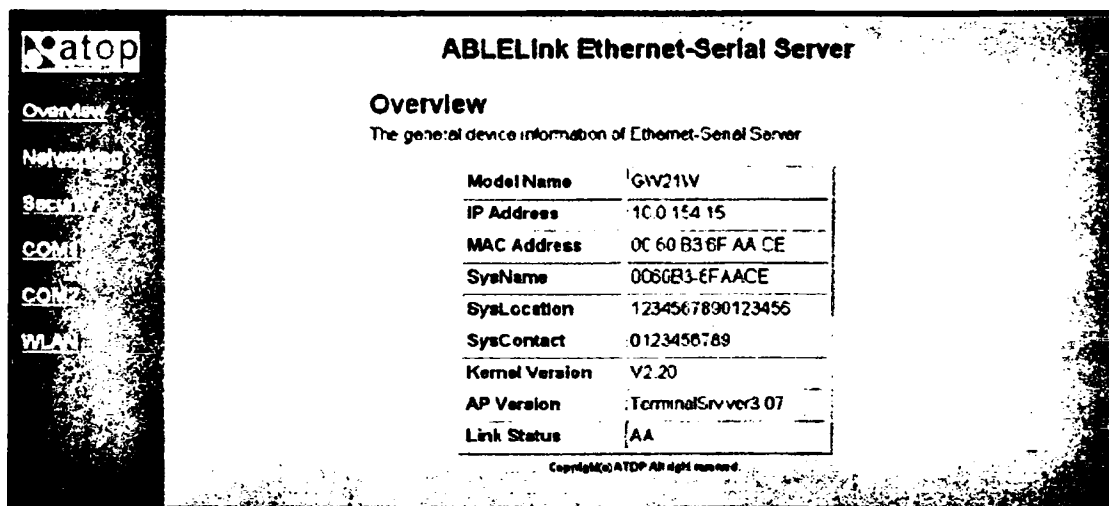
0.Exit 1.Overview 2.Networking 3.Security 4.Com1 5.Com2 6.WLAN

Input choice and enter(0~6): -
    
```

Σχήμα 15: Περιβάλλον DOS

παρουσιάζεται ένα παράδειγμα κονσόλας επικοινωνίας RS-232.

Ο δεύτερος τρόπος ελέγχου του συστήματος είναι μέσω ιστοσελίδας



Σχήμα 16: Παραθυρικό περιβάλλον HTML

υπερκειμένου (σχήμα 16). Ο web server, λειτουργεί στον σταθμό ασύρματης διασύνδεσης. Αυτό, δίνει τη δυνατότητα ελέγχου των ρυθμίσεων της συσκευής από οποιοδήποτε Η/Υ ή PDA που είναι συνδεδεμένος στο σύστημα.

Έτσι ο έλεγχος είναι ευκολότερος χωρίς να απαιτούνται ιδιαίτερες γνώσεις από

atop

Overview
Network
Security
COM1
COM2
WLAN

TCP/IP

To configure network settings of Ethernet-Serial Server. After saving configuration you have to restart the device to make the settings effective.

DHCP	<input type="checkbox"/> Obtain an IP automatically
IP Address	10.0.0.100
Default Gateway	10.0.0.254
Subnet Mask	255.255.0.0

SNMP

By enabling SNMP you allow the management utility to collect the information of Ethernet-Serial Server. You can change the device network identity as well by changing the system name, location and contact.

SNMP	<input checked="" type="checkbox"/> Enable
SysName	EWS1W
SysLocation	location
SysContact	contact

Save Configuration Restart

Σχήμα 17: Ρυθμίσεις ενσύρματου δικτύου

τον χρήστη, αλλά είναι απαραίτητη η ύπαρξη κόμβου πρόσβασης.

atop

Overview
Network
Security
COM1
COM2
WLAN

ABLELink Ethernet-Serial Server

Security

The default password is null, you can change the password by filling in the new password to New Password and Verified Password fields, be aware that password is case sensitive.

Old Password	
New Password	
Verified Password	

Save Configuration

Σχήμα 18: Ρυθμίσεις ασφάλειας

Και με τους δύο τρόπους, μπορούμε να ελέγξουμε όλες τις παραμέτρους του ασύρματου σταθμού διασύνδεσης Wi - Fi κατά τον ίδιο τρόπο, αλλά με χρήση διαφορετικών εφαρμογών. Οι παράμετροι αυτές, αναλύονται ακολούθως:

2.2.1 Ρυθμίσεις ενσύρματου δικτύου

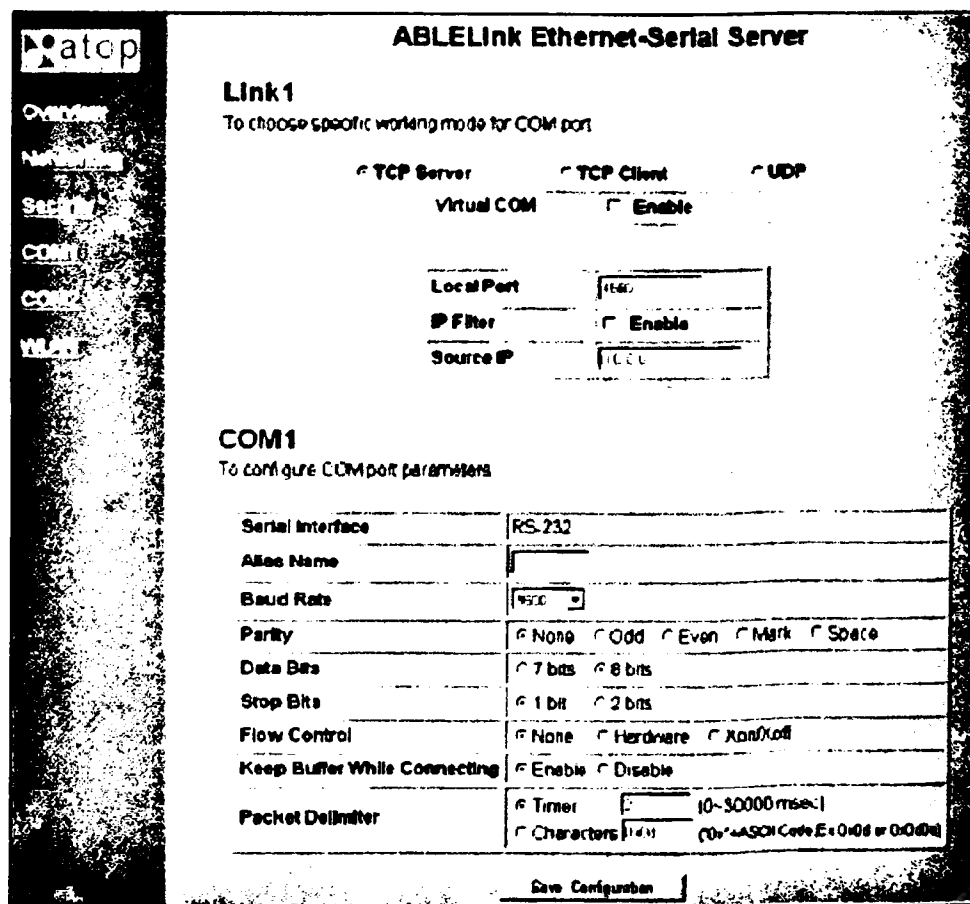
Όπως φαίνεται και στο σχήμα 17, στο πεδίο ελέγχου αυτό, μπορούμε να ρυθμίσουμε την ενθυλάκωση TCP/IP στο πακέτο δεδομένων,. Είτε ορίζουμε τα στοιχεία του πεδίου IP ή ζητούμε από τον σταθμό να ζητήσει και να ορίσει αυτόματα από τον κόμβο πρόσβασης τα στοιχεία αυτά (DHCP). Επιπλέον, μπορούμε να ενεργοποιήσουμε την εφαρμογή διαχείρισης δικτύου Simple Network Management Protocol (SNMP) και να ορίσουμε την ταυτότητα του σταθμού στο δίκτυο.

2.2.2 Ρυθμίσεις ασφάλειας

Στο πεδίο αυτό, σχήμα 18, ορίζουμε όνομα χρήστη και κωδικό πρόσβασης ώστε να διασφαλιστεί η πρόσβαση σε όλες τις ρυθμίσεις του ασύρματου σταθμού διασύνδεσης Wi - Fi.

2.2.3 Ρυθμίσεις σειριακής θύρας 1 και 2

Ο ασύρματος σταθμός διασύνδεσης Wi - Fi, διαθέτει δυο όμοιες θύρες

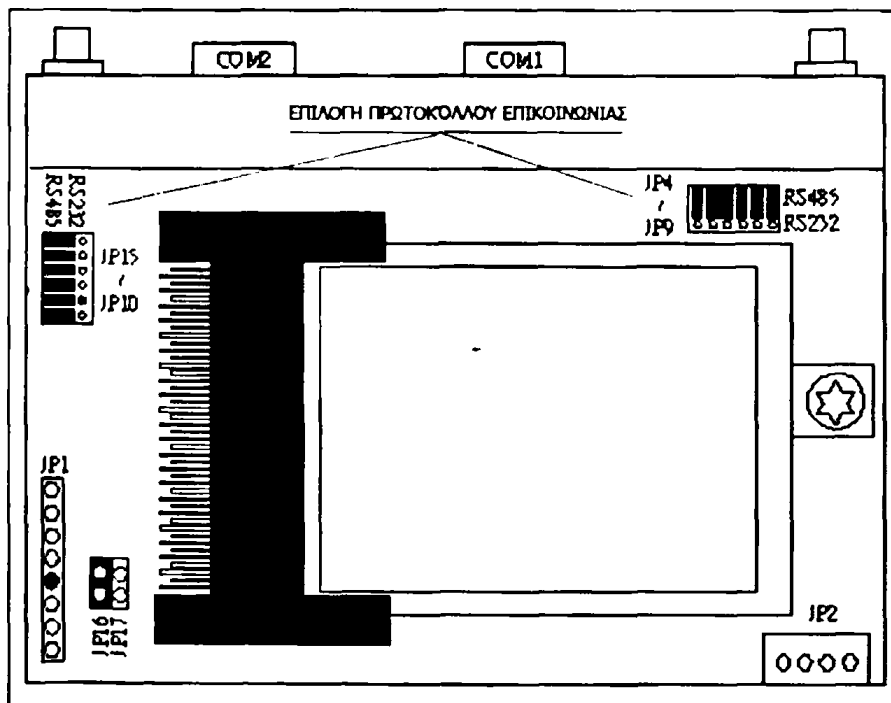


Σχήμα 19: Ρυθμίσεις σειριακής θύρας

σειριακής επικοινωνίας RS-232 ή/και RS-485. Οπότε, οι ρυθμίσεις των δυο



θυρών είναι επίσης όμοιες αλλά σε διαφορετικά πεδία. Η επιλογή RS-232 ή RS-485 για κάθε θύρα, γίνεται στο υλικό του σταθμού και όχι από το λογισμικό, αλλάζοντας τη θέση ενός jumper, όπως φαίνεται στο σχήμα 20. Στο σχήμα 19, παρουσιάζεται το πεδίο, από το οποίο μπορούμε να ελέγξουμε όλες τις παραμέτρους του πρωτοκόλλου. Αναφορικά, μπορούμε να ρυθμίσουμε την ταχύτητα πακέτων δεδομένων της θύρας, το bit ισοτιμίας (parity bit), τον αριθμό των bits τέλους, των αριθμό των bits του πακέτου δεδομένων, τον έλεγχο ροής δεδομένων, τον καταχωρητή (buffer) της επικοινωνίας καθώς και το συνολικό μέγεθος και διάρκεια των πακέτων. Επιπλέον, μπορούμε να καθορίσουμε την χρήση της εφαρμογής Virtual Com⁸ και τον τύπο ενθυλάκωσης (TCP server, TCP client ή UDP) και να καθορίσουμε αν θα χρησιμοποιήσουμε κάποιο φίλτρο IP.



Σχήμα 20: Επιλογή RS232 / RS485

2.2.4 Ρυθμίσεις ασυρμάτου δικτύου

Στο πεδίο αυτό πρωταρχικά, μπορούμε να επιλέξουμε την τοπολογία του δικτύου, δηλαδή αν θα είναι υποδομής (infrastructure) ή συγκεκριμένου σκοπού (Ad-hoc). Αν επιλεγεί δίκτυο Ad-hoc, τότε μπορούμε να επιλέξουμε και

⁸ Η εφαρμογή αυτή θέτει μια εικονική θύρα του Η/Υ σε μια διεύθυνση IP

το συγκεκριμένο κανάλι που θα χρησιμοποιηθεί. Στη συνέχεια, μπορούμε να επιλέξουμε την ταχύτητα μεταφοράς δεδομένων και την ταυτότητα του δικτύου ESSID στο οποίο θέλουμε να συνδεθεί ο σταθμός. Τέλος, μπορούμε να επιλέξουμε κωδικοποίηση δεδομένων WEP, ρυθμίζοντας το κατανεμημένο κλειδί.

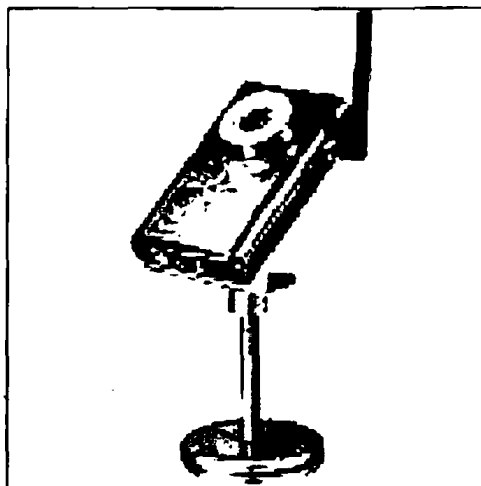
Ο ρόλος του ασύρματου σταθμού διασύνδεσης Wi - Fi, είναι να γεφυρώνει το ενσύρματο τμήμα του συστήματος με το ασύρματο. Για να επιτευχθεί αυτό, χρησιμοποιεί ενθυλάκωση TCP/IP και ταχύτητα μεταφοράς δεδομένων, αυτόματα προσαρμοσμένη στην ισχύ του σήματος του ασυρμάτου δικτύου με μέγιστη τα 11Mbps. Ο τύπος του ασυρμάτου δικτύου είναι infrastructure και για την ασφάλεια των δεδομένων, χρησιμοποιείται κωδικοποίηση WEP με κατανεμημένο κλειδί. Από της δύο σειριακές θύρες χρησιμοποιείται η μία σε RS-232 με ταχύτητα μετάδοσης πακέτων δεδομένων 9.6 Kbps και πακέτα των 8 bits, με 1 bit τερματισμού, χωρίς parity bit και έλεγχο ροής δεδομένων. Για την πραγματοποίηση της επικοινωνίας, χρησιμοποιείται η εφαρμογή Virtual com [30].



2.3 Η κάμερα του συστήματος (IP camera)

Ένα σημαντικό και καινοτόμο τμήμα του συστήματος, είναι η κάμερά του. Το κομμάτι αυτό του συστήματος, δίνει στο χρήστη τη δυνατότητα να έχει εκτός των άλλων και οπτική επιτήρηση του χώρου.

Η κάμερα (σχήμα 21) είναι τύπου IP, δηλαδή εκπέμπει σε συγκεκριμένη διεύθυνση δικτύου ενσύρματα ή ασύρματα. Ενσύρματα, μπορεί να συνδεθεί σε 10/100 baseT δίκτυο, ενώ ασύρματα λειτουργεί με το πρότυπο 802.11g. Τα πρωτόκολλα τα οποία υποστηρίζει είναι τα TCP/IP, HTTP, SMTP, FTP, DDNS, UPnP, Telnet, NTP, DNS και DHCP. Διαθέτει κεντρική μονάδα επεξεργασίας (CPU) Trimedia PNX1300 με μνήμες RAM 16Mb SDRAM και ROM 2Mb flash ROM.



Σχήμα 21: Η κάμερα του συστήματος

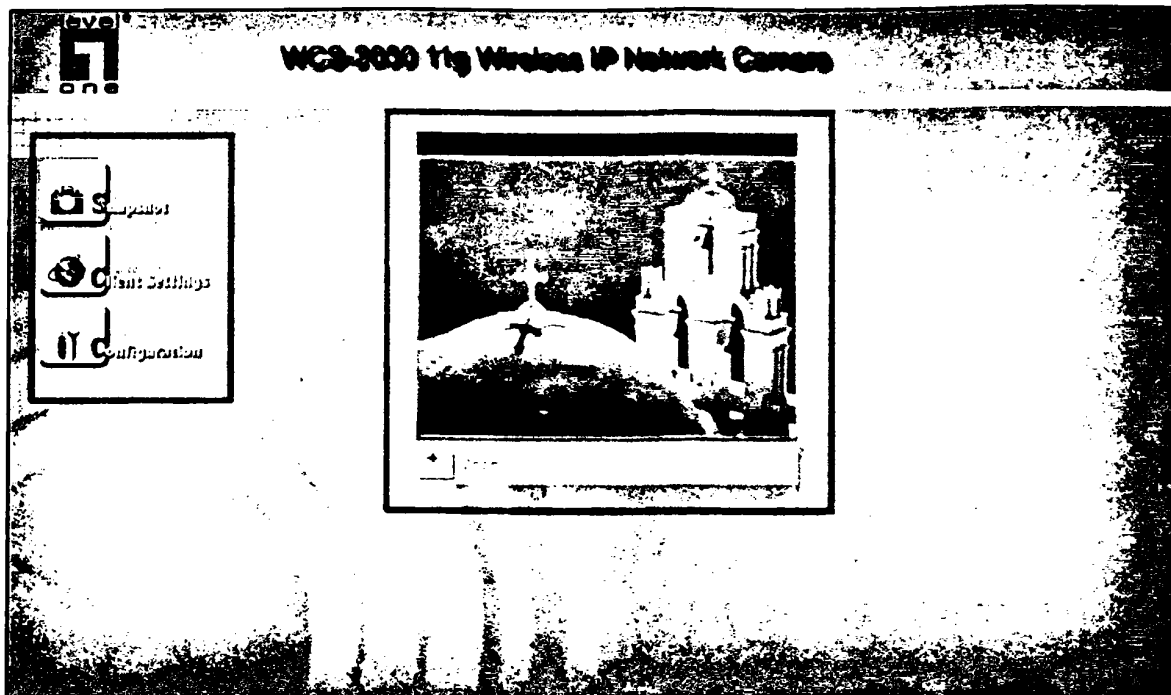
Ο φακός είναι τύπου fixed με σταθερή εστιακή απόσταση 4 χιλιοστά και διάφραγμα F2.0. Ο αισθητήρας της κάμερας είναι 1/4 της ίντσας και μέγιστης ανάλυσης 640x480. Υπάρχει ακόμη ηλεκτρονικό κλείστρο με χρόνο λειτουργίας από 1/60 έως 1/15000 δευτερόλεπτα. Ο αλγόριθμος συμπίεσης του βίντεο είναι MPEG4 και της εικόνας JPEG. Η ποιότητα εικόνας, το μέγεθος του κάδρου και η ταχύτητα μετάδοσης των πακέτων δεδομένων είναι προσαρμόσιμα. Επιπλέον, η κάμερα διαθέτει πολυκαντευθυντικό μικρόφωνο, με συχνότητα λειτουργίας από 50 έως 16000 Hz και σηματοθορυβικό λόγο πάνω από 60 dB. Ο ήχος ψηφιοποιείται με ρυθμό δειγματοληψίας στα 24 Kbps. Τέλος η τροφοδοσία της κάμερας είναι 5VDC, 2A [18].

Ο έλεγχος της κάμερας γίνεται μέσω web server, ο οποίος λειτουργεί στη συσκευή. Στο κεντρικό παράθυρο ελέγχου (σχήμα 22), υπάρχουν δύο κύρια πλαίσια. Αριστερά, ένα με συντομεύσεις για τις ρυθμίσεις της κάμερας και στο κέντρο ένα με συνεχή προβολή της εικόνας που έρχεται από την κάμερα.

Στο αριστερό πλαίσιο, υπάρχουν τρεις συντομεύσεις.

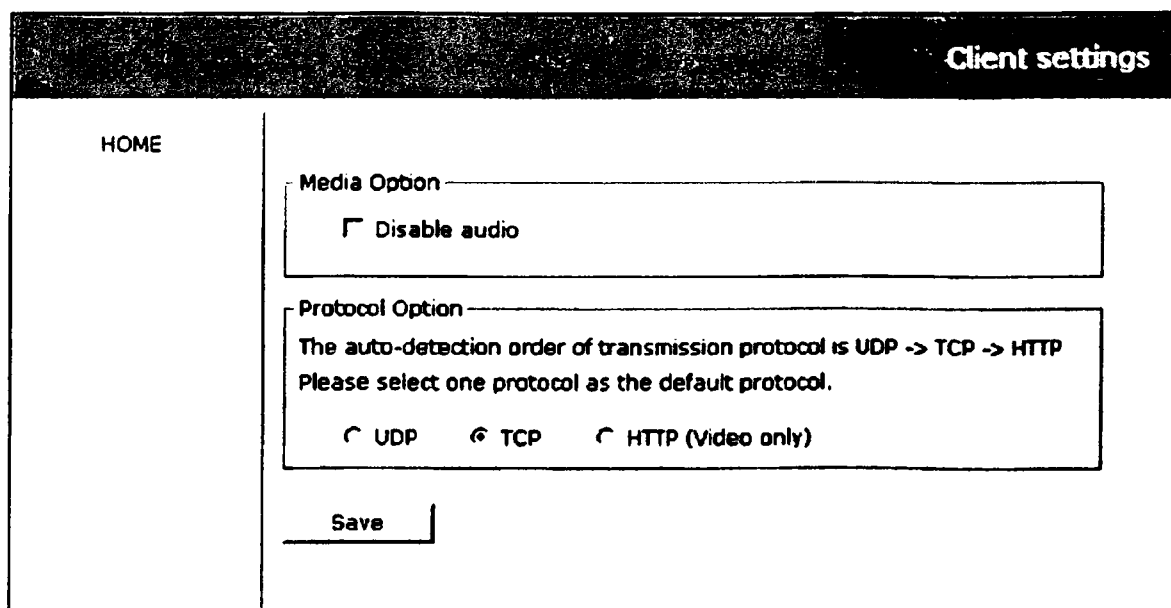
2.3.1 Snapshot.

Η συντόμευση αυτή επιτρέπει στο χρήστη να αποθηκεύσει το τρέχον



Σχήμα 22: Το κεντρικό παράθυρο ελέγχου

στιγμιότυπο (σχήμα 22). Μόλις επιλεχθεί το αντίστοιχο εικονίδιο, ανοίγει ένα καινούργιο παράθυρο με το αντίστοιχο στιγμιότυπο, το οποίο ο χρήστης



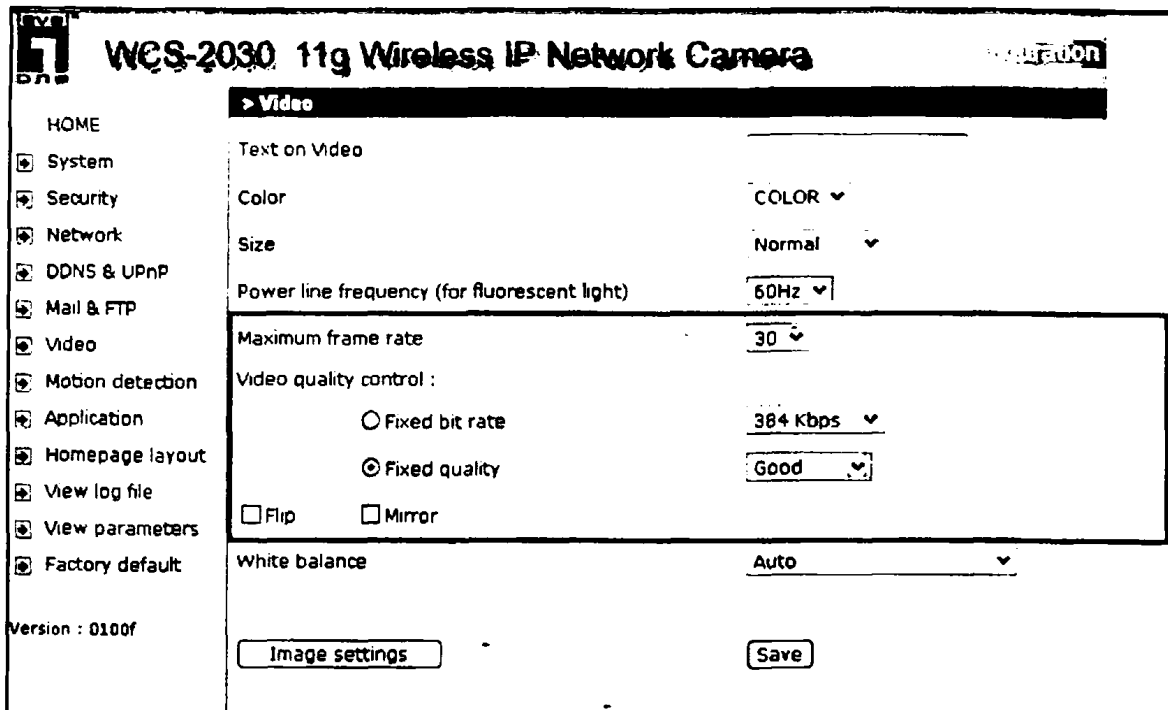
Σχήμα 23: Το παράθυρο Client Settings

μπορεί να αποθηκεύσει στον Η/Υ ή στο PDA από το οποίο έχει συνδεθεί με την κάμερα.



2.3.2 Client Settings.

Η συντόμευση αυτή περιέχει ρυθμίσεις για το πρωτόκολλο και για τον ήχο (σχήμα 23). Για το πρωτόκολλο ο χρήστης έχει τρεις επιλογές UDP, TCP και HTTP. Επιλέγοντας UDP, επιτυγχάνεται αδιάκοπη ροή δεδομένων, αλλά δεν ζητείται επανεκπομπή των πακέτων που δεν έφτασαν στον προορισμό τους. Αυτό προσφέρει αδιάκοπη ροή του βίντεο και του ήχου με κάποιες μικρές



Σχήμα 24: Το παράθυρο Configuration

αμελητέες διακοπές. Επιλέγοντας TCP, έχουμε επανεκπομπή των πακέτων που δεν έφτασαν στον προορισμό τους, αλλά αυτό καθυστερεί τη μετάδοση των δεδομένων, καθιστώντας πιο αργή τη ροή του βίντεο και του ήχου. Αν το δίκτυο στο οποίο είναι συνδεδεμένο η κάμερα προστατεύεται από κάποιο τοίχος προστασίας (firewall) και μόνο η θύρα 80 είναι ανοικτή, τότε πρέπει να επιλεγεί το πρωτόκολλο HTTP. Στην περίπτωση αυτή όμως, δεν θα γίνεται αποστολή του ήχου. Η επιλογή που συνιστάται είναι UDP μετά TCP και τέλος HTTP.

2.3.3 Configuration

Η συντόμευση αυτή (σχήμα 24), περιέχει ρυθμίσεις για όλες τις υπόλοιπες λειτουργίες του συστήματος. Ρυθμίσεις για το ασύρματο δίκτυο (SSID, πρωτόκολλα κλπ), ρυθμίσεις ασφάλειας, ρυθμίσεις για την ανάλυση και τα

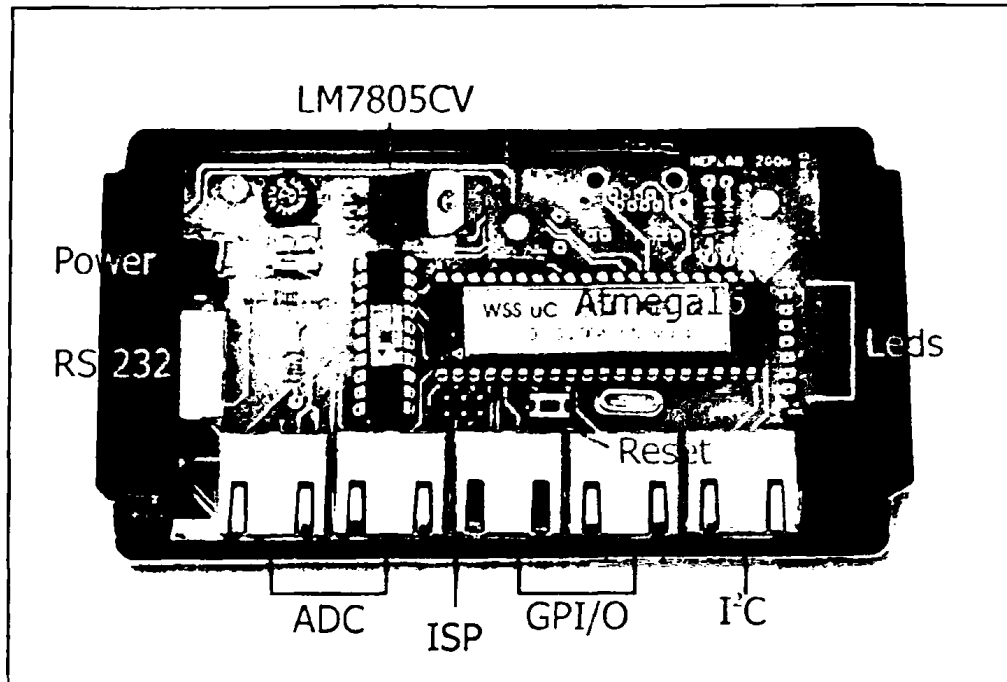
πλαίσια/δευτερόλεπτο του βίντεο. Επιπλέον στο παράθυρο αυτό, ο χρήστης μπορεί να ρυθμίσει την κάμερα, ώστε να του αποστέλλει στιγμιότυπα, εφόσον έχει διαπιστωθεί αυτόματα από την κάμερα αλλαγή σε συγκεκριμένο τμήμα της εικόνας. Δηλαδή όταν ανιχνευθεί κάποια κίνηση. Τα στιγμιότυπα αυτά, αποστέλλονται με ηλεκτρονικό ταχυδρομείο σε συγκεκριμένη προκαθορισμένη διεύθυνση.

Στο σύστημα της παρούσας διπλωματικής, η κάμερα, λειτουργεί με συγκεκριμένες ρυθμίσεις. Το πρωτόκολλο που χρησιμοποιείται είναι HTTP. Το βίντεο είναι ανάλυσης 640x480 και ο αριθμός των πλαισίων ανά δευτερόλεπτο, είναι ανάλογος των χρηστών που είναι συνδεδεμένοι στο σύστημα ταυτόχρονα. Όσοι περισσότεροι χρήστες είναι συνδεδεμένοι στο σύστημα, τόσο λιγότερα είναι τα πλαίσια ανά δευτερόλεπτο ώστε να μπορέσουν να εξυπηρετηθούν όλοι. Ο τυπικός ρυθμός είναι 30 πλαίσια/sec.



2.4 Ο σταθμός προσάρτησης αισθητηρίων.

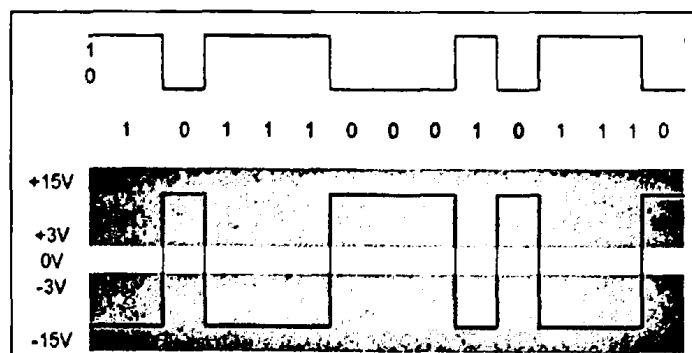
Η συσκευή αυτή κατασκευάστηκε στα πλαίσια της παρούσας διπλωματικής εργασίας. Το κεντρικό σημείο της κατασκευής, είναι ο μικροελεγκτής



Σχήμα 25: Ο σταθμός προσάρτησης αισθητηρίων και ενεργοποιητών αναλυτικά

ATmega16 και γύρω από αυτόν έχουν αναπτυχθεί τα υπόλοιπα τμήματά της (σχήμα 25). Η πλακέτα της κατασκευής είναι διπλής όψης και για τον σχεδιασμό της χρησιμοποιήθηκε το πρόγραμμα Orcad 9.2.

Η συσκευή μπορεί να τροφοδοτηθεί με συνεχή τάση από 8 έως 20 Volts [34]. Η τάση σταθεροποιείται στα 5 volts με σταθεροποιητή τάσης. Στη συγκεκριμένη περίπτωση χρησιμοποιείται ο σταθεροποιητής LM7805CV της εταιρίας ST και η συσκευή τροφοδοτείται με 12 volts από τον σταθμό προσάρτησης

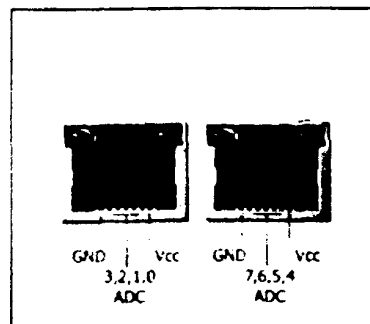


Σχήμα 26: Η επικοινωνία RS-232

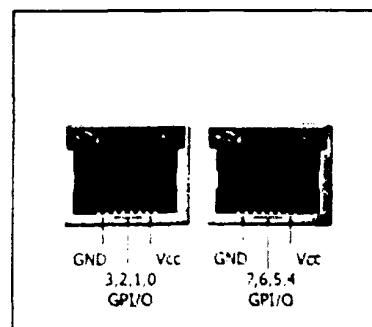
αισθητηρίων. Η σταθεροποιημένη τάση των 5 volts, χρησιμοποιείται για να τροφοδοτήσει την πλακέτα αλλά και τα αισθητήρια ή οποιαδήποτε άλλη

συσκευή που πιθανόν είναι συνδεδεμένη στον σταθμό. Έτσι τίθεται περιορισμός στην τάση λειτουργίας των συνδεδεμένων αισθητηρίων ή συσκευών. Δηλαδή, η τάση τους δεν πρέπει να ξεπερνά τα 5 Volts. Ο μικροελεγκτής, θέτει επίσης ένα δεύτερο περιορισμό, καθώς το συνολικό ρεύμα που μπορεί να προσφέρει στις συσκευές που είναι συνδεδεμένες στον μετατροπέα ADC και στην θύρα GPI/O, δεν πρέπει να ξεπερνά τα 200 mA [15]. Η συσκευή διαθέτει οπτική ένδειξη λειτουργίας και τροφοδοσίας. Η συνεχόμενη πράσινη φωτεινή ένδειξη, δείχνει ότι το σύστημα τροφοδοτείται και η διακεκομμένη πορτοκαλί ότι ο μικροελεγκτής λειτουργεί κανονικά.

Ο μικροελεγκτής επικοινωνεί με τον σταθμό ασύρματης διασύνδεσης Wi - Fi μέσω σειριακής επικοινωνίας RS-232, που απαιτεί τάση η οποία κυμαίνεται από +9 έως -9 volts (σχήμα 26), ενώ ο μικροελεγκτής απαιτεί 0 - 5 volts [16]. Έτσι, χρησιμοποιείται για την επικοινωνία ένας οδηγός, ο οποίος μετατρέπει το σήμα από +9 -9 volts σε 0 +5volts. Στη συγκεκριμένη ανάπτυξη χρησιμοποιείται ο AD233 της εταιρίας Analog Devices.

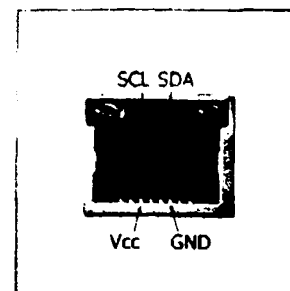


Σχήμα 27: Οι συνδετήρες ADC



Σχήμα 28: Οι συνδετήρες GPI/O

Η διασύνδεση των αισθητηρίων ή συσκευών που συνδέονται στις θύρες I²C, ADC και GPI/O τμήματα του μικροελεγκτή, πρέπει να προσφέρει μηχανική αντοχή και συμβατότητα με υπάρχουσες καλωδιώσεις. Για το λόγο αυτό, επιλέχθηκαν συνδετήρες τύπου RJ-45. Οι συνδετήρες αυτοί (σχήματα 27, 28 και 29), προσφέρουν μηχανική αντοχή και συμβατότητα με δομημένη καλωδίωση UTP⁹ ή STP¹⁰. Στο σχήμα, παρουσιάζονται και οι ακροδέκτες των συνδετήρων. Στον μετατροπέα ADC και στη θύρα GPI/O, η συνδεσμολογία είναι κοινή. κάθε συνδετήρας



Σχήμα 29: Ο συνδετήρας I2C

⁹ Unshielded twisted pairs

¹⁰ Shielded twisted pairs



έχει οκτώ pins, από τα οποία δύο είναι τροφοδοσία, δύο γείωση και τα υπόλοιπα τέσσερα χρησιμοποιούνται για τη μεταφορά του σήματος. Οπότε αναλογούν δύο συνδετήρες για τον μετατροπέα ADC και δύο για τη θύρα GPI/O. Για το δίαυλο I²C, χρησιμοποιείται ένας συνδετήρας. Από τα 8 pins που έχει, δύο χρησιμοποιούνται για τα σήματα SCL και SDA, ένα για τροφοδοσία 5 volts και ένα για γείωση. Τέλος, συνδετήρας τύπου RJ-45 χρησιμοποιείται και για την σειριακή επικοινωνία RS-232 για λόγους συμβατότητας με δομημένη καλωδίωση, όπως ήδη αναφέρθηκε. Από τα 8 pins, δύο χρησιμοποιούνται για τα σήματα εκπομπής και λήψης, Tx και Rx αντίστοιχα, και ένα για γείωση.

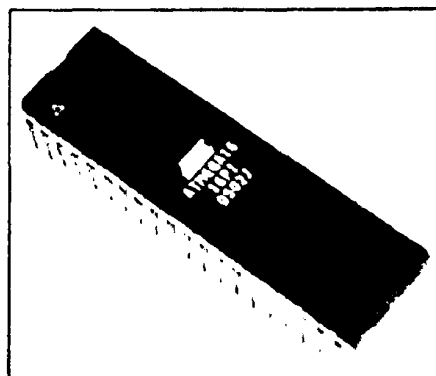
Τέλος, σε περίπτωση που ο μικροελεγκτής εισέλθει σε εσφαλμένη λειτουργία, εκτός από αυτόματη επανεκκίνηση με τον Watchdog timer, μπορεί να γίνει και χειροκίνητα με πλήκτρο reset (σχήμα 25).

Στην πλακέτα της συσκευής, υπάρχουν αναμονές για φωτεινές ενδείξεις (leds) καθώς και για έναν επιπλέον συνδετήρα τύπου RJ-45. Στον τελευταίο συνδετήρα, έχουν συνδεθεί δύο pins της θύρας C (pin6 και pin7) του μικροελεγκτή καθώς και τα αντίστοιχα pins της διασύνδεσης JTAG.

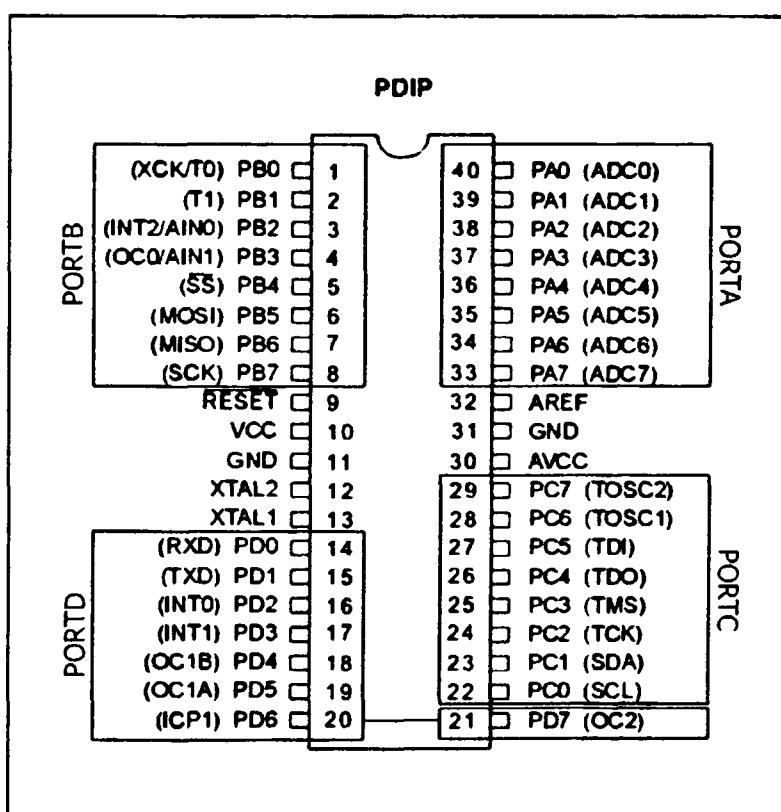
Πρέπει να σημειωθεί, πως το κουτί του σταθμού διασύνδεσης αισθητηρίων, είναι πολυχρηστικό. Δηλαδή, μπορεί να προσαρτηθεί σε ηλεκτρολογικό πίνακα, να προσαρτηθεί σε τοίχο καθώς και να χρησιμοποιηθεί και ως επιτραπέζια συσκευή. Όμως δεν είναι αδιάβροχο, οπότε προορίζεται για εσωτερική χρήση ή εξωτερική αλλά προστατευμένο σε κάποιο πίνακα.

2.5 Ο μικροελεγκτής του συστήματος

Οι σταθμοί προσάρτησης αισθητηρίων που συνδέονται στους ασυρμάτους σταθμούς διασύνδεσης Wi – Fi, βασίζουν τη λειτουργία τους στην ανάπτυξη των λειτουργιών του μικροελεγκτή που περιέχουν. Ο μικροελεγκτής αυτός είναι ο ATmega16 της οικογένειας AVR-8bit της εταιρίας ATMEL. Μπορεί να λειτουργήσει με μέγιστη ταχύτητα τα 16 MHz και τάση τροφοδοσίας τα 5 Volts. Είναι αρχιτεκτονικής RISC και περιέχει τρεις μνήμες, 16Kbytes In-System Self Programmable Flash μνήμη, 512 Bytes EEPROM μνήμη και 1KByte RAM μνήμη. Διαθέτει πλειάδα περιφερειακών όπως μετατροπέα αναλογικού σε ψηφιακό 10-bits, δίαυλο σειριακής επικοινωνίας I²C, σειριακή επικοινωνία USART, επικοινωνία SPI, επικοινωνία JTAG μέσω της οποίας είναι εφικτός ο προγραμματισμός του μικροελεγκτή και



Σχήμα 30: Ο μικροελεγκτής του συστήματος

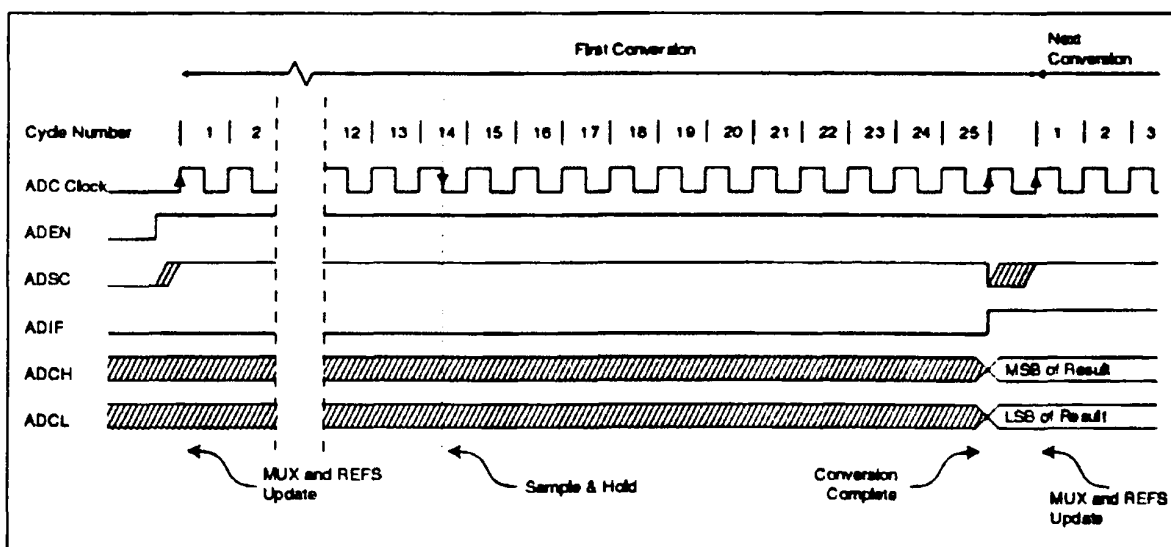


Σχήμα 31: Οι θύρες του μικροελεγκτή

προγραμματιστεί κάθε pin της θύρας ξεχωριστά, έτσι ώστε στην ίδια θύρα να έχουμε pins εισόδου και εξόδου ταυτόχρονα. Όταν γράφεται στον καταχωρητή DDRX λογικό '1' τότε το αντίστοιχο pin λειτουργεί ως είσοδος, ενώ με λογικό '0' λειτουργεί ως έξοδος. Και τα 32 pins έχουν εσωτερικά, αντιστάσεις pull-up και έτσι μπορούν να προσφέρουν ή να δεχθούν ρεύμα 20mA. Συνολικά όμως όλες οι θύρες δεν μπορούν να δεχτούν ή να προσφέρουν ταυτόχρονα περισσότερα από 200mA [15].

2.5.2 Ο μετατροπέας αναλογικού σε ψηφιακό

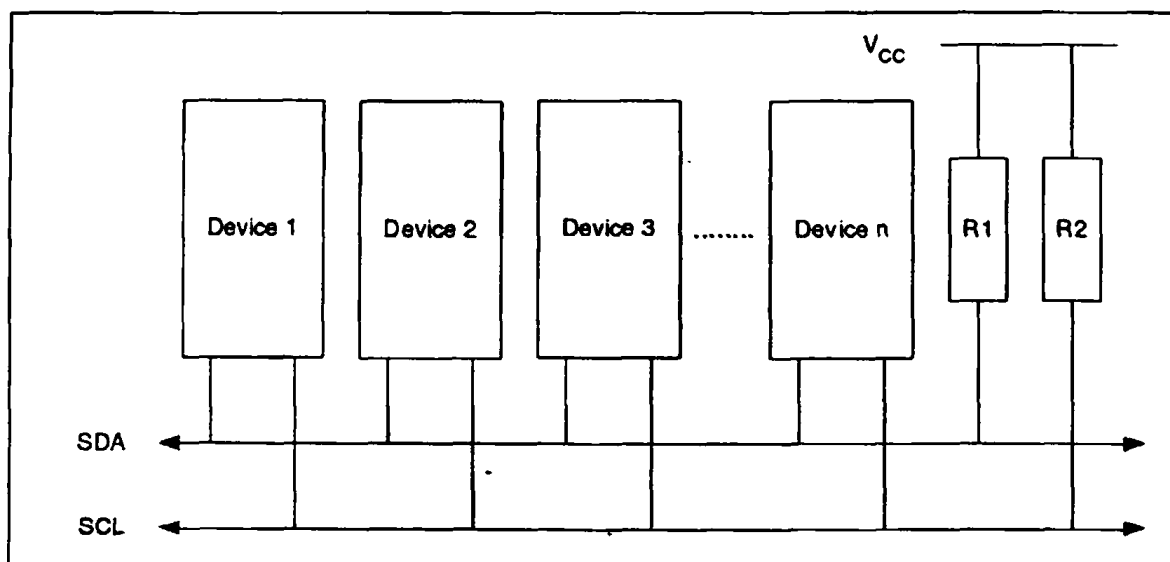
Η θύρα A εκτός από τη βασική λειτουργία εισόδου – εξόδου, λειτουργεί και ως μετατροπέας αναλογικού σε ψηφιακό (ADC), όπως φαίνεται στο σχήμα 32. Κάθε pin της θύρας, μπορεί να λειτουργήσει σαν είσοδος οπότε ο ADC είναι 8 καναλιών. Η μετατροπή του αναλογικού σήματος, γίνεται σε 10 bits και το σήμα εισόδου μπορεί να είναι από 0 έως 5 Volts. Η τάση αναφοράς V_{ref} βάσει της οποίας γίνεται η μετατροπή του αναλογικού σήματος, μπορεί να ορίζεται εξωτερικά (pin V_{ref}), αλλά μπορεί να είναι εσωτερική προγραμματίζοντας στον καταχωρητή ADMUX τα bits REFSn. Γενικά όμως, πρέπει να ισχύει $V_{in} < V_{ref}$.



Σχήμα 33: Χρονισμός του ADC για μία μετατροπή

Για να ξεκινήσει μια μετατροπή, πρέπει να γραφεί λογικό '1' στο bit ADSC. Το bit αυτό παραμένει σε κατάσταση '1' όσο χρόνο διαρκεί η μετατροπή και μηδενίζεται αυτόματα όταν τελειώσει η διαδικασία (σχήμα 33). Εναλλακτικά, η μετατροπή μπορεί να ξεκινήσει αυτόματα με σκανδαλισμό από το bit ADC Auto Trigger Enable (ADCATE) του καταχωρητή ADCSRA. Η μετατροπή ξεκινά

όταν ανιχνευτεί στο bit ADCATE θετικό μέτωπο παλμού. Ο χρόνος που διαρκεί μια μετατροπή εξαρτάται από τη συχνότητα λειτουργίας του ρολογιού του ADC, η οποία ορίζεται από το χρήστη. Οι τιμές που μπορεί να πάρει είναι από 50 έως 200 KHz για ανάλυση 10-bit. Για μικρότερη των 10-bit ανάλυση, το ρολόι μπορεί να λειτουργήσει και σε υψηλότερες συχνότητες. Ο ADC διαθέτει και δύο διαφορεικές εισόδους, μία στα pins ADC0 - ADC1 και μία στα pins ADC2 - ADC3. Οι διαφορεικές εισόδους, έχουν προγραμματιζόμενη απολαβή 1X (0dB), 10X (20dB) και 200X (46dB). Για απολαβή 1X ή 10X, η ανάλυση της μετατροπής είναι 8-bit ενώ για απολαβή 200X η ανάλυση πέφτει στα 7-bit. Η

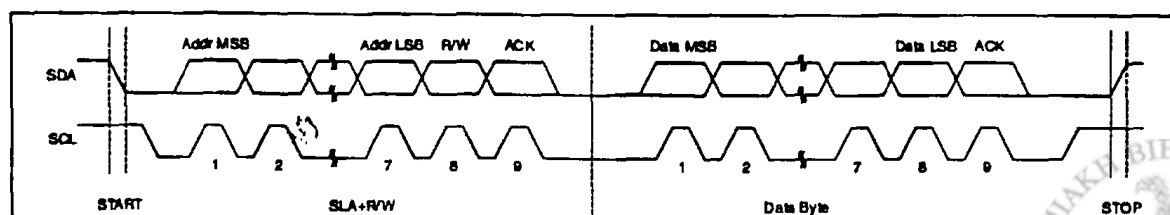


Σχήμα 34: Ο δίαυλος I²C

επιλογή του επιθυμητού καναλιού και απολαβής γίνεται από τα bits MUX4,3,2,1,0 του καταχωρητή ADMUX [15,27].

2.5.3 Ο δίαυλος I²C

Ένα ακόμη περιφερειακό του μικροελεγκτή που χρησιμοποιείται στο σύστημα, είναι ο δίαυλος σειριακής επικοινωνίας I²C και βρίσκεται στα pins 0 και 1 της θύρας C. Ο δίαυλος I²C είναι ένα είδος σειριακής



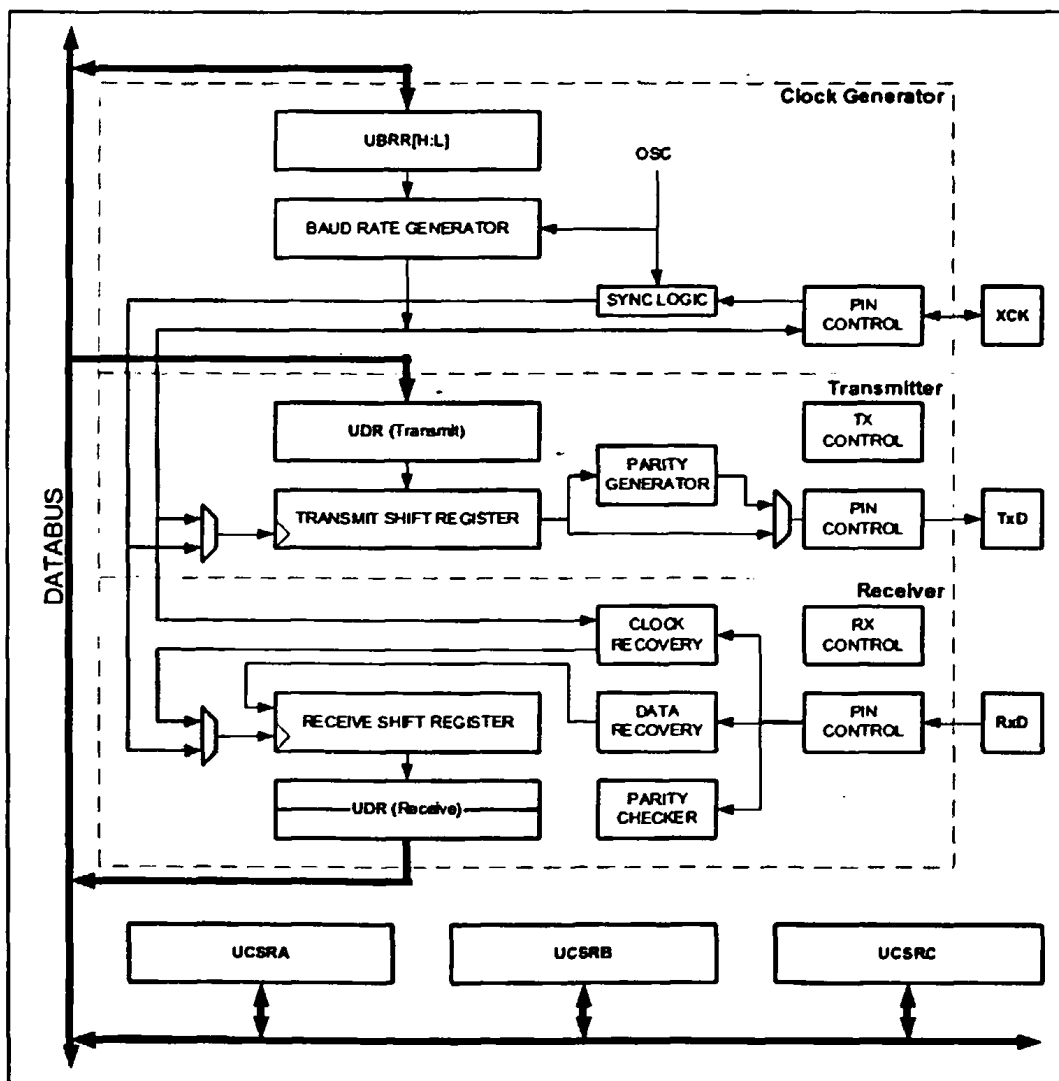
Σχήμα 35: Τυπική μετάδοση δεδομένων

διευθυνοδοτούμενης επικοινωνίας. Η διεύθυνση έχει μέγεθος 7-bits και επομένως το πλήθος των συσκευών που μπορούν να συνδεθούν σε αυτόν, είναι 2^7 δηλαδή 128. Όμως επειδή η διεύθυνση 0 έχει οριστεί ως γενική διεύθυνση κλήσης όλων των συσκευών, το πραγματικό πλήθος συνδεόμενων συσκευών στον δίαυλο, είναι 127. Ο δίαυλος αποτελείται από δυο γραμμές οι οποίες είναι συνδεδεμένες σε αντιστάσεις pull-up, εξωτερικά (σχήμα 34). Η μία γραμμή (SCL), χρησιμοποιείται για τη μεταφορά του σήματος χρονισμού (pin 0) και η άλλη για τη μεταφορά των δεδομένων (pin 1). Στο σχήμα 35, παρουσιάζεται μια τυπική μετάδοση δεδομένων. Πρώτα εκπέμπεται η κατάσταση εκκίνησης (start). Ακολούθως εκπέμπονται τα 7-bits της διεύθυνσης. Μετά, ένα bit που δηλώνει αν στη συσκευή που καλείται θέλουμε να γράψουμε δεδομένα, οπότε αποστέλλεται λογικό '0', ή να διαβάσουμε δεδομένα από αυτή, οπότε αποστέλλεται λογικό '1', και ακολουθούν τα δεδομένα. Τέλος, εκπέμπεται η κατάσταση τερματισμού (stop). Η ταχύτητα μετάδοσης των δεδομένων φτάνει μέχρι και τα 400 KHz. Για να λειτουργήσει ο δίαυλος I²C, απαιτείται μία τουλάχιστον συσκευή να λειτουργεί ως master και οι υπόλοιπες που είναι συνδεδεμένες, ως slave. Η master συσκευή είναι υπεύθυνη για την εκπομπή της κατάστασης start και stop, δηλαδή ξεκινά και τερματίζει μια εκπομπή δεδομένων. Υπάρχει το ενδεχόμενο σε ένα δίαυλο να υπάρχουν περισσότεροι του ενός masters. Στη περίπτωση αυτή, μόνο ένας μπορεί να εκπέμπει κάθε φορά και όλοι τους πρέπει να έχουν συγχρονισμένο το σήμα SCL. Οι καταχωρητές που χρησιμοποιούνται για το δίαυλο I²C, είναι ο TWBR για τον καθορισμό της ταχύτητας μετάδοσης δεδομένων, ο TWCR για τον καθορισμό των παραμέτρων λειτουργίας (start, stop, enable κλπ), ο TWSR που δείχνει την κατάσταση του διαύλου ανάλογα με την τιμή του και τέλος ο TWDR στον οποίο γράφονται τα δεδομένα προς αποστολή και αυτά που παραλαμβάνονται [15,20,27].



2.5.4 Η σύγχρονη/ασύγχρονη σειριακή επικοινωνία

Ο μικροελεγκτής, ανάμεσα στα υπόλοιπα περιφερειακά του, περιέχει και ασύγχρονη/σύγχρονη σειριακή επικοινωνία (USART), στα pins 0 (txd) και 1 (rxd) της θύρας D (σχήμα 36). Είναι πλήρως αμφίδρομη επικοινωνία με μέγιστη ταχύτητα μετάδοσης δεδομένων 2 Mbps για συχνότητα λειτουργίας του μικροελεγκτή τα 16 MHz. Ο χρονοισμός μπορεί να είναι είτε μορφής Master είτε μορφής Slave. Οι καταχωρητές που χρησιμοποιούνται είναι, οι UCSRA, UCSRB και UCSRC για έλεγχο της κατάστασης και των ρυθμίσεων της επικοινωνίας, ο UDR για τα δεδομένα προς αποστολή και από λήψη, οι UBRRL και UBRRH για τον καθορισμό της ταχύτητας μετάδοσης των δεδομένων [15].



Σχήμα 36: Η σειριακή επικοινωνία USART

Στη παρούσα διπλωματική, ο μικροελεγκτής χρησιμοποιείται με συγκεκριμένο τρόπο. Στο δίαυλο I²C, λειτουργεί ως master του διαύλου και με

ταχύτητα μετάδοσης δεδομένων 100 KHz. Όλες οι υπόλοιπες συσκευές που συνδέονται στο δίκτυο, λειτουργούν ως slave. Ο μετατροπέας αναλογικού σε ψηφιακό, λειτουργεί σε όλα του τα κανάλια εισόδου με απολαβή 1X και εξωτερική τάση αναφοράς. Η ταχύτητα μετατροπής είναι ορισμένη σε 1/128 της συχνότητας λειτουργίας του μικροελεγκτή, δηλαδή 125 KHz και η ανάλυση που χρησιμοποιείται είναι των 8-bits. Τέλος, η USART σειριακή επικοινωνία, λειτουργεί ασύγχρονα, με ταχύτητα μετάδοσης δεδομένων 9.6 Kbps, με μέγεθος πακέτου δεδομένων τα 8-bits και 1-bit τερματισμού.



2.6 Τα αισθητήρια του συστήματος

Στο σύστημα της παρούσας διπλωματικής, μπορούν να συνδεθούν αισθητήρια ψηφιακά αλλά και αναλογικά. Τα ψηφιακά αισθητήρια μπορούν να είναι I²C ή παράλληλης εξόδου. Στα αναλογικά αισθητήρια, πρέπει η έξοδος τους, να κυμαίνεται από 0 έως 5 Volts.

Το εύρος των αισθητηρίων που μπορούν να συνδεθούν σε ένα τέτοιο σύστημα, είναι μεγάλο. Καθώς το σύστημα έχει κατασκευαστεί για αυτοματισμό και τηλεμετρία, μπορούν να συνδεθούν αισθητήρια θερμοκρασίας, υγρασίας, ανίχνευσης κίνησης, ανίχνευσης επικίνδυνων αερίων, μαγνητικές επαφές θυρών και γενικά οτιδήποτε άλλο μπορεί να εγκατασταθεί σε ένα κλειστό χώρο όπως ένα εργαστήριο και όχι μόνο.

Το σύστημα, εκτός από λήψη δεδομένων, μπορεί να πραγματοποιήσει και αποστολή όσο αναφορά το δίαυλο I²C και την ψηφιακή θύρα (General Purpose Input Output). Δηλαδή στα τμήματα αυτά, έχουμε αμφίδρομη επικοινωνία. Συνεπώς το εύρος των συσκευών που μπορούν να συνδεθούν στο σύστημα, αυξάνεται αρκετά.

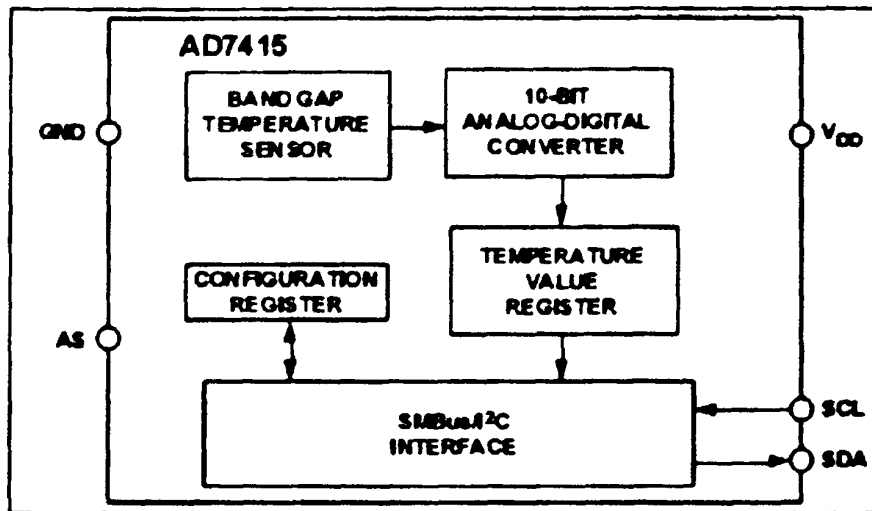
Εκτός από αισθητήρια, μπορούν να συνδεθεί οποιαδήποτε συσκευή επικοινωνίας I²C ή απλής ψηφιακής. Συσκευές επικοινωνίας I²C, όπως αναγνώστες μαγνητικών καρτών, μνήμες, οθόνες υγρών κρυστάλλων, ηλεκτρονικές κλειδαριές. Επιπλέον στη θύρα GPI/O, μπορεί να συνδεθεί οποιοσδήποτε ενεργοποιητής, όπως μετρητές κλπ, αλλά και συσκευές με ηλεκτρονόμους¹¹ (relay). Συσκευές δηλαδή, που μετατρέπουν το ψηφιακό σήμα σε ηλεκτρομηχανική διέγερση.

Στο σύστημα της παρούσας διπλωματικής, έχουν χρησιμοποιηθεί πειραματικά, ένας αισθητήρας θερμοκρασίας I²C, ένας αναλογικός αισθητήρας υγρασίας και ένας ανιχνευτής κίνησης.

- Ο αισθητήρας θερμοκρασίας I²C είναι ο AD7415 της εταιρίας Analog Device (σχήμα 37). Το εύρος μέτρησης θερμοκρασίας, είναι από -40 °C έως +125 °C με ακριβέτα μέτρησης ±0,5°C στους +40°C. Διαθέτει μετατροπέα αναλογικού σε ψηφιακό των 10-bits, οπότε η ψηφιακή ένδειξη θερμοκρασίας

¹¹ Actuators

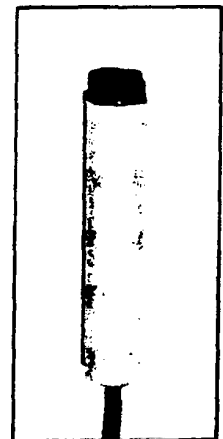
είναι λέξη έως 10-bits. Στη παρούσα εφαρμογή όμως, χρησιμοποιείται λέξη των 8-bits. Ο χρόνος που απαιτείται για μία μετατροπή είναι περίπου 29 μsec και μπορεί να τροφοδοτηθεί με τάση από 2,7 έως 5 Volts. Η διαδικασία μέτρησης και μετατροπή της τιμής σε ψηφιακή λέξη, ξεκινά όταν ο master του διαύλου I²C καλέσει τη διεύθυνση του αισθητηρίου, αποστέλλει bit εγγραφής



Σχήμα 37: Το αισθητήριο AD7415

και την τιμή 40 Hex. Τότε ο αισθητήρας πραγματοποιεί τη διαδικασία μέτρησης και μετατροπής, περιμένοντας τον master να το ξανακαλέσει με bit ανάγνωσης. Έπειτα, πραγματοποιείται η μεταφορά της μέτρησης της θερμοκρασίας από τον αισθητήρα στον master του διαύλου I²C και η διαδικασία τερματίζεται [24].

- Ο αναλογικός αισθητήρας υγρασίας είναι ο HM1500LF της εταιρίας HUMIREL (σχήμα 38). Το εύρος μέτρησης είναι θεωρητικά, από 0% έως 100% με ακρίβεια μέτρησης $\pm 3\%$ και βήμα μέτρησης 0.4%. Η τάση τροφοδοσίας κυμαίνεται από 3 έως 7 Volts και η κατανάλωση περίπου 150 μA . Η τάση εξόδου κυμαίνεται πρακτικά, από 1,325 Volts έως 3.555 Volts για υγρασία από 10% έως 90% αντίστοιχα. Η απόκριση του αισθητήρα είναι γραμμική και αυτό τον καθιστά κατάλληλο για απευθείας διασύνδεση με μικροελεγκτή. Ο αισθητήρας είναι αδιάβροχο και μπορεί να χρησιμοποιηθεί σε κλειστό και ανοιχτό χώρο. Ο χρόνος απόκρισής του, είναι 150 ms κατά την εκκίνηση και για



Σχήμα 38: Το αισθητήριο HM1500LF

πλήρως σταθεροποιημένη μέτρηση, έως 10 sec [25].

• Ο ανιχνευτής κίνησης, είναι ο Electronic PIR motion detector της εταιρίας Kopp (σχήμα 39). Η γωνία δράσης του αισθητήρα, είναι 270° με ακτίνα περίπου 15 μέτρα. Το φορτίο που μπορεί να δεχτεί κυμαίνεται από 60W μέχρι 1KW. Σε περίπτωση που ανιχνευτεί κίνηση, ο αισθητήρας μπορεί να τροφοδοτήσει το φορτίο του από 5 sec έως 18 min. Ο αισθητήρας μπορεί να λειτουργήσει στο φως της ημέρας, σε σκοτάδι αλλά και σε ημίφως. Αυτό οφείλεται στο κατώφλι εναλλαγής, το οποίο ρυθμίζεται αναλογικά. Ο δείκτης προστασίας μπορεί να χρησιμοποιηθεί τόσο σε εσωτερικό χώρο όσο και σε εξωτερικό, με δείκτη προστασίας IP 44. η τάση τροφοδοσίας του είναι 220 Volts.



Σχήμα 39: Ο Ανιχνευτής κίνησης

ΚΕΦΑΛΑΙΟ 3^ο.

Το λογισμικό του συστήματος.

Για τις ανάγκες του προγραμματισμού του κυκλώματος, χρησιμοποιήθηκαν τρεις γλώσσες προγραμματισμού. Για τον προγραμματισμό του μικροελεγκτή χρησιμοποιήθηκε η γλώσσα Assembly. Για τη δημιουργία του προγράμματος διασύνδεσης (interface), χρησιμοποιήθηκε η γλώσσα προγραμματισμού υπερκειμένου HTML. Για την επικοινωνία του προγράμματος διασύνδεσης με τον μικροελεγκτή, χρησιμοποιήθηκε ένα πρότυπο της γλώσσας προγραμματισμού C, το **Common Gateway Interface (CGI)**.

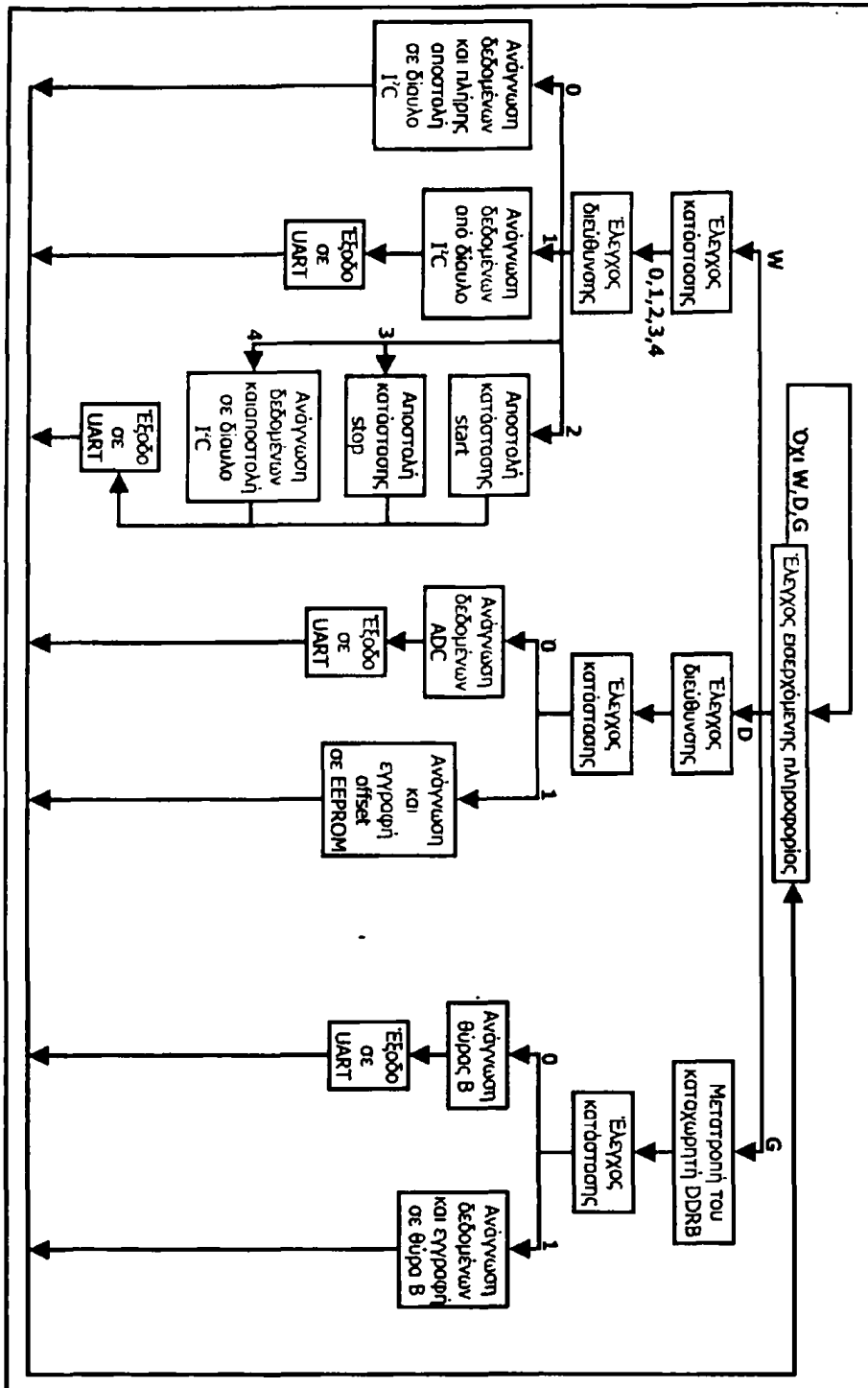
3.1 Το λογισμικό της εφαρμογής σε γλώσσα προγραμματισμού Assembly.

Οι μικροελεγκτές είναι ηλεκτρονικές μηχανές που μπορούν να προγραμματιστούν άμεσα. Για να επιτευχθεί αυτό, απαιτείται μία γλώσσα χαμηλού επιπέδου και ένας μεταφραστής (compiler). Στη παρούσα διπλωματική, η γλώσσα χαμηλού επιπέδου που χρησιμοποιείται είναι η Assembly και μεταφραστής ο AVR Studio 4 της εταιρίας Atmel.

Για την εγγραφή του προγράμματος, υπάρχουν τρεις μέθοδοι. Η μέθοδος του γραμμικού λογισμού, όπου το πρόγραμμα γράφεται σε μια ρουτίνα μεγάλου μήκους. Η μέθοδος του αρθρωτού λογισμού, όπου υπάρχει μία κύρια ρουτίνα και αρκετές μικρότερες υπορουτίνες οι οποίες καλούνται διαρκώς από την κύρια. Στη μέθοδο αυτή, κάθε υπορουτίνα αναλαμβάνει ένα μικρό μέρος της λειτουργίας του προγράμματος. Τέλος, η συνδυαστική μέθοδος, όπου γίνεται συνδυασμός των δύο παραπάνω μεθόδων. Υπάρχει μια όχι πολύ μεγάλη κύρια ρουτίνα και αρκετές υπορουτίνες, οι οποίες αναλαμβάνουν πλήρως ένα τμήμα της συνολικής λειτουργίας του προγράμματος [11,12,13].



Με την πρώτη μέθοδο του γραμμικού λογισμού, ο χρόνος εκτέλεσης του



Σχήμα 40: Block διάγραμμα του προγράμματος σε Assembly

προγράμματος είναι πολύ μικρός, αφού η ροή είναι αδιάλειπτη. Όμως, επειδή το πρόγραμμα είναι γραμμένο σε μία μόνο ρουτίνα, κάποια κομμάτια του επαναλαμβάνονται. Επίσης, είναι αρκετά δύσκολη η παρέμβαση σε αυτό λόγω του μεγέθους του. Με την δεύτερη μέθοδο του αρθρωτού λογισμού, το μέγεθος του προγράμματος είναι πολύ μικρότερη, αφού κανένα κομμάτι δεν

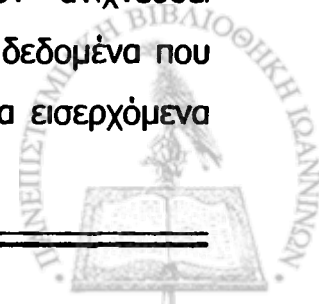
επαναλαμβάνεται αλλά όταν χρειάζεται, απλά καλείται. Η παρέμβαση από τον προγραμματιστή είναι πολύ εύκολη, αφού όλες οι λειτουργίες είναι χωρισμένες σε μικρές υπορουτίνες. Όμως, ο χρόνος εκτέλεσης του προγράμματος είναι κατά πολύ μεγαλύτερος, αφού η ροή συνέχεια διακόπτεται για να κληθούν οι διάφορες υπορουτίνες. Και στη περίπτωση που το πρόγραμμα περιέχει αρκετές και συνεχόμενες μεταπηδήσεις, υπάρχει κίνδυνος ακόμα και για σφάλμα ροής από το μικροελεγκτή. Με την τρίτη μέθοδο του συνδυασμού, πετυγχάνεται μία μέση λύση. Κάθε μέθοδος χρησιμοποιείται για διαφορετικούς σκοπούς. Για παράδειγμα η γραμμική μέθοδος, θα επιλεγεί στην περίπτωση που απαιτείται γρήγορη εκτέλεση για μια όχι πολύ σύνθετη λειτουργία, όπως η διαμόρφωση ASK. Η μέθοδος του αρθρωτού λογισμού, στην περίπτωση που θέλουμε ο μικροελεγκτής να διατελέσει μία πολύπλοκη διεργασία αλλά σε αδιάφορο χρόνο, όπως σε μια ηλεκτρονική κλειδαριά με πληκτρολόγιο και οθόνη.

3.2 Το λογισμικό του μικροελεγκτή σε γλώσσα Assembly

Στην παρούσα διπλωματική εργασία, ο μικροελεγκτής όπως φαίνεται και στο σχήμα 40, πρέπει να ελέγχει διαρκώς για εισερχόμενα δεδομένα και εφόσον αυτά είναι τα επιθυμητά, να προχωρά στο αντίστοιχο τμήμα, δηλαδή στον δίαυλο I²C, στον μετατροπέα ADC ή στη θύρα GPI/O. Συνεπώς, η μέθοδος που ικανοποιεί τις απαιτήσεις του συστήματος είναι η συνδυαστική, διότι έτσι επιτυγχάνεται ικανοποιητική ταχύτητα, μικρότερης έκτασης κώδικας και ευκολία στην παρέμβαση από τον προγραμματιστή.

Όπως φαίνεται και στο σχήμα 40, το πρόγραμμα του μικροελεγκτή αποτελείται από τα εξής επιμέρους τμήματα. Το τμήμα της σειριακής ασύγχρονης επικοινωνίας, το τμήμα ελέγχου των εισερχόμενων δεδομένων, το τμήμα λειτουργίας του διαύλου I²C, το τμήμα του μετατροπέα ADC και το τμήμα της θύρας GPI/O.

Η κεντρική ρουτίνα του προγράμματος, ελέγχει για εισερχόμενα δεδομένα από τη σειριακή ασύγχρονη επικοινωνία (UART). Εφόσον ανιχνεύσει δεδομένα, τα συγκρίνει με τους χαρακτήρες w, d, g και t. Τα δεδομένα που έρχονται είναι χαρακτήρες ASCII οπότε η ρουτίνα συγκρίνει τα εισερχόμενα

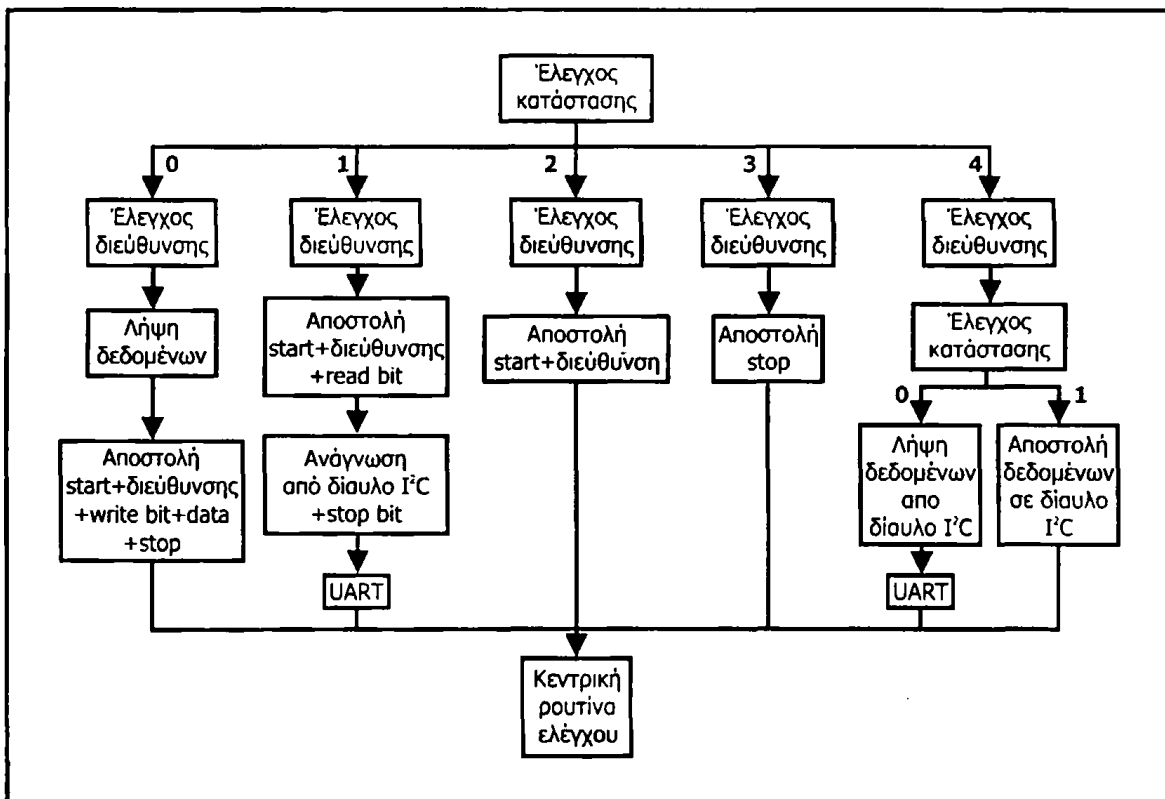


δεδομένα με τις δεκαεξαδικές τιμές των w, d και g, 77, 64, 67 και 74 αντίστοιχα. Αν η σύγκριση καταλήξει σε θετικό αποτέλεσμα τότε η κεντρική ρουτίνα μεταπηδά τη ροή του προγράμματος ανάλογα με το χαρακτήρα που ανιχνεύτηκε. Στην περίπτωση όμως, που η σύγκριση καταλήξει σε αρνητικό αποτέλεσμα, τότε το πρόγραμμα μεταπηδά στην αρχική κατάσταση αναμονής δεδομένων από τη UART και η διαδικασία επαναλαμβάνεται.

3.2.1 Το λογισμικό του μικροελεγκτή για το δίαυλο I²C

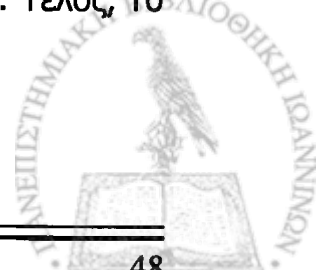
Αν η κεντρική ρουτίνα ανιχνεύσει το χαρακτήρα w, τότε το πρόγραμμα μεταπηδά στην υπορουτίνα του διαύλου I²C. Εκεί, αρχικά πραγματοποιείται έλεγχος κατάστασης, δηλαδή ποια εργασία από τις 5 συνολικά, πρέπει να ακολουθηθεί.

Αν ανιχνευθεί '0' τότε ακολουθείται η εργασία εγγραφής δεδομένων στο



Σχήμα 41: Ο κώδικας του διαύλου I²C

δίαυλο. Η υπορουτίνα διενεργεί λήψη της διεύθυνσης του παραλήπτη και των δεδομένων προς αποστολή από την UART. Έπειτα, στέλνει στο δίαυλο I²C την ακολουθία |start + διεύθυνση + bit εγγραφής + δεδομένα + stop|. Τέλος, το πρόγραμμα επιστρέφει στην κύρια ρουτίνα ελέγχου κατάστασης.

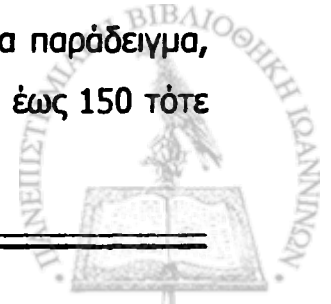


Αν ανιχνευθεί '1', τότε ακολουθείται η εργασία ανάγνωσης από το δίαυλο I²C. Η υπορουτίνα διενεργεί λήψη της διεύθυνσης παραλήπτη και αποστέλλει στο δίαυλο I²C, την ακολουθία |start+διεύθυνση+bit ανάγνωσης| και αναμένει για δεδομένα από το δίαυλο I²C. Όταν ληφθούν τα δεδομένα στέλνει στο δίαυλο stop και στην UART τα δεδομένα που παρελήφθησαν. Τέλος το πρόγραμμα επιστρέφει στην κεντρική ρουτίνα ελέγχου κατάστασης.

Αν ανιχνευθεί '2', τότε ακολουθείται η διαδικασία αποστολής κατάστασης εκκίνησης (start) στο δίαυλο. Η υπορουτίνα διενεργεί λήψη της διεύθυνσης παραλήπτη και αποστέλλει στο δίαυλο την ακολουθία |start+διεύθυνση|. Αφού αποσταλεί start, τότε το πρόγραμμα επιστρέφει στην κεντρική ρουτίνα ελέγχου. Αν ανιχνευθεί '3', τότε ακολουθείται η διαδικασία αποστολής κατάστασης τερματισμού (stop) στο δίαυλο. Αφού αποσταλεί stop, τότε το πρόγραμμα επιστρέφει στην κεντρική ρουτίνα ελέγχου. Τέλος, αν ανιχνευθεί '4', ακολουθείται η διαδικασία αποστολής ή λήψης δεδομένων μη τυποποιημένα, δηλαδή χωρίς κατάσταση start και stop, καθώς και χωρίς διεύθυνση παραλήπτη. Η λειτουργία αυτή, συνδυάζεται με τις δυο προηγούμενες. Δηλαδή πρέπει πρώτα να αποσταλεί στο δίαυλο κατάσταση start μαζί με τη διεύθυνση του παραλήπτη και το bit εγγραφής ή ανάγνωσης, έπειτα τα δεδομένα, ένα ή περισσότερα bytes και τέλος η κατάσταση stop. Στην περίπτωση ανάγνωσης, διενεργείται έλεγχος του διαύλου και τα δεδομένα που λαμβάνονται προωθούνται στην UART. Τέλος, το πρόγραμμα επιστρέφει στην κεντρική ρουτίνα ελέγχου (Παράρτημα Α, σελ. 37).

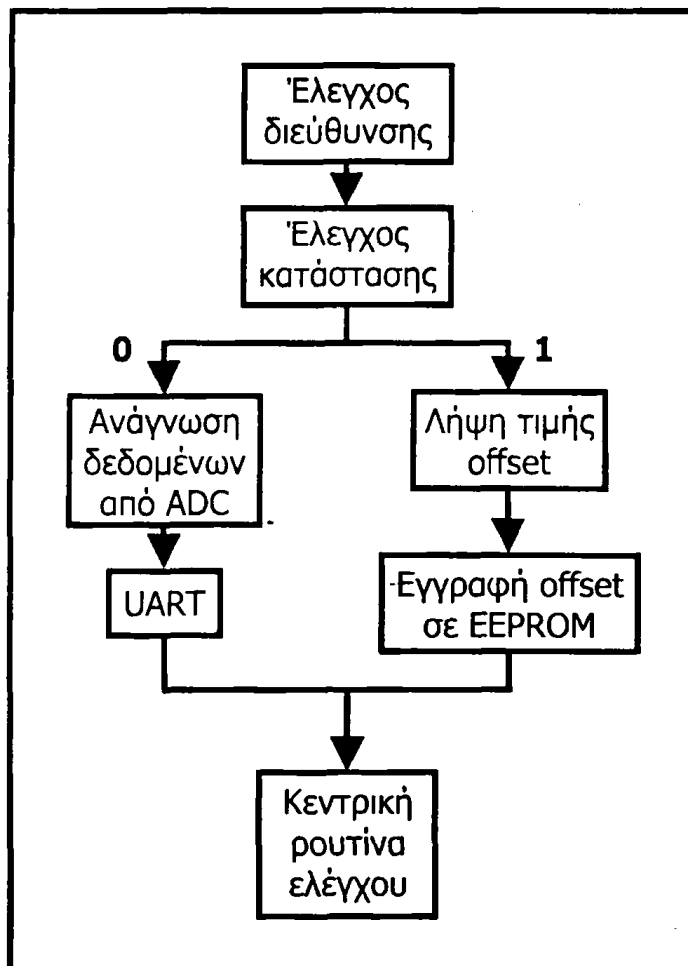
3.2.1 Το λογισμικό του μικροελεγκτή για τον μετατροπέα ADC

Αν η κεντρική ρουτίνα ελέγχου, ανιχνεύσει το χαρακτήρα d, τότε καλείται η υπορουτίνα του μετατροπέα ADC. Αρχικά πραγματοποιείται λήψη της διεύθυνσης του καναλιού (0,1,2...7). Έπειτα πραγματοποιείται έλεγχος κατάστασης, ανάγνωσης η εγγραφής της τιμής offset (σχήμα 42). Η τιμή offset, χρησιμοποιείται για να μεταβάλλει την περιοχή τιμών ορισμένων αισθητηρίων, σε περιοχή τιμών κατανοητή από τον χρήστη. Για παράδειγμα, αν ένα αισθητήριο υγρασίας χρησιμοποιεί περιοχή τιμών από 50 έως 150 τότε



αυτό πρέπει να μετατραπεί σε περιοχή τιμών από 0 έως 100 γιατί αντιπροσωπεύει ποσοστό υγρασίας.

Αν από τον έλεγχο κατάστασης προκύψει '0', τότε ακολουθείται η διαδικασία ανάγνωσης καναλιού. Αποστέλλεται στον μετατροπέα ADC η διεύθυνση του καναλιού που θέλουμε να διαβάσουμε, και ο μικροελεγκτής πραγματοποιεί μία μόνο ανάγνωση του συγκεκριμένου καναλιού. Έπειτα, τα δεδομένα που προκύπτουν από τη μέτρηση, εξέρχονται στην UART αφού αφαιρεθεί από αυτά η τιμή offset του συγκεκριμένου καναλιού και το

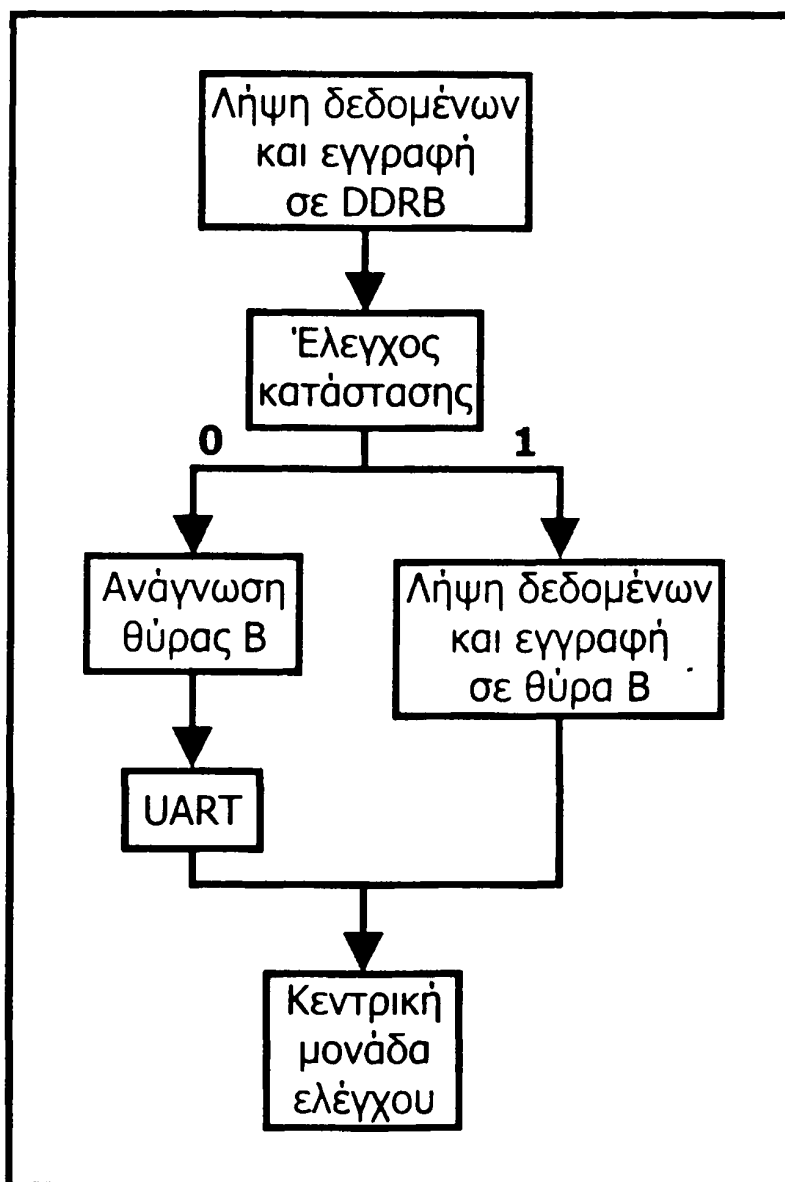


Σχήμα 42: Ο κώδικας του μετατροπέα ADC

πρόγραμμα επιστρέφει στην κεντρική ρουτίνα ελέγχου.

Αν από τον έλεγχο κατάστασης προκύψει '1', τότε ακολουθείται η διαδικασία εγγραφής της τιμής offset για το συγκεκριμένο κανάλι. Πραγματοποιείται λήψη της τιμής offset. Στη συνέχεια, αποστέλλεται στη μνήμη EEPROM του μικροελεγκτή η διεύθυνση που παραλήφθηκε προηγουμένως και στη θέση αυτή της μνήμης αποθηκεύεται η τιμή offset που

έχει ήδη παραληφθεί. Τέλος, το πρόγραμμα, επιστρέφει στην κεντρική ρουτίνα ελέγχου. Συνολικά, στη μνήμη EEPROM αποθηκεύονται 8 τιμές offset, όσα και τα κανάλια του ADC στις διευθύνσεις 0-7 (Παράρτημα Α, σελ. 118).



Σχήμα 43: Ο κώδικας της θύρας B (GPIO)

3.2.3 Το λογισμικό του μικροελεγκτή για τη θύρα GPIO

Αν η κεντρική ρουτίνα ελέγχου, ανιχνεύσει το χαρακτήρα g, τότε η ροή του προγράμματος μεταπηδά στην υπορουτίνα της θύρας B (GPIO) (Σχήμα 43). Αρχικά η υπορουτίνα λαμβάνει από την UART την τιμή που δείχνει ποια bits της θύρας θέλουμε να ορισθούν ως είσοδοι και ποια ως έξοδοι. Έπειτα η τιμή αυτή γράφεται στον καταχωρητή DDRB της θύρας και διενεργείται από την υπορουτίνα έλεγχος κατάστασης.

Από τον έλεγχο, αν προκύψει '0', τότε ακολουθείται η διαδικασία ανάγνωσης της θύρας B. Η υπορουτίνα διαβάζει τα δεδομένα της θύρας B, όσα bits αυτής έχουν ορισθεί ως είσοδοι, και αποστέλλει στην UART την τιμή αυτή. Έπειτα, το πρόγραμμα επιστρέφει στην κεντρική ρουτίνα ελέγχου.

Από τον έλεγχο κατάστασης, αν προκύψει '1', τότε ακολουθείται η διαδικασία εγγραφής στη θύρα B. Η υπορουτίνα λαμβάνει από την UART τα δεδομένα προς εγγραφή και έπειτα τα εξάγει στη θύρα B. Τέλος το πρόγραμμα, μεταπηδά στην κεντρική μονάδα ελέγχου (Παράρτημα A, σελ.121).

3.2.4 Το λογισμικό του μικροελεγκτή για τον έλεγχο σκανδαλισμού της κάμερας

Αν η κεντρική ρουτίνα ανιχνεύσει το χαρακτήρα t, τότε η ροή του προγράμματος μεταπηδά και πάλι στην υπορουτίνα ελέγχου της θύρας B (GPI/O). Η εργασία όμως που ακολουθεί ο μικροελεγκτής είναι αποκλειστικά ο έλεγχος του bit 0 της θύρας. Αν ανιχνευτεί λογικό '1', τότε επιστρέφει την τιμή '1' σε ASCII κωδικοποίηση, δηλαδή τον δεκαεξαδικό αριθμό 31. Αν ανιχνευτεί λογικό '0', τότε επιστρέφει την τιμή '0' σε ASCII κωδικοποίηση, δηλαδή το δεκαεξαδικό αριθμό 30 (Παράρτημα A, σελ.).

3.2.5 Το λογισμικό του μικροελεγκτή για τη σύγχρονη/ασύγχρονη σειριακή επικοινωνία

Ο έλεγχος και η λειτουργία της ασύγχρονης σειριακής επικοινωνίας, πραγματοποιούνται από τρεις συγκεκριμένες υπορουτίνες. Η πρώτη πραγματοποιεί την αρχικοποίηση της επικοινωνίας. Δηλώνεται η ταχύτητα μετάδοσης δεδομένων, που στην προκειμένη περίπτωση είναι 9.6 Kbps. Στη συνέχεια απενεργοποιείται ο έλεγχος ροής δεδομένων, απενεργοποιείται η λειτουργία ισοτιμίας (parity) και δηλώνεται ένα ως bit τερματισμού. Η δεύτερη και η τρίτη υπορουτίνα πραγματοποιούν τη λήψη και την αποστολή των δεδομένων. Στην υπορουτίνα λήψης γίνεται έλεγχος για μη διαβασμένα δεδομένα στον καταχωρητή λήψης. Αν υπάρχουν δεδομένα που δεν έχουν διαβαστεί, τότε η υπορουτίνα προχωρά στην ανάγνωσή τους. Διαφορετικά περιμένει μέχρι να υπάρξουν. Στη υπορουτίνα αποστολής, γίνεται έλεγχος του

καταχωρητή αποστολής. Αν είναι άδειος, τότε εγγράφονται σε αυτόν τα δεδομένα προς αποστολή και αποστέλλονται. Αν δεν είναι άδειος, τότε η υπορουτίνα περιμένει μέχρι να αδειάσει και προχωρά στην παραπάνω διαδικασία αποστολής (Παράρτημα Α, σελ. 113).

3.2.6 Το λογισμικό του μικροελεγκτή για τους χαρακτήρες ASCII

Τα δεδομένα που λαμβάνονται από της UART, είναι ακολουθίες ASCII, ενός γράμματος αρχικά και ακολούθως αριθμών. Επειδή όμως ο μικροελεγκτής δεν αντιλαμβάνεται τις τιμές ASCII των αριθμών, υπάρχει στο πρόγραμμα υπορουτίνα που πραγματοποιεί αυτήν τη μετατροπή. Διαβάζει την τιμή ASCII του αριθμού και αφαιρεί από την τιμή αυτή τον δεκαεξαδικό αριθμό 30. Το αποτέλεσμα που προκύπτει είναι ο αριθμός αυτούσιος. Για παράδειγμα ο αριθμός 3 σε χαρακτήρα ASCII μεταφράζεται σε δεκαεξαδικό 33. Οπότε αν αφαιρεθεί από το 33 το 30, τότε προκύπτει 3 (Παράρτημα Α, σελ. 114).

3.2.7 Το λογισμικό του μικροελεγκτή για τον WDT

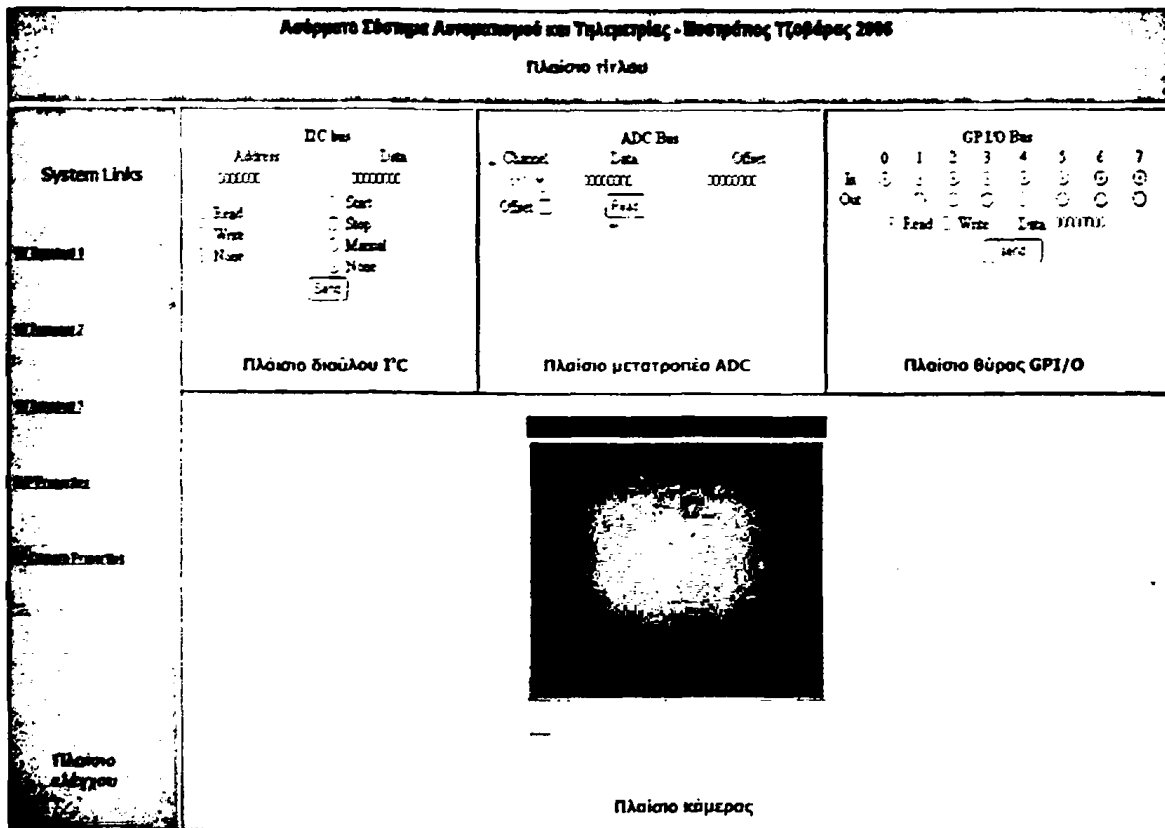
Τέλος στο πρόγραμμα της παρούσας διπλωματικής, χρησιμοποιείται και ο Watchdog timer. Είναι ένα από τα περιφερειακά του μικροελεγκτή και όταν ενεργοποιείται πρέπει να επανεκκινηθεί πριν ολοκληρωθεί ο χρόνος λειτουργίας του. Αν δεν επανεκκινηθεί, τότε ο Watchdog timer προκαλεί γενική επανεκκίνηση σε ολόκληρο το μικροελεγκτή. Η χρήση του είναι πολύ σημαντική γιατί αν ο μικροελεγκτής για οποιονδήποτε λόγο εισέλθει εσφαλμένα, σε ένα στέρμονα βρόγχο, τότε ο Watchdog timer δεν θα επανεκκινηθεί και ο μικροελεγκτής θα επανεκκινηθεί ξεκινώντας πάλι την προγραμματισμένη λειτουργία του. Στη προκειμένη περίπτωση, ο Watchdog timer έχει ορισθεί για κύκλο λειτουργίας 0.5 sec και επανεκκινείται σε κάθε υπορουτίνα [15] (Παράρτημα Α, σελ. 112).



3.3 Το λογισμικό της εφαρμογής σε HTML.

Η γλώσσα υπερκειμένου HTML (Hyper Text Markup Language), χρησιμοποιείται κυρίως για κατασκευή ιστοσελίδων στατικού τύπου. Δηλαδή, ιστοσελίδες οι οποίες απλά εμφανίζουν στον χρήστη κάποια δεδομένα. Απαραίτητη προϋπόθεση για την εμφάνιση των περιεχομένων των ιστοσελίδων, είναι η χρήση ενός φυλλομετρητή διαδικτύου (Web Browser).

Ο κώδικας μιας ιστοσελίδας, αποτελείται από ετικέτες (markup tags). Οι ετικέτες αυτές δείχνουν τον τρόπο με τον οποίο ο φυλλομετρητής θα χειριστεί τα περιεχόμενά τους. Υπάρχουν αρκετές κατηγορίες ετικετών. Για παράδειγμα, αυτές που δηλώνουν ότι τα περιεχόμενά τους είναι προς εμφάνιση, αυτές που δηλώνουν ότι τα περιεχόμενά τους είναι εντολές κλπ. Για παράδειγμα η ετικέτα `<html>.....</html>`, δηλώνει πως το περιεχόμενά της είναι κώδικας HTML. Η ετικέτα `.....`, δηλώνει πως τα περιεχόμενά της πρέπει να εμφανιστούν τονισμένα (bold) [8,9,31].



Σχήμα 44: Η ιστοσελίδα της συστήματος

Η ιστοσελίδα της παρούσας διπλωματικής, σχήμα 44, περιέχει επιμέρους πλαίσια τα οποία καλούνται από τον κεντρικό κώδικα. Δηλαδή, μέσα σε μία ιστοσελίδα, εμφανίζονται άλλες ιστοσελίδες. Η ετικέτα που πραγματοποιεί

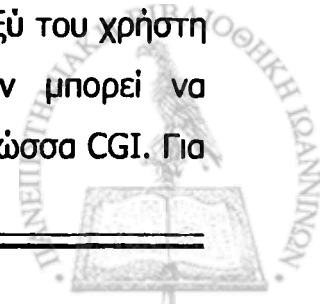
αυτή τη λειτουργία, είναι η `<frameset>` `</frameset>`. Δεσμεύει ένα τμήμα του παραθύρου της ιστοσελίδας και εκεί προβάλλει τα περιεχόμενά της. Το τμήμα που δεσμεύει μπορεί να πάρει συγκεκριμένες διαστάσεις ή κάποια αναλογία του παραθύρου της ιστοσελίδας [8,9,31]. Συνολικά, στο κώδικα HTML του συστήματος υπάρχουν 7 πλαίσια.

i. **Το πλαίσιο του τίτλου.** Είναι ένας απλός κώδικας HTML ο οποίος εμφανίζει τον τίτλο της παρούσας διπλωματικής με τη μορφή h2, δηλαδή γραμματοσειρά Arial και μέγεθος μεγάλο (Παράρτημα A, σελ. 85).

ii. **Το πλαίσιο ελέγχου.** Στο πλαίσιο αυτό υπάρχουν διασυνδέσεις με άλλες ιστοσελίδες, από τις οποίες πραγματοποιείται ο έλεγχος του κόμβου πρόσβασης, των ασυρμάτων σταθμών διασύνδεσης, της κάμερας. Επίσης υπάρχει και διασύνδεση με το εγχειρίδιο χρήσης του συστήματος. Η ετικέτα που πραγματοποιεί την κάθε διασύνδεση είναι η `` `` και τα περιεχόμενα της εμφανίζονται ως κείμενο που όταν επιλεγεί ανοίγει την ιστοσελίδα που έχει ορισθεί να συνδέσει [8,9,31] (Παράρτημα A, σελ. 86).

iii. **Το πλαίσιο της κάμερα.** Στο πλαίσιο αυτό, εμφανίζεται η εικόνα που λαμβάνεται από την ασύρματη κάμερα IP του συστήματος. Η εμφάνιση του βίντεο, πραγματοποιείται σε ξεχωριστό πλαίσιο μέσα στο ήδη υπάρχων. Το πλαίσιο του βίντεο, δημιουργείται με τη βοήθεια εντολών της γλώσσας Javascript. Η γλώσσα Javascript χρησιμοποιείται για την πραγματοποίηση εργασιών μέσα στον κώδικα της HTML. Έτσι στο συγκεκριμένο πλαίσιο, δηλώνεται ο Web Browser που πρέπει να χρησιμοποιηθεί (Microsoft Internet Explorer), το μέγεθος του πλαισίου του βίντεο, τη διεύθυνση του αρχείου του βίντεο και τα δεδομένα που θα προβληθούν μαζί με το βίντεο (ώρα, ημερομηνία και ψηφιακή μεγέθυνση της εικόνας) [3,8,9,10,31,32] (Παράρτημα A, σελ. 89).

iv. **Το πλαίσιο του διαύλου I²C.** Η λειτουργία του πλαισίου αυτού είναι να στέλνει τα δεδομένα που εισάγει ο χρήστης, στον δίαυλο I²C και να παρουσιάζει τα δεδομένα που έρχονται από το δίαυλο I²C. Δηλαδή, στο πλαίσιο αυτό πραγματοποιείται μια αμφίδρομη επικοινωνία μεταξύ του χρήστη και του συστήματος. Επειδή η γλώσσα HTML όμως, δεν μπορεί να πραγματοποιήσει αμφίδρομη επικοινωνία, συνεργάζεται με τη γλώσσα CGI. Για



τη συνεργασία αυτή, χρησιμοποιείται η κατηγορία ετικετών εισόδου. Η κατηγορία ετικετών αυτή, δημιουργήθηκε για την κάλυψη της ανάγκης για αμφίδρομη επικοινωνία των ιστοσελίδων, που προέκυψε από τη ραγδαία ανάπτυξη του Διαδικτύου. Σε μία ιστοσελίδα, όλες οι ετικέτες εισόδου, περιέχονται στην ετικέτα ελέγχου `<form> </form>`, στην οποία όλα τα δεδομένα που εισάγονται στα περιεχόμενά της, ομαδοποιούνται και καλείται η εφαρμογή που θα τα διαχειριστεί από συγκεκριμένη διεύθυνση [8,9,31,32]. Στο πλαίσιο του διαύλου I²C, οι ετικέτες εισόδου που χρησιμοποιούνται, είναι οι text, radio και submit. Με την ετικέτα εισόδου text ο χρήστης μπορεί να εισάγει δεδομένα προς το δίαυλο I²C και να δέχεται δεδομένα από αυτό. Με την ετικέτα εισόδου radio δημιουργείται στοιχείο επιλογής. Τα στοιχεία επιλογής μπορούν να ομαδοποιηθούν και τότε σε κάθε ομάδα δε μπορεί να επιλεγεί παραπάνω από ένα ταυτόχρονα. Έτσι ο χρήστης έχει τη δυνατότητα επιλογής συγκεκριμένων εργασιών, όπως την ανάγνωση ή την εγγραφή δεδομένων στο δίαυλο I²C. Με την ετικέτα εισόδου submit, δημιουργείται ένα εικονικό κουμπί στη ιστοσελίδα, και όταν ενεργοποιείται καλείται η εφαρμογή που έχει ορισθεί προηγουμένως, για να επεξεργαστεί τα εισαγμένα δεδομένα (Παράρτημα Α, σελ. 90).

ν. **Το πλαίσιο του μετατροπέα ADC.** Η λειτουργία του πλαισίου αυτού, είναι η εμφάνιση των δεδομένων που προκύπτουν από το μετατροπέα ADC του μικροελεγκτή. Η επικοινωνία μεταξύ χρήστη και συστήματος είναι και σε αυτό το πλαίσιο αμφίδρομη. Ο χρήστης εισάγει στο σύστημα τη διεύθυνση του καναλιού που τον ενδιαφέρει και το σύστημα επιστρέφει σε αυτόν, την τιμή που προκύπτει από τη μέτρηση του μετατροπέα στο συγκεκριμένο κανάλι. Η διάρθρωση του κώδικα του πλαισίου στηρίζεται στην ετικέτα `<form> </form>`. Μέσα στην ετικέτα αυτή υπάρχουν οι ετικέτες εισόδου text, option, checkbox και submit. Υπάρχουν δύο ετικέτες text, στη μία εμφανίζονται τα δεδομένα του μετατροπέα και στην άλλη ο χρήστης εισάγει την τιμή offset του κάθε καναλιού. Οι ετικέτες option εισάγονται ως ομάδα και δημιουργούν λίστα επιλογής της διεύθυνσης του καναλιού. Συνολικά, υπάρχουν οκτώ ετικέτες option όσα και τα κανάλια του μετατροπέα ADC. Η ετικέτα checkbox, δείχνει πότε τα δεδομένα που εισάγονται στο πεδίο offset

πρέπει να αποσταλούν στο σύστημα ως η τιμή offset του επιλεγμένου καναλιού. Τέλος η ετικέτα submit όταν επιλεγεί, καλεί την εφαρμογή που θα επεξεργαστεί τα δεδομένα που εισάγονται από το χρήστη (Παράρτημα Α, σελ. 91).

vi. **Το πλαίσιο της θύρας B(GPI/O).** Το πλαίσιο αυτό διενεργεί τον έλεγχο της θύρας B. Και σε αυτήν την περίπτωση, γίνεται χρήση της ετικέτας <form> </form>. Στα περιεχόμενά της, υπάρχει δύο ομάδες ετικετών εισόδου option. Η πρώτη δηλώνει ποια από τα bits της θύρας θα λειτουργούν ως είσοδοι και ποια ως έξοδοι. Η δεύτερη δηλώνει αν θα διαβάσουμε η θα εγγράψουμε δεδομένα. Επίσης υπάρχει μία ετικέτα εισόδου text, στην οποία εισάγονται ή παρουσιάζονται τα δεδομένα της θύρας, από και προς το σύστημα. Τέλος, υπάρχει και στο πλαίσιο αυτό, η ετικέτα εισόδου submit, η οποία καλεί την εφαρμογή που θα επεξεργαστεί τα δεδομένα του πλαισίου (Παράρτημα Α, σελ. 92).

vii. **Το πλαίσιο σκανδαλισμού της κάμερας.** Το πλαίσιο αυτό είναι το μοναδικό που δεν έχει οπτικό αποτέλεσμα στην ιστοσελίδα της παρούσας διπλωματικής εργασίας. Ο σκοπός του είναι απλά να ανανεώνεται ανά 5 δευτερόλεπτα και σε κάθε ανανέωσή του να καλεί ένα αρχείο CGI. Για να επιτευχθεί αυτό γίνεται χρήση της ετικέτας <meta> [3,8,9,10,31]. Στα περιεχόμενά της δηλώνεται η ανανέωση του πλαισίου ανά 5 δευτερόλεπτα και η διαδρομή του αρχείου CGI (Παράρτημα Α, σελ. 95).

3.4 Η γλώσσα προγραμματισμού C και το πρότυπο CGI.

3.4.1 Εισαγωγή

Καθώς η χρήση του διαδικτύου ακολουθούσε ραγδαία εξέλιξη, η ανάγκη για αμφίδρομη επικοινωνία ανάμεσα στις ιστοσελίδες και στο χρήστη ήταν μεγάλη. Για το λόγο αυτό, δημιουργήθηκαν στη γλώσσα υπερκειμένου HTML, οι επικέτες εισόδου, οι οποίες συγκεντρώνουν τα δεδομένα που εισάγει ο χρήστης, και καλούν προκαθορισμένες εφαρμογές οι οποίες τα επεξεργάζονται [2,4,5,6,7,28].

Η γλώσσα προγραμματισμού C, δημιουργεί εφαρμογές οι οποίες λαμβάνουν τα δεδομένα των ιστοσελίδων και προβαίνουν στην επεξεργασία τους. Έτσι δημιουργούνται ιστοσελίδες αμφίδρομης επικοινωνίας, με αποτέλεσμα, ο χρήστης να έχει τη δυνατότητα να μπορεί να παρεμβαίνει στα δεδομένα που του παρουσιάζει ένας Web Browser. Οι εφαρμογές αυτές, βασίζονται στο πρότυπο CGI.

Το πρότυπο CGI, αναπτύχθηκε από την NCSA¹² και από το ευρωπαϊκό εργαστήριο στοιχειωδών σωματιδίων CERN, για την κάλυψη της ανάγκης ενός συνεπούς εξωτερικού προγράμματος διασύνδεσης του Web Browse με τον Server. Χρησιμοποιεί τις εντολές της γλώσσας προγραμματισμού C. Έτσι, λαμβάνει από τον κώδικα της HTML δεδομένα, με τη χρήση του Πρωτοκόλλου μεταφοράς Υπερκειμένου (HyperText Transfer Protocol). Στη συνέχεια, τα επεξεργάζεται και δημιουργεί ή παρεμβαίνει στο περιβάλλον, στο οποίο θα οδηγήσει τα αποτελέσματα της επεξεργασίας. Πρέπει να τονιστεί στο σημείο αυτό, ότι δεν παρουσιάζονται πάντα στον χρήστη απτά αποτελέσματα. Για παράδειγμα, όταν σε μία ιστοσελίδα γίνεται εγγραφή του χρήστη σε μία υπηρεσία, η εφαρμογή CGI θα κληθεί να καταχωρήσει τα στοιχεία του χρήστη σε κάποια τράπεζα δεδομένων χωρίς να επιστρέψει κάποια ένδειξη. Ο τρόπος με τον οποίο το πρότυπο CGI λαμβάνει τα δεδομένα από μία ιστοσελίδα, στηρίζεται σε δύο μεθόδους, την POST και την GET [2,4,32].

Η μέθοδος POST¹, χρησιμοποιείται σε περιπτώσεις όπου τα δεδομένα που εισάγονται από μια ιστοσελίδα, δεν είναι μία απλή ακολουθία χαρακτήρων

¹² National Center for Supercomputing Applications

ASCII. Για παράδειγμα τα δεδομένα από μία ιστοσελίδα αποστολής ηλεκτρονικού ταχυδρομείου, ή τα δεδομένα από μία ιστοσελίδα εγγραφής σε μία βάση δεδομένων. Τα δεδομένα που προκύπτουν από την ιστοσελίδα διαβάζονται από την εφαρμογή CGI ως τυπική είσοδος. Όταν τα δεδομένα πάρουν αυτή τη μορφή, τότε η μεταβλητή REQUEST_METHOD της μεθόδου POST, ενεργοποιείται δείχνοντας την ύπαρξη δεδομένων και η μεταβλητή REQUEST_LENGTH δείχνει το μέγεθος τους [2].

Η μέθοδος GET, χρησιμοποιείται όταν τα δεδομένα που προκύπτουν από την ιστοσελίδα είναι χαρακτήρες ASCII. Οι χαρακτήρες αυτοί, αποθηκεύονται στη μεταβλητή QUERY_STRING και τοποθετούνται σε μία ακολουθία της μορφής keyword1=value1&keyword2=value2&..., η οποία ενσωματώνεται στο URL¹³ της ιστοσελίδας. Όμως δεν διαβάζονται από την εφαρμογή ως τυπική είσοδος, αλλά ως απλή ακολουθία με χρήση της εντολής sscanf [2,4].

3.4.2 Χρήση του προτύπου CGI της γλώσσας C, στο λογισμικό του συστήματος.

Στην παρούσα διπλωματική, τα δεδομένα που εισάγονται στην ιστοσελίδα είναι απλοί αριθμοί, δηλαδή μόνο χαρακτήρες ASCII. Συνεπώς η καταλληλότερη μέθοδος, είναι η GET. Τα δεδομένα που εισάγονται από την ιστοσελίδα, τοποθετούνται σε ακολουθία. Για παράδειγμα τα δεδομένα του διαύλου I²C, παίρνουν τη μορφή a=144&b=24&c=1&d=0&twiButton=Send, όπου a,b,c,d,twibutton είναι οι μεταβλητές της ιστοσελίδας και 144,24,1,0,send οι τιμές τους αντίστοιχα. Η ακολουθία αυτή, ενσωματώνεται στο URL της ιστοσελίδας χρησιμοποιώντας ως διαχωριστικό, τον χαρακτήρα '?'. Η μορφή του είναι:

```
http://IP/cgi-bin/file.cgi?a=144&b=24&c=1&d=0&twiButton=Send
```

Η ανάγνωση των δεδομένων από την εφαρμογή CGI γίνεται με χρήση της εντολής sscanf. Το τμήμα του κώδικα που πραγματοποιεί αυτήν την εργασία, είναι :

```
data = getenv("QUERY_STRING");
```

```
sscanf (data,"a=%ld&b=%ld&c=%ld&d=%ld", &a,&b,&c,&d);
```

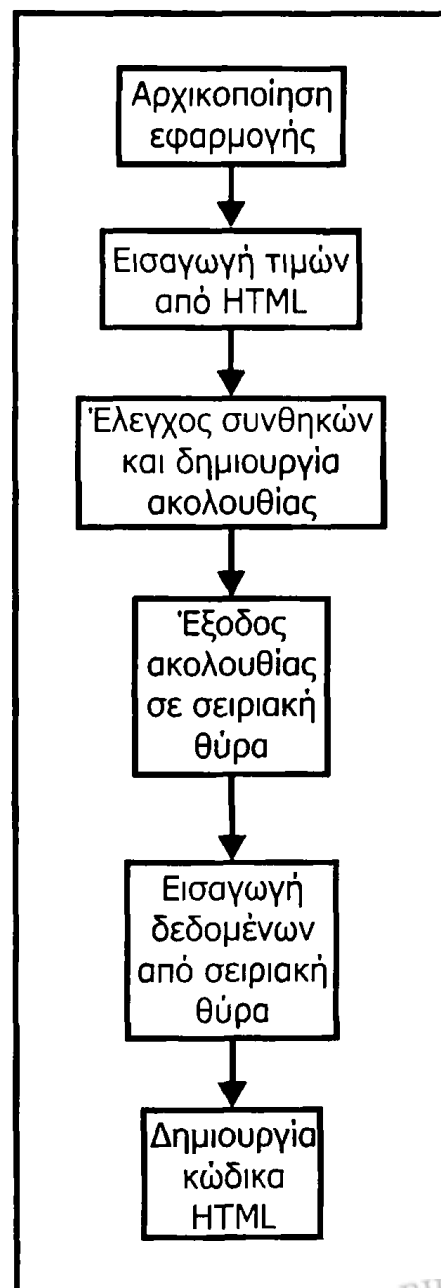
¹³ Uniform Resource Locator



όπου στην πρώτη γραμμή αποθηκεύεται στη μεταβλητή data ολόκληρη η ακολουθία, ενώ στη δεύτερη διαβάζονται και αποθηκεύονται σε ξεχωριστές μεταβλητές, οι επιμέρους τιμές της ακολουθίας.

Εφαρμογές CGI, χρησιμοποιούνται από το πλαίσιο του διαύλου I²C, του μετατροπέα ADC και της θύρα B. Η δομή και των τριών εφαρμογών είναι κοινή και το μόνο που αλλάζει είναι οι μεταβλητές που χρησιμοποιούνται (σχήμα 45). Η επεξεργασία που διενεργούν στα δεδομένα των μεταβλητών είναι η ίδια. Ανάλογα με τις τιμές τους, ικανοποιούν συγκεκριμένες και προκαθορισμένες συνθήκες, οι οποίες δημιουργούν μία ακολουθία χαρακτήρων ASCII, η οποία αντιπροσωπεύει την εντολή του χρήστη προς το σύστημα. Αν δηλαδή πρέπει να αναγνωστεί ο δίαυλος I2C η να γραφεί κάποια τιμή στη θύρα B κλπ. Η ακολουθία αυτή αποστέλλεται στην σειριακή θύρα του server του συστήματος και η εφαρμογή CGI, αναμένει από τη θύρα την απάντηση του συστήματος. Έπειτα, η εφαρμογή CGI δημιουργεί μία ιστοσελίδα, στην οποία παρουσιάζει την απόκριση του συστήματος και η διαδικασία επαναλαμβάνεται σε κάθε κλήση της εφαρμογής.

Ο τρόπος με τον οποίο, η εφαρμογή CGI δημιουργεί ιστοσελίδα, είναι με τη χρήση της εντολής εξόδου της γλώσσας C, printf. Η εντολή εξάγει οτιδήποτε υπάρχει στα περιεχόμενά της. Οπότε, βάζοντας στα περιεχόμενα της printf τον κώδικα της HTML, θα έχουμε ως αποτέλεσμα την εξαγωγή από την εφαρμογή του προτύπου CGI, μιας καινούργιας ιστοσελίδας. Για λόγους ομοιομορφίας, οι ιστοσελίδες που δημιουργούν οι εφαρμογές CGI του



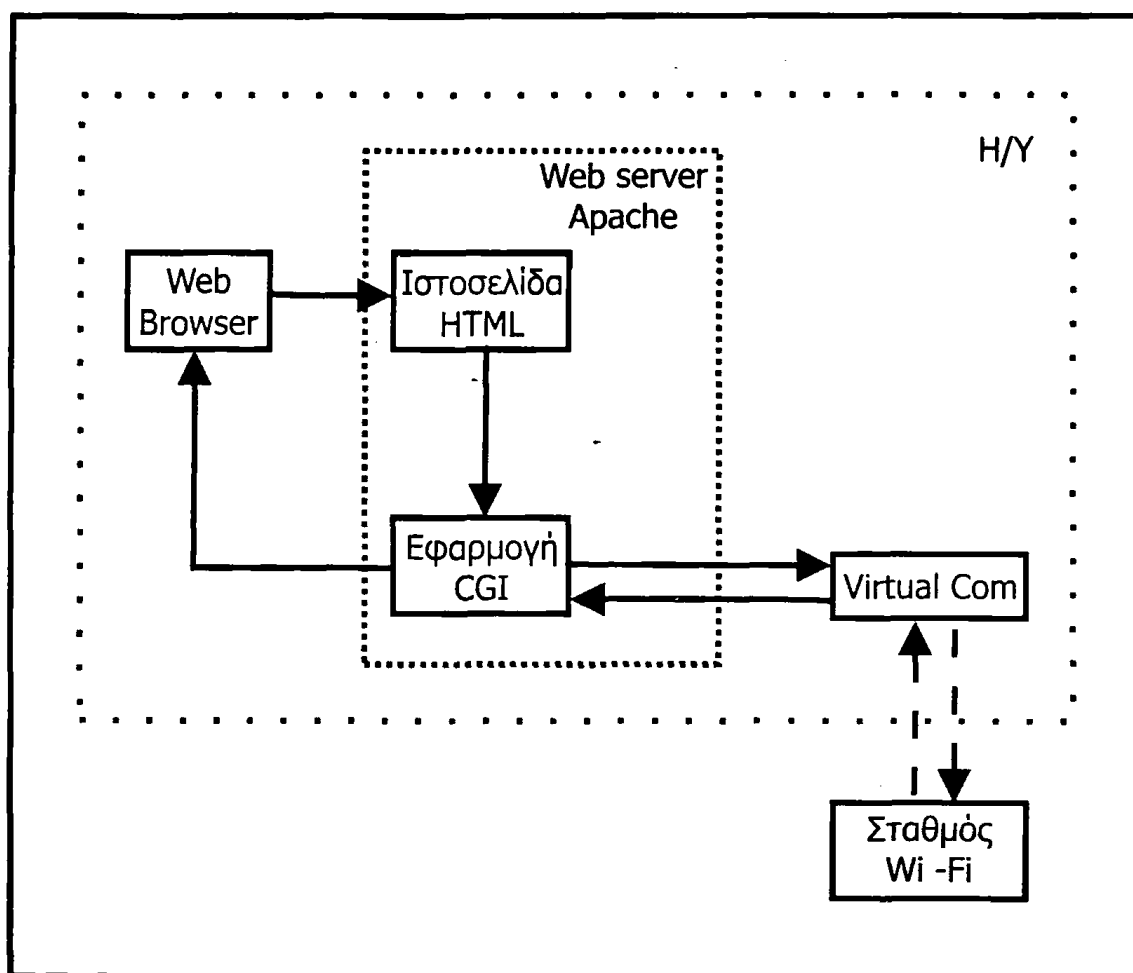
Σχήμα 45: Η εφαρμογή CGI

συστήματος, είναι όμοιες με τις ιστοσελίδες που καλούν τις εφαρμογές αυτές. Έτσι στο χρήστη φαίνεται ότι ανανεώνεται συνεχώς κάθε ιστοσελίδα, προβάλλοντας τα δεδομένα που επιστρέφονται από το σύστημα, ενώ στην πραγματικότητα αναδημιουργείται.



3.5 Ο εξυπηρετητής του συστήματος.

Ένα τμήμα του συστήματος της παρούσας διπλωματικής εργασίας, αποτελείται από ενσύρματο δίκτυο και το υπόλοιπο από ασύρματο δίκτυο. Όπως προαναφέρθηκε, κεντρικό σημείο του ασυρμάτου δικτύου, είναι ο κόμβος πρόσβασης, ο οποίος δημιουργεί και υποστηρίζει το δίκτυο Wi - Fi. Το αντίστοιχο κεντρικό σημείο του ενσύρματου δικτύου του συστήματος, αποτελεί ο Η/Υ του συστήματος. Δηλαδή είναι το σημείο που υποστηρίζει τη λειτουργία του προγράμματος που χρησιμοποιείται ως εξυπηρετητής διαδικτύου (web server) και της εφαρμογής που διασυνδέει την εικονικών θυρών των εφαρμογών CGI με τις σειριακές θύρες των ασυρμάτων σταθμών



Σχήμα 46: Η λειτουργία του Η/Υ.

προσάρτησης Wi - Fi (σχήμα 46).

Οι ελάχιστες απαιτήσεις για τον Η/Υ βάσει των αναγκών των επιμέρους τμημάτων του συστήματος, είναι:

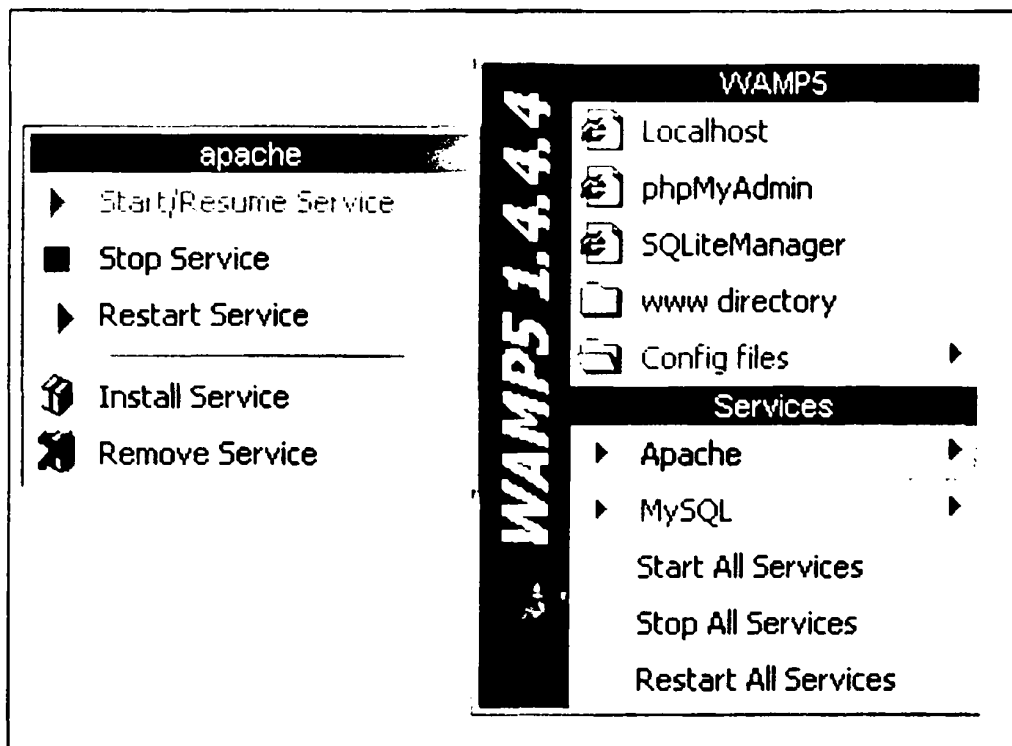
- Ο επεξεργαστής πρέπει να είναι P1 τουλάχιστον.

- Η μνήμη RAM 64 Mbytes ή μεγαλύτερη.
- Κάρτα δικτύου 10/100 Mbps, καθώς και πρόσβαση στο διαδίκτυο (ADSL, cable).
- Λειτουργικό σύστημα Windows 98 ή νεότερη έκδοση.

3.5.1 Ο εξυπηρετητής διαδικτύου.

Ο Η/Υ του συστήματος υποστηρίζει την εφαρμογή που λειτουργεί ως Web Server. Δηλαδή, είναι η εφαρμογή που διαχειρίζεται αρχεία HTML, PHP κλπ, και την πρόσβαση σε αυτά από το διαδίκτυο. Επίσης η εφαρμογή αυτή, διαχειρίζεται και τα αρχεία CGI, τα οποία επεξεργάζονται τα δεδομένα που εισάγονται από τα αρχεία HTML.

Ο Web Server που χρησιμοποιεί το σύστημα, είναι ο Wamp5 (σχήμα 47). Είναι μία εφαρμογή που διανέμεται δωρεάν στο κοινό και αποτελεί έναν σταθερό και υψηλής απόδοσης Web Server. Περιέχει τις εφαρμογές Apache,

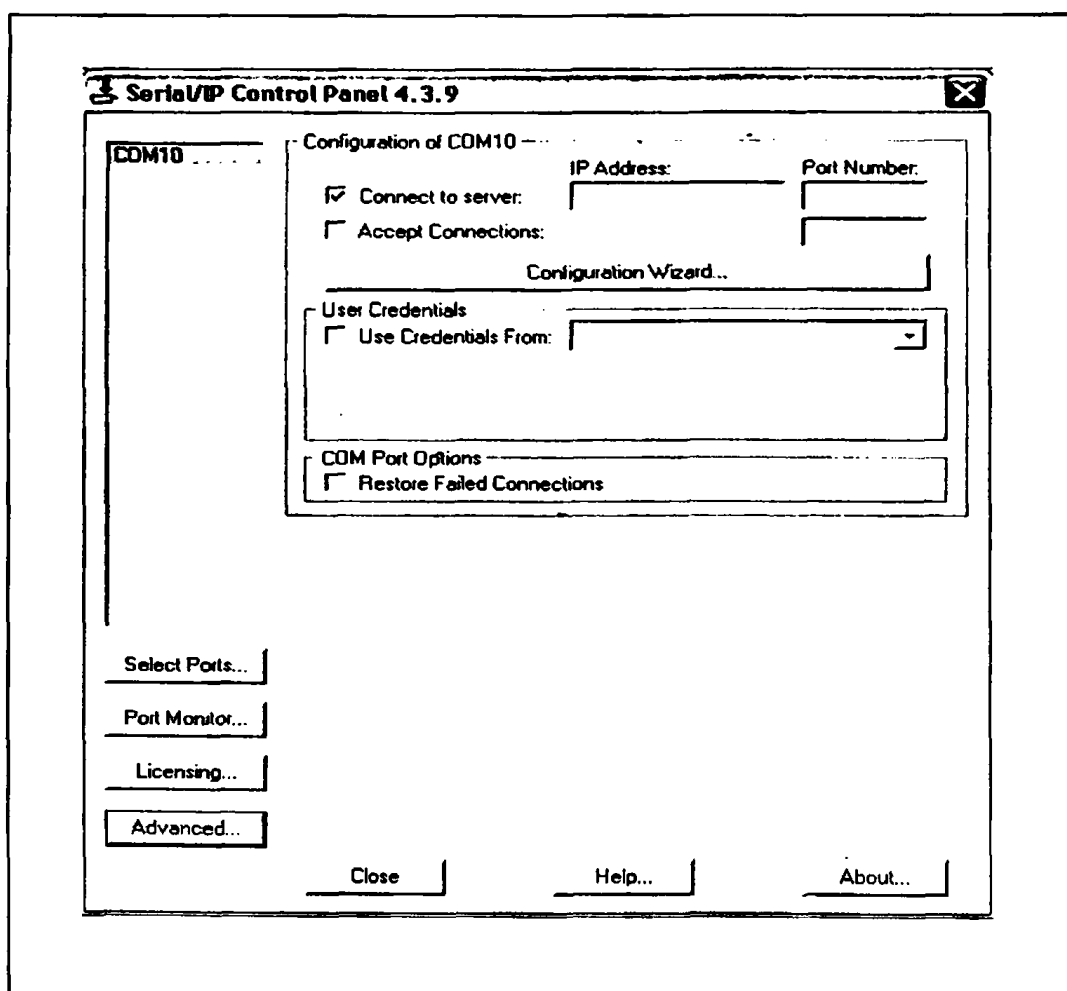


Σχήμα 47: Ο εξυπηρετητής διαδικτύου Wamp.

PHP και MySQL database και κατά τη λειτουργία του ενεργοποιούνται οι επιμέρους Web Servers, Apache και MySQL. Για τις ανάγκες λειτουργίας του συστήματος, χρησιμοποιείται μόνο η υπηρεσία του Web Server Apache, καθώς ο Web Server καλείται να διαχειριστεί μόνο αρχεία HTML.

Ο Apache, για τον έλεγχο της λειτουργίας του χρησιμοποιεί ένα αρχείο τύπου configuration, το HTTPD.conf. Στο αρχείο αυτό, είναι αποθηκευμένες όλες οι οδηγίες για την συνολική λειτουργία του Apache. Δηλαδή, οι ρυθμίσεις του πρωτοκόλλου HTTP, οι θέσεις των αρχείων HTML, καθώς και τα δικαιώματα πρόσβασης σε αυτά.

Για τις ανάγκες λειτουργίας του συστήματος, δημιουργήθηκε ένας φάκελος¹⁴ στον Web Server, μέσα στον οποίο αποθηκεύτηκαν όλα τα αρχεία HTML. Οι ρυθμίσεις για τον φάκελο αυτόν, καθορίζουν την ακριβή θέση του, ελεύθερη πρόσβαση σε οποιοδήποτε αρχείο του, καθώς και την εικονική θύρα



Σχήμα 48: Η εφαρμογή Virtual Com

που θα χρησιμοποιεί το πρωτόκολλο HTTP, που στη συγκεκριμένη εφαρμογή είναι η θύρα 80.

¹⁴ Η πλήρης διεύθυνση του φακέλου είναι: "C:/wamp/www/tests"

Τα αρχεία CGI αποθηκεύονται σε συγκεκριμένο φάκελο, τον CGI-BIN. Οι ρυθμίσεις του φακέλου, είναι επίσης αποθηκευμένες στο αρχείο HTTPD.conf. Ο φάκελος έχει ελεύθερη πρόσβαση, προκαθορισμένη διεύθυνση¹⁵, και επιτρέπεται η ανανέωση των αρχείων του. Κάτι τέτοιο άλλωστε είναι αναγκαίο, αφού τα αρχεία CGI διαμορφώνουν τα περιεχόμενά τους ανάλογα με τα δεδομένα που εισάγουν από τις ιστοσελίδες που τα καλούν. Συνεπώς, θα ανανεώνονται κάθε φορά που θα καλούνται από τις ιστοσελίδες.

3.5.2 Εφαρμογή εικονικής θύρας.

Τα δεδομένα που εισάγονται από τις ιστοσελίδες στις εφαρμογές CGI, πρέπει αφού επεξεργαστούν, να μεταφερθούν στο μικροελεγκτή του σταθμού προσάρτησης. Οι εφαρμογές CGI εξάγουν και εισάγουν δεδομένα μέσω της σειριακής θύρας επικοινωνίας RS-232 του Η/Υ. Ο σταθμός προσάρτησης όμως, είναι συνδεδεμένος με τον ασύρματο σταθμό διασύνδεσης Wi - Fi, επίσης μέσω σειριακής διασύνδεσης RS-232. Οπότε, δημιουργείται ένα κενό διασύνδεσης ανάμεσα στις στη θύρα επικοινωνίας των εφαρμογών CGI και στον ασύρματο σταθμός διασύνδεσης Wi - Fi. Το κενό αυτό καλύπτει η εφαρμογή εικονικής θύρας, Virtual Com.

Η εφαρμογή Virtual Com¹⁶ (σχήμα 48), δημιουργεί στον Η/Υ, μία εικονική θύρα επικοινωνίας. Έπειτα, αντιστοιχεί τη θύρα με μία διεύθυνση δικτύου IP. Έτσι, όταν οποιαδήποτε εφαρμογή του Η/Υ προσπαθήσει να επικοινωνήσει με τη θύρα αυτή, ουσιαστικά επικοινωνεί με την προκαθορισμένη διεύθυνση δικτύου. Η εφαρμογή, μπορεί να δημιουργήσει και να αντιστοιχίσει με διευθύνσεις δικτύου IP έως και 256 εικονικές θύρες επικοινωνίας.

Στη παρούσα διπλωματική, υπάρχουν τρεις ασύρματοι σταθμοί διασύνδεσης Wi - Fi και τρεις ιστοσελίδες που επικοινωνούν με τους σταθμούς, αντίστοιχα. Συνεπώς, θα πρέπει να δημιουργηθούν τρεις εικονικές θύρες στον Η/Υ από την εφαρμογή Virtual Com, οι οποίες θα αντιστοιχιστούν με τις διευθύνσεις δικτύου IP των τριών ασυρμάτων σταθμών διασύνδεσης Wi - Fi του συστήματος. Έτσι, η κάθε εφαρμογή CGI, θα μπορεί να επικοινωνεί

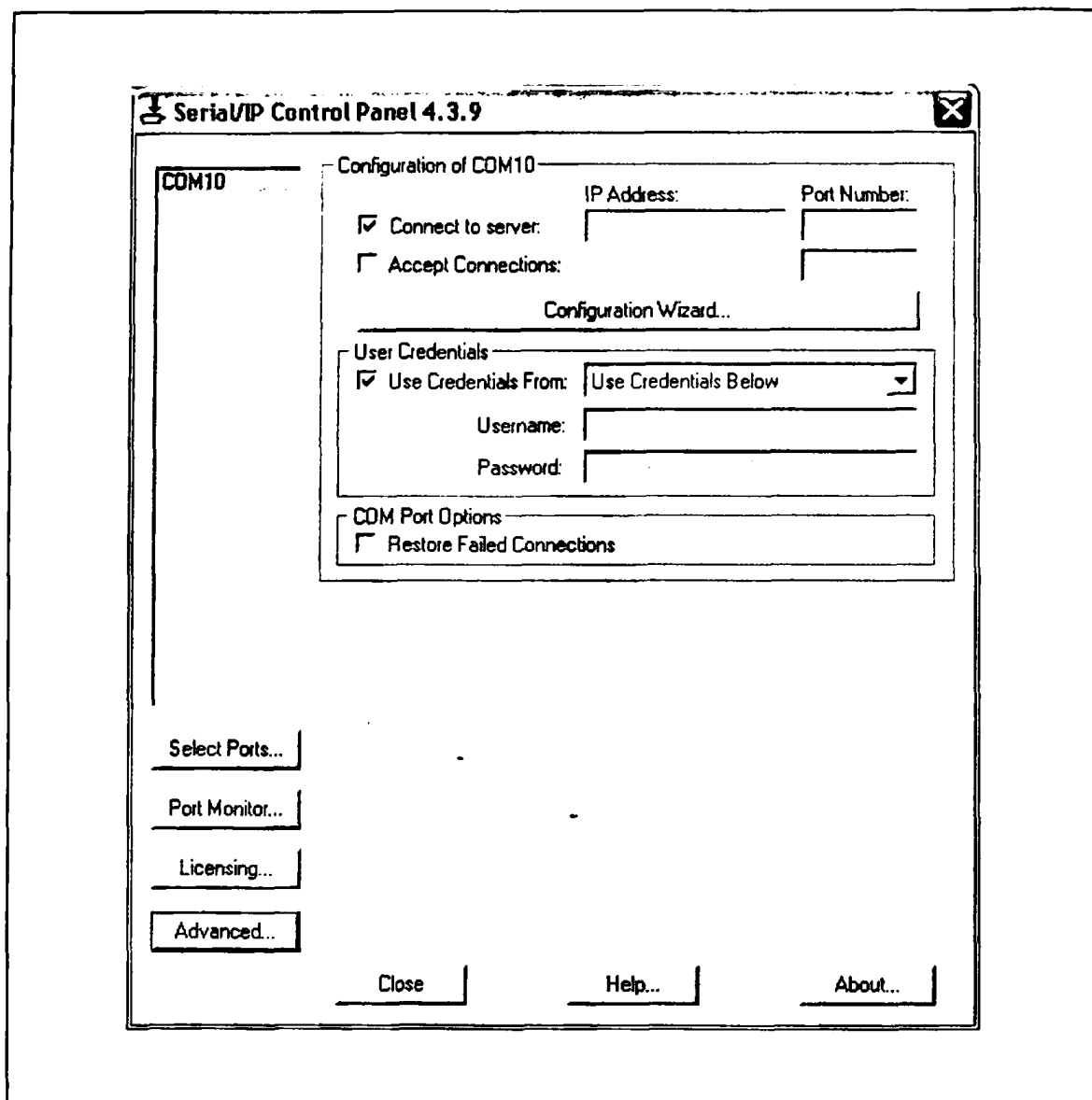
¹⁵ Η πλήρης διεύθυνση του φακέλου είναι: "C:/wamp/apache/cgi-bin"

¹⁶ Η συγκεκριμένη εφαρμογή είναι της εταιρίας ATOP.



με έμμεσο τρόπο, με τον μικροελεγκτή του κάθε σταθμού προσάρτησης, που είναι συνδεδεμένος με τον ασύρματο σταθμό διασύνδεσης Wi – Fi.

Το πρωτόκολλο που χρησιμοποιεί η εφαρμογή Virtual Com για τη μεταφορά των πακέτων δεδομένων, είναι TCP/IP. Επίσης η εφαρμογή δεν



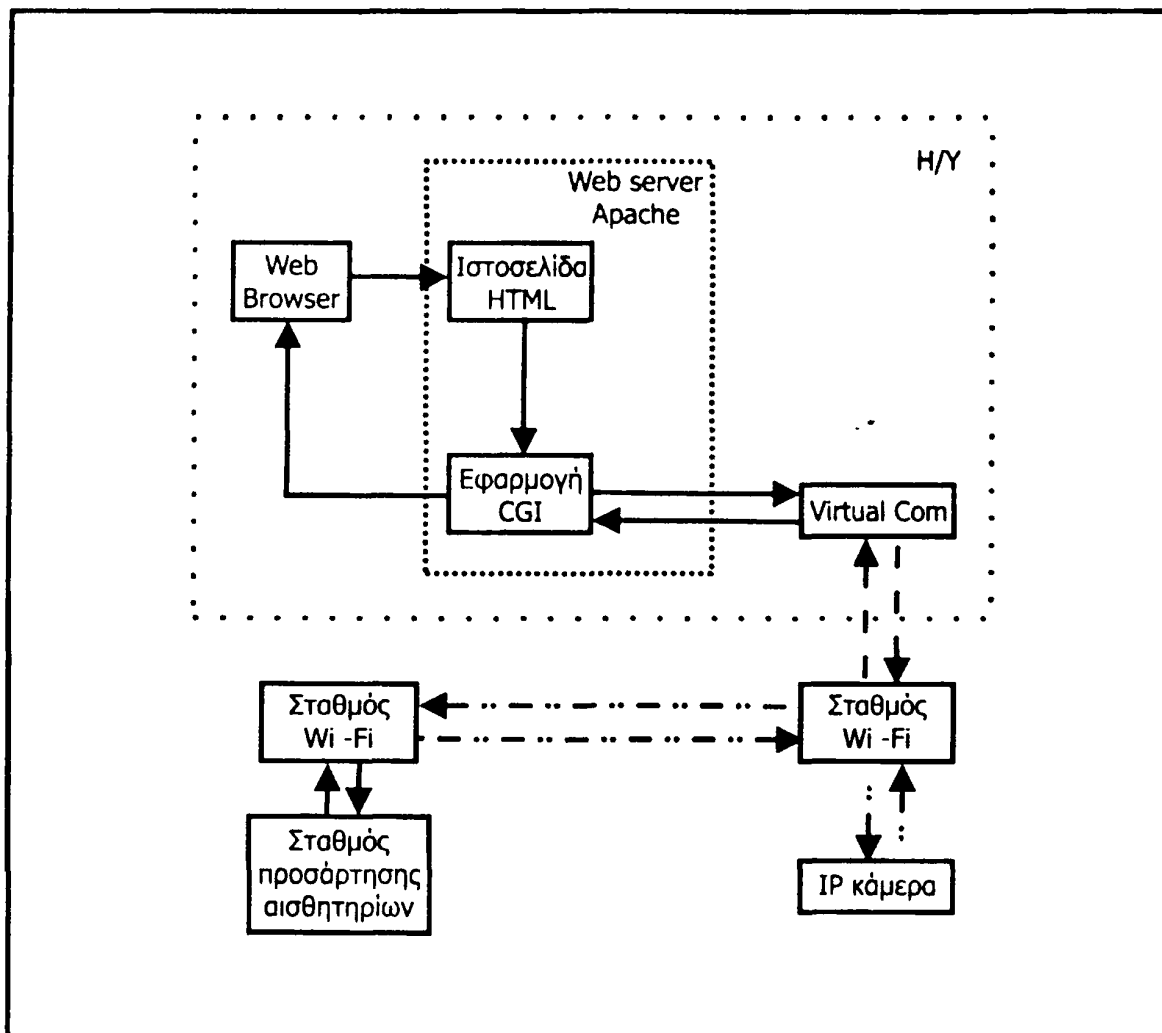
Σχήμα 49: Ο πίνακας ελέγχου της εφαρμογής Virtual Com

προσφέρει κρυπτογράφηση των δεδομένων. Υπάρχει όμως η επιλογή, για ερώτηση ονόματος χρήστη και κωδικό πρόσβασης, πριν την αντιστοίχιση της εικονικής θύρας με την διεύθυνση δικτύου IP (σχήμα 49).

ΚΕΦΑΛΑΙΟ 4^ο.

Λειτουργία και αποτίμηση συστήματος.

Στα προηγούμενα κεφάλαια, αναπτύχθηκε το σύστημα της παρούσας διπλωματικής εργασίας τόσο ως προς τα λογισμικά του τμήματα, όσο και ως προς τα υλικά του τμήματα. Στο παρόν κεφάλαιο παρουσιάζεται η λειτουργία των αντίστοιχων τμημάτων. Δηλαδή, η επιμέρους λειτουργία του λογισμικού

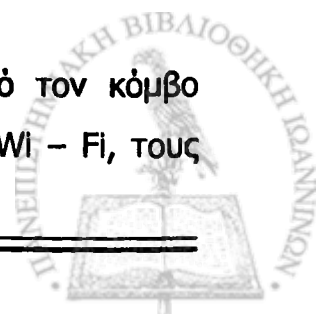


Σχήμα 50: Η λειτουργία του συστήματος

της πλήρους εφαρμογής και του υλικού ολόκληρου του συστήματος. Επιπλέον, θα παρουσιαστεί η αποτίμηση του ολοκληρωμένου συστήματος .

4.1 Η λειτουργία των τμημάτων του υλικού.

Το σύστημα της παρούσας διπλωματικής, αποτελείται από τον κόμβο πρόσβασης, τον Η/Υ, τους σταθμούς ασύρματης διασύνδεσης Wi - Fi, τους



σταθμούς προσάρτησης αισθητηρίων καθώς και μία κάμερα IP. Ο κόμβος πρόσβασης συνδέεται καλωδιακά με τον Η/Υ και με την πρόσβαση στο διαδίκτυο. Συνδέεται όμως ασύρματα, με τους σταθμούς ασύρματης διασύνδεσης Wi – Fi και τη κάμερα IP. Τέλος, οι σταθμοί ασύρματης διασύνδεσης Wi – Fi, συνδέονται ενσύρματα με τους σταθμούς προσάρτησης αισθητηρίων.

Όταν ο χρήστης ζητήσει να διαβάσει τα δεδομένα ενός αισθητηρίου, τότε η πληροφορία ακολουθεί συγκεκριμένη πορεία. Από τον Η/Υ η εντολή μεταφέρεται στον κόμβο πρόσβασης ο οποίος την μεταβιβάζει ασύρματα στον αντίστοιχο σταθμό ασύρματης διασύνδεσης Wi – Fi. Από εκεί, διαβιβάζεται ενσύρματα στον σταθμό προσάρτησης αισθητηρίων και αφού ο μικροελεγκτής εκτελέσει την εργασία που του έχει ζητηθεί, επιστρέφει το αποτέλεσμα ακολουθώντας την αντίστροφη πορεία.

4.1.1 Ο κόμβος πρόσβασης

Ο κόμβος πρόσβασης, όπως έχει προαναφερθεί αποτελεί το κεντρικό σημείο του ασυρμάτου δικτύου. Λειτουργεί αυτόνομα, χωρίς δηλαδή την ανάγκη υποστήριξης από Η/Υ. Μόλις τροφοδοτηθεί, αναζητά τους σταθμούς που βρίσκονται μέσα στο χώρο κάλυψης. Αφού πραγματοποιηθεί η ενέργεια αυτή, το ασύρματο δίκτυο είναι σε λειτουργία και πλέον τα πακέτα δεδομένων μπορούν να μεταβιβαστούν. Οι περισσότεροι σταθμοί που μπορεί να δεχτεί είναι μεταξύ 30 και 70, ανάλογα με την κίνηση πακέτων δεδομένων του δικτύου [14,19,26]. Η διαμόρφωση που χρησιμοποιείται είναι DSSS για το πρότυπο 802.11b και OFDM για το πρότυπο 802.11g.

Η σχετική υγρασία και θερμοκρασία λειτουργίας είναι 0% - 90% και 0°C – 40°C αντίστοιχα, καθιστώντας τη συσκευή κατάλληλη μόνο για εσωτερική χρήση. Η τροφοδοσία του είναι 12VDC/1A ενώ καταναλώνει 22W [19].

4.1.2 Ο σταθμός ασύρματης διασύνδεσης

Ο σταθμός ασύρματης διασύνδεσης, αποτελεί το σταθμό (node) του ασύρματου δικτύου. Λειτουργεί αυτόνομα, χωρίς δηλαδή την υποστήριξη Η/Υ.

αφού τροφοδοτηθεί, προσπαθεί να εντοπίσει τον κόμβο πρόσβασης και να συνδεθεί με αυτόν. Αν η σύνδεση επιτευχθεί τότε παράγεται ένα ηχητικό σήμα επιβεβαίωσης.

Η σχετική υγρασία και τροφοδοσία λειτουργίας είναι 20% - 90% και 0°C – 50°C αντίστοιχα. Συνεπώς η συσκευή ενδείκνυται μόνο για εσωτερική χρήση. Η τροφοδοσία λειτουργίας είναι 13VDC με κατανάλωση ρεύματος 350mA [17].

4.1.3 Η κάμερα IP

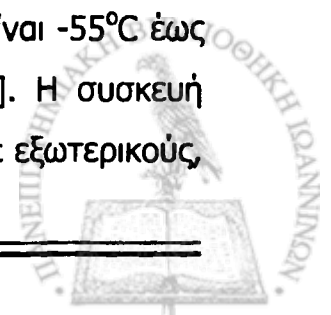
Η κάμερα IP, αποτελεί ακόμη ένα αυτόνομο τμήμα του συστήματος. Μόλις τροφοδοτηθεί, ανιχνεύει και συνδέεται με τον κόμβο πρόσβασης που είναι διαθέσιμος. Η σύνδεση επαληθεύεται με μπλε φωτεινή ένδειξη. Αφού συνδεθεί, ξεκινά τη λήψη εικόνας διαμορφώνοντας το ρυθμό των πλαισίων ανά δευτερόλεπτο, ανάλογα με την ισχύ του σήματος του κόμβου πρόσβασης και τον αριθμό των χρηστών που είναι συνδεδεμένη με αυτήν.

Η κάμερα μπορεί να λειτουργήσει με σχετική υγρασία και θερμοκρασία χώρου 0% - 95% και 0°C – 50°C αντίστοιχα. Συνεπώς κρίνεται ακατάλληλη για εξωτερική χρήση [18].

4.1.4 Ο σταθμός διασύνδεσης αισθητηρίων

Ο σταθμός διασύνδεσης αισθητηρίων, αποτελεί και την κατασκευή της παρούσας διπλωματικής εργασίας. Λειτουργεί όπως και οι προηγούμενες συσκευές αυτόνομα. Μόλις τροφοδοτηθεί, ο μικροελεγκτής της συσκευής μπαίνει σε λειτουργία και αναμένει εντολή από το χρήστη. Υπάρχουν δυο φωτεινές ενδείξεις. Μία κόκκινη που υποδηλώνει την ύπαρξη τροφοδοσίας και μία πράσινη που αναβοσβήνει κάθε φορά που καταφθάνουν δεδομένα μέσω της σειριακής επικοινωνίας στη συσκευή.

Η τροφοδοσία της συσκευής είναι 13VDC 130mA και της τα προσφέρει ο σταθμός ασύρματης διασύνδεσης. Η θερμοκρασία λειτουργίας είναι -55°C έως 125°C ενώ η σχετική υγρασία λειτουργίας 0% έως 90% [15]. Η συσκευή μπορεί να λειτουργήσει τόσο σε εσωτερικούς χώρους όσο και σε εξωτερικούς,



αλλά επειδή δεν είναι υδατοστεγή, πρέπει πάντα να είναι προστατευμένη από το νερό.

4.1.4 Ο διακομιστής του συστήματος

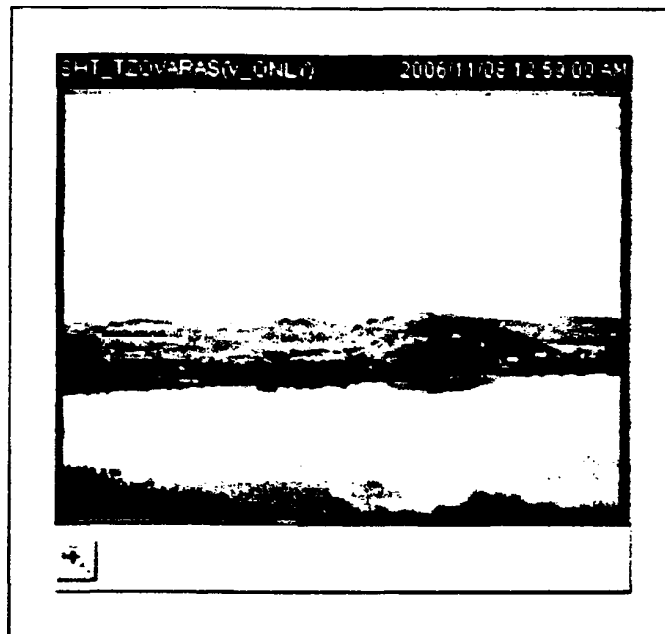
Ο διακομιστής του συστήματος αποτελεί το κεντρικό σημείο του ενσύρματου δικτύου και αποτελείται από ένα Η/Υ. Κατά τη διάρκεια της λειτουργίας του υποστηρίζει τη λειτουργία των εφαρμογών HTML και CGI καθώς και των εφαρμογών VirtualCom και Wamp, που αναλύονται παρακάτω. Σε περίπτωση επανεκκίνησης του Η/Υ, οι παραπάνω εφαρμογές επαναλειτουργούν αυτόματα εξασφαλίζοντας συνεχή και αδιάκοπη λειτουργία του συστήματος.

4.2 Η λειτουργία των τμημάτων του λογισμικού.

Όπως έχει ήδη αναλυθεί στο 3^ο κεφάλαιο, το λογισμικό του συστήματος αποτελείται από τα επιμέρους τμήματα κώδικα HTML, CGI και Assembly. Η λειτουργία τους, ακολουθεί τη σειρά HTML → CGI → Assembly → CGI → HTML.

4.2.1 Ο κώδικας HTML

Το τμήμα του λογισμικού με το οποίο ο χρήστης έχει άμεση επαφή, είναι οι εφαρμογές HTML. Οποιαδήποτε λειτουργία του συστήματος ενεργοποιείται από τις εφαρμογές αυτές, μετά από παρέμβαση του χρήστη. Η πρόσβαση στις εφαρμογές, γίνεται απλά, μέσω του Web Browser Internet Explorer¹⁷, κάτι που αποτελεί και το μεγάλο πλεονέκτημα του



Σχήμα 51: Το πλαίσιο της κάμερας

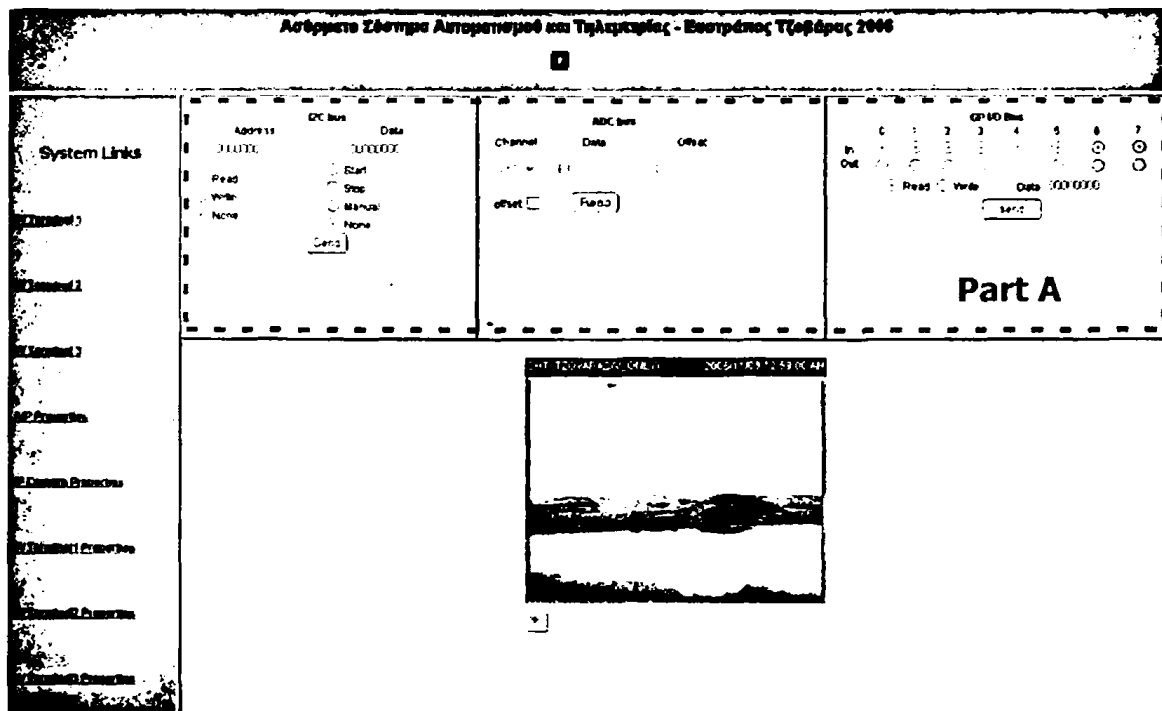
συστήματος. Ο χρήστης δηλαδή, θέτει σε λειτουργία το σύστημα από οποιοδήποτε σημείο του πλανήτη που έχει πρόσβαση στο Διαδίκτυο. Έτσι, μπορεί να αναγνώσει από οπουδήποτε τις τιμές των αισθητηρίων του συστήματος, να ενεργοποιήσει τις συσκευές που είναι συνδεδεμένες είτε στο δίαυλο I²C είτε στη θύρα GPI/O, καθώς και να έχει οπτική επιτήρηση του χώρου που καλύπτει η κάμερα. Επιπλέον το σύστημα αυτόματα ανά 5 δευτερόλεπτα, προσφέρει τον έλεγχο συγκεκριμένου pin του μικροελεγκτή, και αν αυτό είναι ενεργοποιημένο, σκανδαλίζει την κάμερα και εμφανίζει στον Web Browser καινούργιο παράθυρο με την εικόνα της κάμερας εκείνης της στιγμής. Τρία είναι τα πλαίσια του κώδικα HTML θα

¹⁷ Internet Explorer της εταιρίας Microsoft

ενεργοποιηθούν αρχικά, μόλις ο χρήστης συνδεθεί με την ιστοσελίδα του συστήματος. Το πλαίσιο των διασυνδέσεων, το πλαίσιο της κάμερας και το πλαίσιο του σκανδαλισμού της κάμερας.

Το πλαίσιο της κάμερας λειτουργεί αυτόματα, χωρίς να χρειαστεί η παρέμβαση από το χρήστη (σχήμα 51). Σε χρόνο που είναι ανάλογος της σύνδεσης με το διαδίκτυο που χρησιμοποιείται, το πλαίσιο συνδέεται με την κάμερα και προβάλλεται ζωντανά και συνεχόμενα η εικόνα του χώρου επιτήρησης. Στο πάνω μέρος της εικόνας, αναγράφονται η ημερομηνία το όνομα του δικτύου και η ώρα. Επίσης στο πλαίσιο της κάμερας, ο χρήστης έχει τη δυνατότητα να μεγεθύνει την εικόνα και να αλλάξει το πεδίο εστίασης.

Το πλαίσιο του σκανδαλισμού της κάμερας, λειτουργεί αυτόματα και χωρίς την δυνατότητα παρέμβασης από το χρήστη, ανανεώνοντας το περιεχόμενό



Σχήμα 52: Το τμήμα A της ιστοσελίδας του συστήματος

του σε τακτά χρονικά διαστήματα. Αν ανιχνευτεί σκανδαλισμός της κάμερας, τότε εμφανίζεται σε καινούργιο παράθυρο το καρτέ της στιγμής του σκανδαλισμού της κάμερας. Σε αντίθετη περίπτωση, το πλαίσιο απλά θα ανανεώνεται.

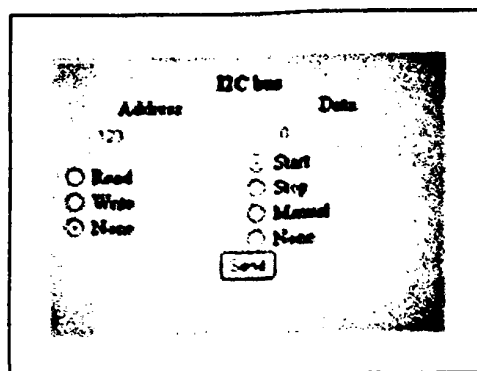
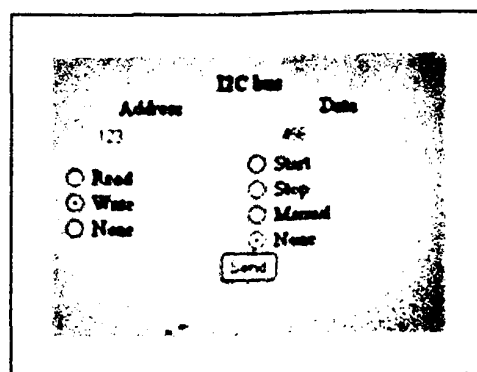
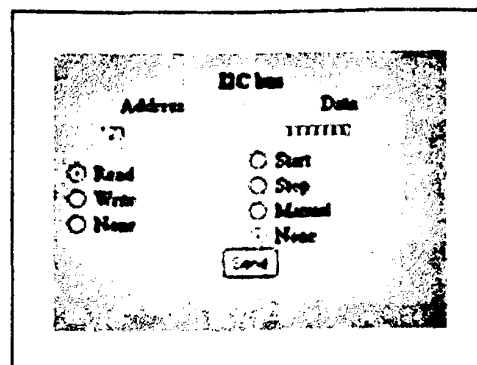
Το πλαίσιο των διασυνδέσεων, δίνει στο χρήστη τη δυνατότητα να επιλέξει με ποιο τμήμα του συστήματος θέλει να συνδεθεί. Μπορεί να συνδεθεί με το τμήμα A (Part A) ή με τις ιστοσελίδες ελέγχου των ρυθμίσεων του κόμβου

πρόσβασης, των ασυρμάτων σταθμών διασύνδεσης Wi – Fi και της κάμερας του συστήματος. Τέλος, μπορεί να επιλέξει να συνδεθεί με τον εγχειρίδιο χρήσης (user manual) του συστήματος, για να τον συμβουλευτεί σχετικά με τη λειτουργία του. Η πρόσβαση στις ιστοσελίδες ελέγχου των ρυθμίσεων του κόμβου πρόσβασης, της κάμερας και των σταθμών ασύρματης διασύνδεσης Wi – Fi, είναι εφικτή μόνο από το εσωτερικό δίκτυο του συστήματος και όχι από το διαδίκτυο, για λόγους ασφάλειας, και προστατεύονται με κωδικό.

Το τμήμα A περιλαμβάνει τα πλαίσια I²C, ADC και GPI/O ενός εκ των τριών ασυρμάτων σταθμών διασύνδεσης Wi – Fi (σχήμα 52). Η επιλογή του τμήματος αυτού, γίνεται από τα W Terminal 1, W Terminal 2 και W Terminal 3. Οι διασυνδέσεις αυτές οδηγούν τα πλαίσια I²C, ADC και GPI/O και με χρήση εντολών Javascript, εμφανίζουν σε αυτά τις αντίστοιχες εφαρμογές HTML. Στη συνέχεια τα πλαίσια καλούν τις αντίστοιχες εφαρμογές CGI, οι οποίες διαχειρίζονται τα δεδομένα που προκύπτουν.

Στο πλαίσιο του διαύλου I²C, ο χρήστης έχει πέντε επιλογές:

- Ανάγνωση του διαύλου (read)
- Εγγραφή στο δίαυλο (write)
- Αποστολή κατάστασης εκκίνησης (start)
- Αποστολή κατάστασης τερματισμού (stop)
- Μη αυτόματη ανάγνωση ή εγγραφή του διαύλου (manual)

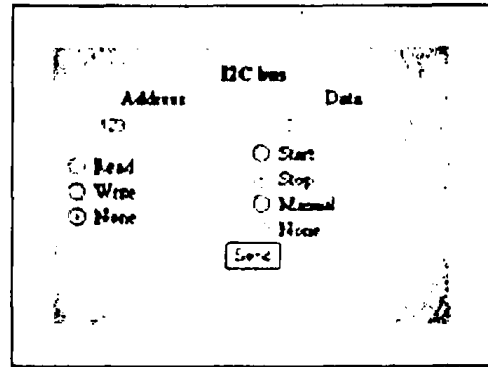


Σχήμα 53: Η λειτουργία του διαύλου I²C



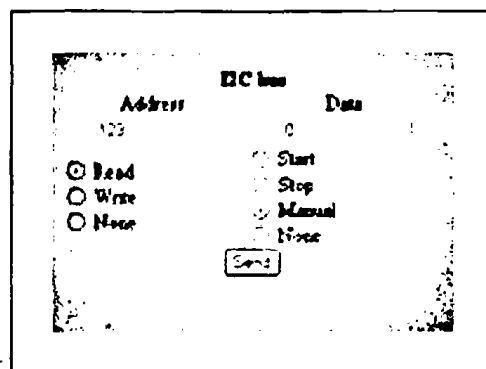
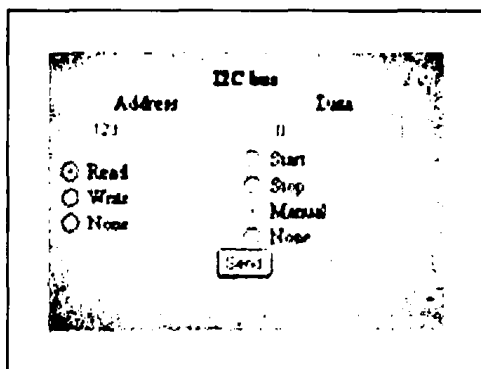
4.2.1.1 Πλαίσιο I²C

Στην περίπτωση ανάγνωσης, ο χρήστης πρέπει να εισάγει τη διεύθυνση της συσκευής I²C που επιθυμεί, να επιλέξει την εντολή read από την αριστερή στήλη επιλογών και none από τη δεξιά στήλη επιλογών (σχήμα 53,54,55). Έτσι η ετικέτα εισόδου address παίρνει την τιμή της διεύθυνσης, η ετικέτα εισόδου read την τιμή '1' και η ετικέτα εισόδου none την τιμή '0'.



Σχήμα 54: Η λειτουργία του διαύλου I²C

Στην περίπτωση εγγραφής, ο χρήστης πρέπει να εισάγει τη διεύθυνση της συσκευής που επιθυμεί, τα δεδομένα που θέλει να εγγράψει στην αντίστοιχη



Σχήμα 55: Η λειτουργία του διαύλου I²C

διεύθυνση, να επιλέξει στην εντολή write από την αριστερή στήλη επιλογών και none από τη δεξιά στήλη επιλογών. Έτσι, η ετικέτα εισόδου address παίρνει την τιμή της διεύθυνσης, η ετικέτα εισόδου data την τιμή των δεδομένων προς εγγραφή, η ετικέτα εισόδου write την τιμή '0' και η ετικέτα εισόδου none την τιμή '0'.

Στην περίπτωση αποστολής κατάστασης εκκίνησης, ο χρήστης πρέπει να εισάγει την διεύθυνση της συσκευής που επιθυμεί, να επιλέξει την εντολή none από τη δεξιά στήλη επιλογών και την επιλογή start και από την αριστερή στήλη επιλογών. Έτσι, η ετικέτα εισόδου παίρνει την τιμή της διεύθυνσης, η ετικέτα εισόδου none την τιμή '0' και η ετικέτα εισόδου start την τιμή '2'.

Στην περίπτωση αποστολής κατάστασης τερματισμού, ο χρήστης πρέπει να επιλέξει την εντολή none από τη δεξιά στήλη επιλογών και την εντολή stop

από την αριστερή στήλη επιλογών. Έτσι, η ετικέτα εισόδου none παίρνει την τιμή '0' και η ετικέτα εισόδου stop την τιμή '3'.

Τέλος, στην περίπτωση αποστολής μη αυτόματης εγγραφής ή ανάγνωσης, ο χρήστης πρέπει να εισάγει τη διεύθυνση της συσκευής που επιθυμεί, να επιλέξει την εντολή write ή read από την δεξιά στήλη επιλογών και την εντολή manual από την αριστερή στήλη επιλογών. Έτσι, η ετικέτα εισόδου address παίρνει την τιμή της διεύθυνσης, η ετικέτα εισόδου write ή read την τιμή '0' ή '1' αντίστοιχα και η ετικέτα εισόδου manual, την τιμή '4'. Μετά από κάθε συνδυασμό, ο χρήστης επιλέγει το πλήκτρο send και η εφαρμογή καλεί το αντίστοιχο αρχείο CGI.

4.2.1.2 Πλαίσιο ADC

Στο πλαίσιο του μετατροπέα ADC, ο χρήστης έχει δύο επιλογές:

- Ανάγνωση της τιμής ενός καναλιού
- Εγγραφή στην μνήμη EEPROM της τιμής offset ενός καναλιού.

Για να διαβάσει ο χρήστης την τιμή του μετατροπέα, η διαδικασία είναι απλή (σχήμα 56). Απλά επιλέγει το κανάλι που επιθυμεί και δεν σημειώνει την επιλογή offset. Έτσι, η ετικέτα εισόδου channel παίρνει την τιμή της διεύθυνσης του καναλιού και η ετικέτα εισόδου offset την τιμή '0'. Για να εγγράψει την τιμή offset ενός καναλιού, εισάγει την αντίστοιχη τιμή, επιλέγει το κανάλι που επιθυμεί και σημειώνει την επιλογή offset. Έτσι, η ετικέτα εισόδου channel παίρνει την τιμή της διεύθυνσης του καναλιού, η ετικέτα εισόδου data την τιμή offset και η ετικέτα εισόδου offset την τιμή '1'. Μετά από κάθε περίπτωση, ο χρήστης επιλέγει το πλήκτρο read και η εφαρμογή καλεί το αντίστοιχο αρχείο CGI.

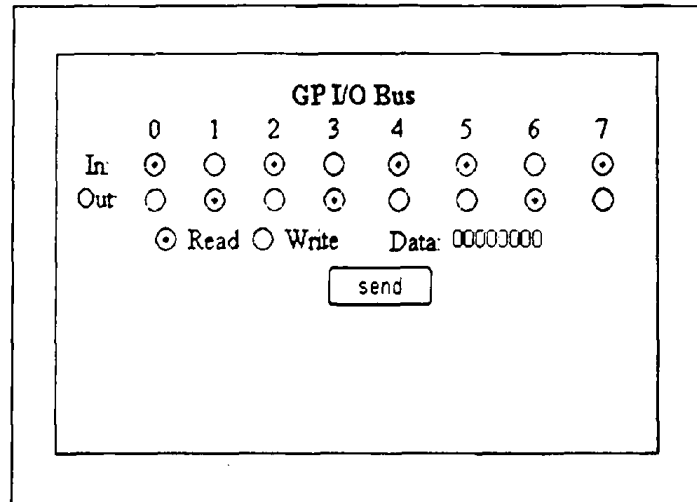
Σχήμα 56: Το πλαίσιο του μετατροπέα ADC

4.2.1.3 Πλαίσιο GPI/O

Στο πλαίσιο της θύρας GPI/O, ο χρήστης έχει επίσης δύο επιλογής:

- Ανάγνωση των bits της θύρας
- Εγγραφή των bits της θύρας

Σε κάθε μία από τις παραπάνω περιπτώσεις, ο χρήστης πρέπει πρώτα να επιλέξει ποια bits της θύρας θα είναι εισοδοί και ποια έξοδοι (σχήμα 57). Για να γίνει αυτό, επιλέγει στις 8 ομάδες ετικετών εισόδου in out την αντίστοιχη. Για εισοδο η αντίστοιχη τιμή είναι '0' και για έξοδο η τιμή είναι '1'. Για



Σχήμα 57: Το πλαίσιο της θύρας GPI/O

εγγραφή στη θύρα ο χρήστης εισάγει τα δεδομένα και επιλέγει write. Έτσι, η ετικέτα εισόδου write παίρνει την τιμή '1'. Για ανάγνωση, επιλέγει read και η ετικέτα εισόδου read παίρνει την τιμή '0'. Πρέπει να σημειωθεί πως αν ένα bit της θύρας είναι δηλωμένο σαν εισόδος και ο χρήστης προσπαθήσει να εγγράψει δεδομένα σε αυτό, η ενέργεια θα αποτύχει και η τιμή του αντίστοιχου bit της πόρτας δεν θα μεταβληθεί. Μετά από κάθε περίπτωση, ο χρήστης επιλέγει send και η εφαρμογή καλεί το αντίστοιχο αρχείο CGI.

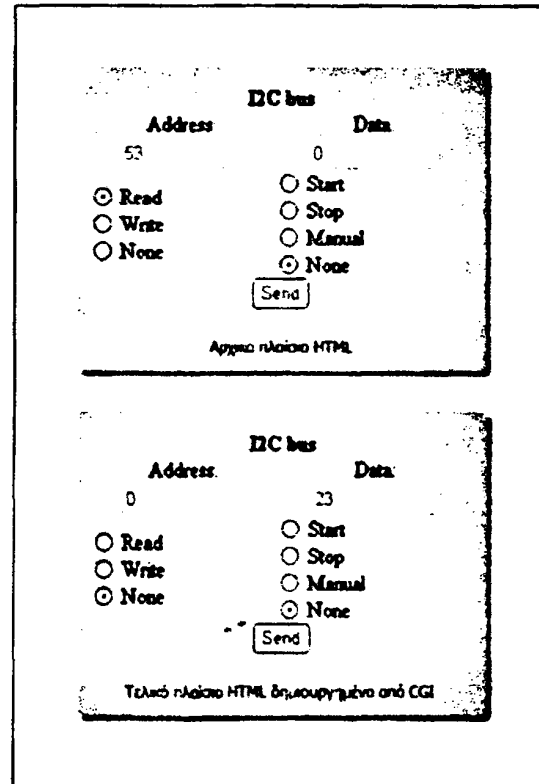
4.2.1.4 Πλαίσιο σκανδαλισμού της κάμερας

Στο πλαίσιο αυτό, ο χρήστης δεν μπορεί να παρεμβληθεί της λειτουργίας. Το πλαίσιο αυτόματα, εκτελεί συγκεκριμένη λειτουργία η οποία επαναλαμβάνεται σε προκαθορισμένα χρονικά διαστήματα των 5 δευτερολέπτων.

4.2.2 Ο κώδικας του προτύπου CGI

Ο κώδικας HTML των πλαισίων I²C, ADC και GPI/O από μόνος του δεν μπορεί να πραγματοποιήσει καμία επεξεργασία δεδομένων. Την επεξεργασία αυτή την αναλαμβάνουν οι εφαρμογές του προτύπου CGI. Δηλαδή πίσω από τα πλαίσια I²C, ADC και GPI/O λειτουργεί εφόσον κληθεί, το αντίστοιχο αρχείο CGI.

Το αρχείο CGI του διαύλου I²C, όταν κληθεί από το αντίστοιχο αρχείο HTML, εισάγει τις τιμές των αντίστοιχων ετικετών εισόδου. Έπειτα, διενεργεί δύο ελέγχους. Αν έχει επιλεγεί ένα από τα read – write ή ένα από τα start – stop – manual. Ο έλεγχος αυτός διενεργείται βάσει των τιμών των αντίστοιχων ετικετών εισόδου. Στη συνέχεια, ανάλογα με την συνθήκη που ικανοποιείται, τα δεδομένα αποκτούν συγκεκριμένη μορφή ακολουθίας. Πρέπει να σημειωθεί ότι πάντα προηγείται των δεδομένων ο χαρακτήρας 'w' και μετά από κάθε τιμή των ετικετών εισόδου, η εφαρμογή CGI παρεμβάλλει τον χαρακτήρα '#' ώστε να ξεχωρίζουν οι τιμές μεταξύ τους. Αναλυτικότερα σε κάθε περίπτωση θα έχουμε το εξής αποτέλεσμα:



Σχήμα 58: Τα πλαίσια του διαύλου I²C

- Read:
w"address"#"read"#"
- Write:
w"address"#"write"#"data"#"
- Start:
w"address"#"start"#"
- Stop:
w"stop"#"
- Manual read:
w"manual"#"read"#"
- Manual write:
w"manual"#"write"#"data"#"

Για παράδειγμα, στην περίπτωση της εγγραφής της τιμής '123' στη συσκευή με διεύθυνση '12', η ακολουθία των δεδομένων θα έχει τη μορφή:
w#12#0#123#



Μετά τη δημιουργία της ακολουθίας των δεδομένων, η εφαρμογή CGI αρχικοποιεί την σειριακή θύρα επικοινωνίας του Η/Υ και στέλνει στον καταχωρητή δεδομένων της θύρας την ακολουθία. Πρέπει να σημειωθεί ότι η ακολουθία αποστέλλεται σε ASCII κωδικοποίηση. Τα δεδομένα λαμβάνονται από τον μικροελεγκτή και αφού πραγματοποιήσει την αντίστοιχη ενέργεια επιστρέφει στην ίδια θύρα επικοινωνίας το αποτέλεσμα. Η εφαρμογή CGI, λαμβάνει τα δεδομένα, δημιουργεί το αντίστοιχο αρχείο HTML και τα τοποθετεί στην ετικέτα εισόδου data. Έτσι, στον χρήστη το αποτέλεσμα θα είναι απλά η ανανέωση της τιμής της ετικέτας εισόδου data αφού εμφανισιακά δεν αλλάζει τίποτα άλλο (σχήμα 58).

Το αρχείο CGI του μετατροπέα ADC, λειτουργεί με παρόμοιο τρόπο. Αφού κληθεί, εισάγει τις τιμές των ετικετών εισόδου και ελέγχει την τιμή της επιλογής offset. Τα δεδομένα που προκύπτουν τοποθετούνται σε ακολουθίες με το χαρακτήρα 'd' πάντα στην αρχή και τον χαρακτήρα '#' μετά από την τιμή της κάθε ετικέτας εισόδου. Οι περιπτώσεις που μπορεί να προκύψουν είναι:

- Read:
d"channel"#"offset"#"
- Offset:
d"channel"#"offset"#"data"#"

Για παράδειγμα, στην περίπτωση εγγραφής της τιμής offset '52' για το κανάλι '2', η ακολουθία θα έχει τη μορφή: *d2#1#52#*

Μετά τη δημιουργία της ακολουθίας, η εφαρμογή CGI όπως και στην προηγούμενη περίπτωση, αρχικοποιεί την σειριακή επικοινωνία του Η/Υ και στέλνει στον καταχωρητή δεδομένων την ακολουθία. Η ακολουθία και στην

The figure shows two HTML form frames for the ADC converter. The top frame is labeled "Αρχικό πλαίσιο HTML" (Initial HTML frame) and contains the following elements: a "Channel" dropdown menu with "ch1" selected, a "Data" input field with "0", an "Offset" input field with "0", and a "Read" button. The bottom frame is labeled "Τελικό πλαίσιο HTML δημιουργημένο από CGI" (Final HTML frame generated by CGI) and contains the same elements, but with the "Data" input field now containing "53".

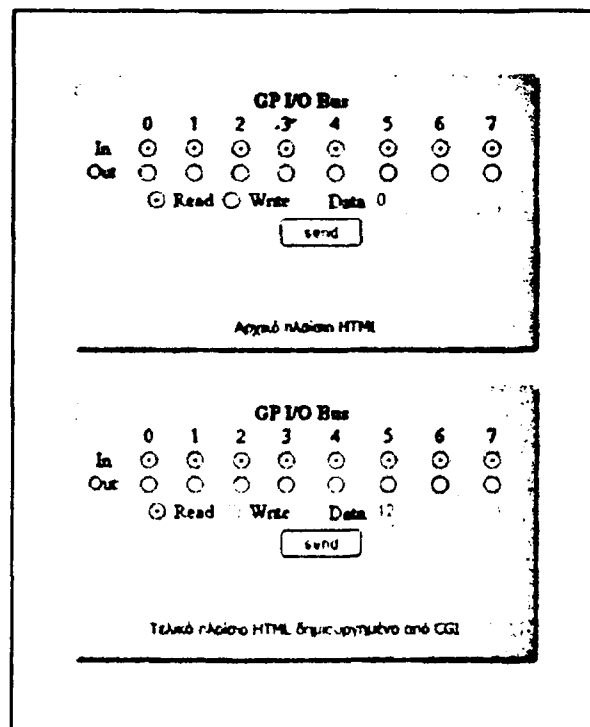
Σχήμα 59: Τα πλαίσια του μετατροπέα ADC

περίπτωση αυτή είναι κωδικοποιημένη κατά ASCII. Αφού αποσταλούν τα δεδομένα στον μικροελεγκτή, εκτελείται η αντίστοιχη εντολή και επιστρέφει το αποτέλεσμα στην ίδια θύρα, εφόσον έχουμε αίτηση για ανάγνωση καναλιού. Η εφαρμογή CGI δημιουργεί το πλαίσιο του κώδικα HTML τοποθετώντας στην ετικέτα εισόδου data το αποτέλεσμα της μέτρησης ή τίποτα αν προηγήθηκε εγγραφή τιμής offset (σχήμα 59).

Το αρχείο CGI της θύρας GPI/O, λειτουργεί κατά παρόμοιο τρόπο με τα δύο προαναφερθέντα αρχεία. Εισάγει τις τιμές των ετικετών εισόδου από το αντίστοιχο αρχείο HTML. Οι τιμές των ετικετών εισόδου που δείχνουν ποια bits της θύρας έχουν δηλωθεί σαν είσοδος και ποια σαν έξοδος, προστίθενται με λογικό ΚΑΙ ώστε να προκύψει ένα byte, το οποίο θα εγγραφεί απευθείας στον καταχωρητή DDRB του μικροελεγκτή. Στη συνέχεια διενεργεί έλεγχο, αν δηλαδή ο χρήστης έχει ζητήσει ανάγνωση ή εγγραφή της θύρας. Ανάλογα με την απαίτηση του χρήστη, δημιουργείται η αντίστοιχη ακολουθία με τις τιμές που προέκυψαν από την ανάγνωση των ετικετών εισόδου. Στην περίπτωση αυτή, οι ακολουθίες αρχίζουν με τον χαρακτήρα 'g' και μετά από την τιμή της κάθε ετικέτας εισόδου τοποθετείται ο χαρακτήρας '#'. Οι περιπτώσεις ακολουθιών που προκύπτουν είναι οι εξής:

- Read:
g"in/out byte"#read#
- Write:
g"in/out byte"#write"#data"#

Για παράδειγμα, στην περίπτωση εγγραφής της τιμής '120' στη θύρα έχοντας δηλώσει όλα τα bits ως εισόδους, η ακολουθία που θα προέκυπτε θα είχε τη μορφή: *g255#1#120*.



Σχήμα 60: Τα πλαίσια της θύρας GPI/O

Το αρχείο CGI του σκανδαλισμού της κάμερας, όταν καλείται αποστέλλει στη σειριακή θύρα του

Η εφαρμογή CGI, εγγράφει στη συνέχεια την ακολουθία που είναι κωδικοποιημένη κατά ASCII, στον καταχωρητή δεδομένων της σειριακής θύρας επικοινωνιών του Η/Υ. Η ακολουθία αποστέλλεται στον μικροελεγκτή, και αφού εκτελέσει την αντίστοιχη ενέργεια επιστρέφει το αποτέλεσμα στην ίδια θύρα. Η εφαρμογή CGI δημιουργεί τον αντίστοιχο κώδικα HTML και τοποθετεί το αποτέλεσμα της διεργασίας του μικροελεγκτή στην ετικέτα εισόδου data (σχήμα 60).

Το αρχείο CGI του σκανδαλισμού της κάμερας, όταν καλείται αποστέλλει στη σειριακή θύρα επικοινωνιών του Η/Υ, το χαρακτήρα 't' ο οποίος οδηγείται έπειτα στο μικροελεγκτή. Στη συνέχεια ο μικροελεγκτής επιστρέφει στην ίδια θύρα το αποτέλεσμα ελέγχου σκανδαλισμού της κάμερας. Αν ανιχνευτεί σκανδαλισμός, τότε επιστρέφεται ο χαρακτήρας '1' και το αρχείο CGI καλεί το αρχείο HTML το οποίο θα εμφανίσει το καρέ της στιγμής του σκανδαλισμού και η διαδικασία ελέγχου επαναλαμβάνεται. Στην περίπτωση που δεν ανιχνευτεί σκανδαλισμός τότε η διαδικασία ελέγχου απλά επαναλαμβάνεται αφού περάσει το χρονικό διάστημα των 5 δευτερολέπτων.

4.2.3 Ο κώδικας της Assembly

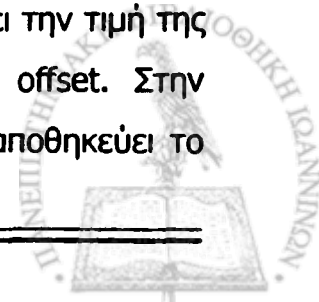
Το τμήμα του κώδικα του μικροελεγκτή, που είναι γραμμένο σε γλώσσα προγραμματισμού Assembly, διενεργεί τη σημαντικότερη και πιο σύνθετη λειτουργία του συστήματος της παρούσας εργασίας. Λαμβάνει εντολές από τον χρήστη μέσω των εφαρμογών HTML και CGI, τις επεξεργάζεται και διενεργεί τις αντίστοιχες λειτουργίες.

Ο πρώτος έλεγχος που διενεργεί, είναι η ανίχνευση ενός εκ των τριών χαρακτήρων w, d, g που αντιστοιχούν στα τμήματά του I²C, ADC και GPI/O αντίστοιχα. Αν δεν ανιχνευτεί ένας εκ των τριών χαρακτήρων, ο μικροελεγκτής επιστρέφει την τιμή 'FF' που αποτελεί ένδειξη εσφαλμένης λειτουργίας. Στη συνέχεια ο μικροελεγκτής αναλύει τα υπόλοιπα δεδομένα της ακολουθίας και αντιδρά ανάλογα.

Στην περίπτωση που ανιχνευτεί 'w', ο μικροελεγκτής αντιλαμβάνεται ότι πρόκειται για εντολές που αφορούν το διαύλο I²C. Το επόμενο byte αποθηκεύεται ως η διεύθυνση της συσκευής. Μετά από κάθε έλεγχο byte, ο μικροελεγκτής, αναμένει να ανιχνεύσει τον χαρακτήρα '#', ο οποίος διαχωρίζει ομάδες χαρακτήρων. Σε αντίθετη περίπτωση, τερματίζει την διαδικασία και επιστρέφει την τιμή 'FF', ως ένδειξη εσφαλμένης λειτουργίας. Εφόσον ανιχνευτεί ο χαρακτήρας '#', τότε ελέγχει την τιμή του επόμενου byte. Ανάλογα με την τιμή αυτή, ο μικροελεγκτής αντιλαμβάνεται την εργασία που πρέπει να πράξει, δηλαδή ανάγνωση, εγγραφή του διαύλου ή αποστολή κατάστασης start ή stop. Στην περίπτωση της εγγραφής, ο έλεγχος της ακολουθίας συνεχίζεται και εφόσον ανιχνευτεί ο χαρακτήρας '#', το επόμενο byte που λαμβάνεται, αποθηκεύεται ως τα δεδομένα προς εγγραφή.

Ο μικροελεγκτής στις περιπτώσεις εγγραφής και ανάγνωσης, εκτελεί πλήρη λειτουργία. Δηλαδή, αποστέλλει κατάσταση εκκίνησης, την διεύθυνση της συσκευής μαζί με το bit εγγραφής/ανάγνωσης, αποστέλλει ή λαμβάνει ένα byte δεδομένων και τέλος αποστέλλει κατάσταση τερματισμού. Στην περίπτωση όμως της μη αυτόματης εγγραφής/ανάγνωσης, αποστέλλει μόνο τη διεύθυνση της συσκευής με το bit εγγραφής/ανάγνωσης και τα δεδομένα. Οπότε πρέπει να έχει προηγηθεί η αποστολή κατάστασης εκκίνησης και να ακολουθήσει η αποστολή κατάστασης τερματισμού. Με τον τρόπο αυτό, μπορούν να αποσταλούν ή να ληφθούν δεδομένα περισσότερα του ενός byte.

Στην περίπτωση ανίχνευσης του χαρακτήρα 'd', ο μικροελεγκτής αντιλαμβάνεται ότι πρόκειται για εντολή που αφορά τον μετατροπέα ADC. Το επόμενο byte αποθηκεύεται ως η διεύθυνση του καναλιού και αν δεν ακολουθήσει ο χαρακτήρας '#' τότε η διαδικασία τερματίζεται και επιστρέφεται η τιμή 'FF', ως ένδειξη εσφαλμένης λειτουργίας. Το επόμενο byte καθορίζει αν θα γίνει ανάγνωση του συγκεκριμένου καναλιού ή αν θα εγγραφεί τιμή offset στην αντίστοιχη διεύθυνση της μνήμης EEPROM. Στην πρώτη περίπτωση ο μικροελεγκτής εφόσον ανιχνεύσει τον χαρακτήρα '#', διενεργεί μία ανάγνωση του αντίστοιχου καναλιού και επιστρέφει την τιμή της μέτρησης αφού αφαιρέσει από αυτήν την αντίστοιχη τιμή offset. Στην δεύτερη περίπτωση, μετά την ανίχνευση του χαρακτήρα '#', αποθηκεύει το



επόμενο byte απευθείας στην αντίστοιχη θέση της μνήμης EEPROM επιστρέφοντας την τιμή '0' στην εφαρμογή CGI. Η τιμή offset, χρησιμοποιείται για την διόρθωση της κλίμακας λειτουργίας των αισθητηρίων που είναι συνδεδεμένα στον μετατροπέα ADC. Παράδειγμα, για τον αισθητήρα υγρασίας που χρησιμοποιείται από το σύστημα, απαιτείται η τιμή offset 103, ώστε η κλίμακα μετρήσεων να είναι από 0% έως 100%.

Στην περίπτωση που ανιχνευτεί ο χαρακτήρας 'g', ο μικροελεγκτής αντιλαμβάνεται ότι η εντολή αφορά τη θύρα GPI/O. Το byte που ακολουθεί καθορίζει ποια bits είναι είσοδοι και ποια έξοδοι. Οπότε αφού ληφθεί, αποθηκεύεται στον καταχωρητή DDRB του μικροελεγκτή. Στη συνέχεια αφού ανιχνευθεί ο χαρακτήρας '#', ελέγχεται το επόμενο byte, το οποίο καθορίζει αν θα γίνει ανάγνωση ή εγγραφή της θύρας. Στην περίπτωση της ανάγνωσης, ο μικροελεγκτής προχωρά στην ενέργεια αυτή μόλις ανιχνεύσει τον χαρακτήρα '#'. Τότε, επιστρέφει την τιμή των bits της θύρας. Στην περίπτωση της εγγραφής, το byte που ακολουθεί τον χαρακτήρα '#', εξάγεται στην θύρα. Σε περίπτωση που ένα byte δεν ακολουθείται από τον χαρακτήρα '#', ο μικροελεγκτής, επιστρέφει την τιμή 'FF', ως ένδειξη σφάλματος.

Πρέπει να σημειωθεί, ότι μπορούν να ορισθούν ορισμένα bits ως είσοδοι και ταυτόχρονα τα υπόλοιπα ως έξοδοι. Οπότε ο χρήστης πρέπει να προσέξει ποια δεδομένα θα εξάγει, γιατί τα bits που είναι ορισμένα ως είσοδοι, δεν θα αλλάξουν την κατάστασή τους.

4.3 Αποτίμηση του συστήματος.

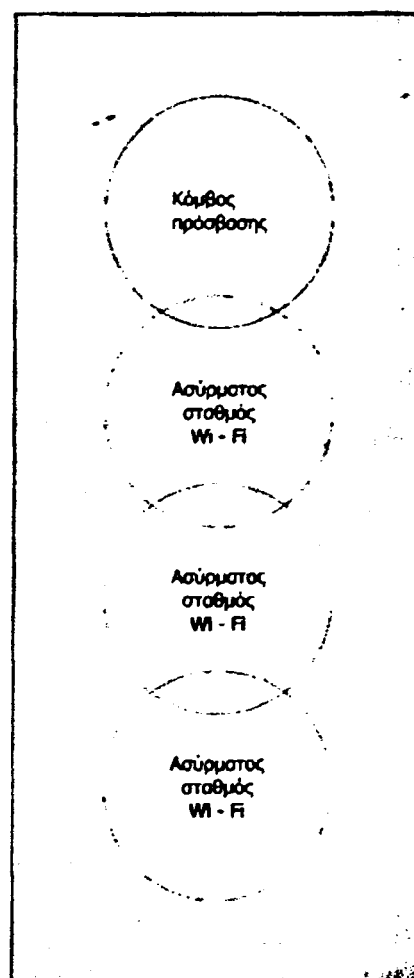
Το σύστημα αυτοματισμού και τηλεμετρίας εργαστηρίου, αποτελεί μια ολοκληρωμένη λύση. Προσφέρει τόσο επιτήρηση μέσω αισθητηρίων όσο και οπτική μέσω της κάμερας του συστήματος. Είναι εύκολο στη λειτουργία του, καθώς ο χρήστης δεν απαιτείται να κατέχει ειδικές γνώσεις.

Το μεγάλο πλεονέκτημα του συστήματος όμως, είναι η δυνατότητα πρόσβασης από οποιοδήποτε σημείο του πλανήτη αρκεί να υπάρχει πρόσβαση στο διαδίκτυο. Έτσι ο χρήστης μπορεί να έχει επιτήρηση του χώρου που έχει εγκαταστήσει το σύστημα, ενώ βρίσκεται οπουδήποτε. Επιπλέον, στο σύστημα μπορούν να συνδεθούν ταυτόχρονα, περισσότεροι του ενός χρήστες, και να χρησιμοποιήσουν τις δυνατότητές του.

Καθώς το σύστημα περιλαμβάνει τρεις σταθμούς ασύρματης διασύνδεσης Wi - Fi και τρεις σταθμούς διασύνδεσης αισθητηρίων, οι θέσεις στις οποίες μπορούν να συνδεθούν αισθητήρια ή συσκευές συνολικά στο σύστημα είναι:

- $3 \times 8 = 24$ θέσεις αισθητηρίων ή συσκευών αναλογικής εξόδου
- $3 \times 8 = 24$ θέσεις ψηφιακών αισθητηρίων η ενεργοποιητών
- $3 \times 128 = 384$ θέσεις αισθητηρίων ή συσκευών I²C

Η εμβέλεια κάλυψης του συστήματος, εξαρτάται από τον χώρο στον οποίο λειτουργεί και από τη διάταξη των σταθμών ασύρματης διασύνδεσης Wi - Fi. Οι τελευταίοι μπορούν να συνδεθούν με τον κόμβο πρόσβασης σε διάταξη αστέρα αλλά και σε διάταξη σημείο - σημείο (σχήμα 61). Η διάταξη αστέρα προσφέρει τυπική ακτίνα κάλυψης σε ανοικτό



σχήμα 61: Η διάταξη σημείου-σημείου

χώρο, περίπου ενός χιλιομέτρου, ενώ η διάταξη σημείο – σημείο, προσφέρει τυπική κάλυψη σε ευθεία περίπου τέσσερα χιλιόμετρα. Μετά από μετρήσεις που έγιναν σε τριώροφο κτήριο, τοποθετώντας τον κόμβο πρόσβασης στον δεύτερο όροφο, υπήρξε επαρκής κάλυψη με ισχύ σήματος 10% σε όλους τους ορόφους στο ίδιο επίπεδο.

Το σύστημα προσφέρει επιπλέον, ευκολία στην τοποθέτηση, διότι δεν απαιτείται οποιαδήποτε προεγκατάσταση. Ο Η/Υ και ο κόμβος πρόσβασης τοποθετούνται στο σημείο που υπάρχει πρόσβαση στο διαδίκτυο. Οι σταθμοί ασύρματης διασύνδεσης Wi – Fi, τοποθετούνται οπουδήποτε εντός της εμβέλειας κάλυψης. Οι σταθμοί προσάρτησης αισθητηρίων και ενεργοποιητών, μπορούν να συνδεθούν οπουδήποτε εντός ακτίνας 200 μέτρων, καθώς αυτή είναι η τυπική ωφέλιμη ακτίνα, για τη διασύνδεση RS-232 [29].

Οι πλειάδα αισθητηρίων και ενεργοποιητών, που είναι δυνατό να συνδεθούν στο σύστημα, το καθιστούν πολυχρηστικό. Το σύστημα είναι δυνατό να χρησιμοποιηθεί ως σύστημα τηλεμετρίας, με τη διασύνδεση αντίστοιχων αισθητηρίων. Επίσης είναι δυνατό να χρησιμοποιηθεί ως σύστημα ελέγχου και επιτήρησης, συνδέοντας αισθητήρια κίνησης, συσκευές πρόσβασης (ανάγνωση κάρτας, ηλεκτρονικές κλειδαριές κ.α.) καθώς και επιπλέον κάμερες. Τέλος είναι δυνατό να χρησιμοποιηθεί και ως σύστημα διαχείρισης κατοικίας ή επαγγελματικού χώρου, συνδέοντας στο σύστημα τις οικιακές συσκευές με ενεργοποιητές και αισθητήρια για τον έλεγχο του χώρου.

4.4 Μελλοντική εξέλιξη του συστήματος

Το σύστημα της παρούσας διπλωματικής εργασίας αποτελείται από πρόσφατα τεχνολογικά επιτεύγματα, όπως ο σταθμός ασύρματης διασύνδεσης, η IP κάμερα και η ασύρματη δικτύωση Wi - Fi. Για το λόγο αυτό, η δυναμική του για μελλοντική εξέλιξη, είναι μεγάλη.

Αρχικά θα μπορούσε να τροποποιηθεί το HTML τμήμα του λογισμικού του συστήματος ώστε να είναι δυνατή η πλοήγηση σε αυτό από κινητά τηλέφωνα τα οποία διαθέτουν τεχνολογία ταχείας μεταφοράς δεδομένων (3G, EDGE

κλη). Η τροποποίηση που θα χρειαζόταν να γίνει είναι πάνω στην ανάλυση των πλαισίων.

Επιπλέον, θα μπορούσε να αυξηθεί η ακτίνα κάλυψης του συστήματος με την αναβάθμιση της ασύρματης δικτύωσης Wi – Fi σε ασύρματη δικτύωση WiMAX. Έτσι η ακτίνα κάλυψης από τα 300 μέτρα που είναι κατά μέσω όρο με το Wi – Fi θα μπορούσε να αυξηθεί σε μερικά χιλιόμετρα με το WiMAX [14,26,33].

Μια μελλοντική προοπτική για το σύστημα είναι η λειτουργία του ασυρμάτου σταθμού διασύνδεσης και ως Web server. Με τον τρόπο αυτό, δεν θα είναι υποχρεωτική η ύπαρξη Η/Υ στο σύστημα. Έτσι το σύστημα θα γίνει μικρότερο και πιο ευέλικτο καθώς και πιο οικονομικό αφού το κόστος ενός Η/Υ δεν είναι αμελητέο. Για να γίνει αυτό, θα πρέπει να δημιουργηθεί κατάλληλο λογισμικό για το σταθμό ασύρματης διασύνδεσης, ώστε να εκτελεί όλες τις απαιτούμενες ενέργειες ενός Web server.



[16] ANALOG DEVICES: ADM231L–ADM234L/ADM236L–ADM241L (2005)
http://www.analog.com/UploadedFiles/Data_Sheets/ADM231L_232L_233L_234L_236L_237L_238L_239L_241L.pdf

[17] Atop Technologies, Inc: ABLELink Serial-Ethernet Server GW21W (2003)

[18] LevelOne: WCS-2030 11g Wireless IP Network Camera (2004)

[19] NETGEAR, Inc.: NETGEAR Wireless Router WGR-614 (2004)

[20] Philips Semiconductors: THE I²C-BUS SPECIFICATION VERSION 2.1 (2000)
http://www.nxp.com/acrobat_download/literature/9398/39340011.pdf

[21] ATMEL Corporation: AVR315: Using the TWI module as I2C master (2004)
http://www.atmel.com/dyn/resources/prod_documents/doc2564.pdf

[22] ATMEL Corporation: AVR100: Accessing the EEPROM (2006)
http://www.atmel.com/dyn/resources/prod_documents/doc0932.pdf

[23] ATMEL Corporation: AVR103: Using the EEPROM Programming Modes (2005)
http://www.atmel.com/dyn/resources/prod_documents/doc2578.pdf

[24] ANALOG DEVICES: AD7414/AD7415 Temperature sensors (2005)
http://www.analog.com/UploadedFiles/Data_Sheets/AD7414_7415.pdf

[25] Humirel, Inc: HM1500LF, Technical Specifications (2005)
<http://www.humirel.com/product/fichier/HM1500LF-HPC062RevB.pdf>

[26] www.wi-fi.org

[27] www.atmel.com

[28] www.avrferaks.net

[29] www.analog.com

[30] www.atop.com.tw

[31] www.w3schools.com

[32] www.htmlcodetutorial.com

[33] www.wikipedia.org

[34] www.stee.com



ΠΑΡΑΡΤΗΜΑ Α

Το λογισμικό του συστήματος.

A.1 Ο κώδικας της γλώσσας HTML.

A.1.1 Κεντρική ιστοσελίδα

Το παρακάτω τμήμα κώδικα αποτελεί τον κορμό του συνολικού κώδικα HTML. Μέσα στο τμήμα αυτό ορίζονται οι θέσεις και το μέγεθος των περιοχών που θα καταλαμβάνουν τα επιμέρους τμήματα του κώδικα HTML.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-
7" />
<title>thesis</title>
</head>
<frameset rows="10%,90%">
<frame noresize="noresize" src="Titlecell.html">
<frameset cols="15%,85%" bordercolor="#0000FF">
<frame noresize="noresize" src="linkscell1.html">
<frameset rows="30%,70%">
<frameset cols="30%,35%,35%">
<frame noresize="noresize" src="titlecell.html" name="twicell">
<frame noresize="noresize" src="titlecell.html" name="adccell">
<frame noresize="noresize" src="titlecell.html" name="gpiocell">
</frameset>
<frame noresize="noresize" src="cameracell2.html">
</frameset>
</frameset></frameset>
<noframes></noframes>
</html>
```

A.1.2 Πλαίσιο τίτλου

Το τμήμα αυτό του κώδικα HTML περιέχει μόνο τον τίτλο της ιστοσελίδα

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-
7" />
<title>titlecell</title>
<style type="text/css">
<!--
body {
background-color: #0099FF;
}
.style1 {
```



```

font-family: Arial, Helvetica, sans-serif;
font-size: large;
}
-->
</style></head>

<body>
  <h2 align="center" class="style1">Ασύρματο Σύστημα Αυτοματισμού και
  Τηλεμετρίας - Ευστράτιος Τζοβάρας 2006</h2>
</body>
</html>

```

A.1.3 Πλαίσιο διασυνδέσεων

Το παρακάτω τμήμα του κώδικα HTML, περιέχει διασυνδέσεις για τις ρυθμίσεις των συσκευών του συστήματος, καθώς και για διασυνδέσεις για τους τρεις σταθμούς προσάρτησης αισθητηρίων.

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-
1">
<title>system links</title>
<style type="text/css">
<!--
body {
  background-color: #0099FF;
}
.style3 {font-family: Verdana, Arial, Helvetica, sans-serif; font-
size: medium;}
.style4 {
  font-family: Arial, Helvetica, sans-serif;
  font-size: 12px;
}
body,td,th {
  color: #000000;
}
-->
</style></head>
<SCRIPT language=JavaScript>
  <!--

```



```

function change_frames(file1,file2,file3)
{
parent.twicell.location=file1;
parent.adccell.location=file2;
parent.gpiocell.location=file3;
}
//-->
</script>
<body><p>&nbsp;</p>
      <h2 align="center" class="style3">System Links</h2>
      <p>&nbsp;</p>
      <tr><td      height="50"      align="center"      valign="middle"
bordercolor="#000000">
          <a
href="javascript:change_frames('twicell.html','adccell2.html','gpioce
ll1.html')"><h3 class="style4">W Terminal 1</h3></a>
          <p>&nbsp;</p></td>
      </tr>
      <tr><td      height="50"      align="center"      valign="middle"
bordercolor="#000000">
          <a
href="javascript:change_frames('twicell.html','adccell2.html','gpioce
ll1.html')"><h3 class="style4">W Terminal 2</h3></a>
          <p>&nbsp;</p></td>
      </tr>
      <tr><td      height="50"      align="center"      valign="middle"
bordercolor="#000000">
          <a
href="javascript:change_frames('twicell.html','adccell2.html','gpioce
ll1.html')"><h3 class="style4">W Terminal 3</h3></a>
          <p>&nbsp;</p></td>
      </tr>
      <tr>
          <td      height="50"      align="center"      valign="middle"
bordercolor="#000000">
              <a      href="http://192.168.1.1"      target="_blank"><h3
class="style4">A/P Properties</h3></a>
              <p>&nbsp;</p></td>
      </tr>
      <tr>

```



```
<td height="50" align="center" valign="middle"
bordercolor="#000000">
  <h3 class="style4">
    <a href="http://195.130.115.43:8080"
target="_blank">IP Camera Properties</a></h3></a>
  <p>&nbsp;</p></td>
</tr>
<tr>
  <td height="50" align="center" valign="middle"
bordercolor="#000000">
  <h3 class="style4">
    <a href="http://195.130.115.43:8080"
target="_blank">W Terminal1 Properties</a></h3></a>
  <p>&nbsp;</p></td>
</tr>
<tr>
  <td height="50" align="center" valign="middle"
bordercolor="#000000">
  <h3 class="style4">
    <a href="http://195.130.115.43:8080"
target="_blank">W Terminal2 Properties</a></h3></a>
  <p>&nbsp;</p></td>
</tr>
<tr>
  <td height="50" align="center" valign="middle"
bordercolor="#000000">
  <h3 class="style4">
    <a href="http://195.130.115.43:8080"
target="_blank">W Terminal3 Properties</a></h3></a>
  <p>&nbsp;</p></td>
</tr>
<tr>
  <td height="50" align="center" valign="middle"
bordercolor="#000000">
  <h3 class="style4">
    <a href="http://195.130.115.43:8080"
target="_blank">User Manual</a></h3></a>
  <p>&nbsp;</p></td>
</tr>
</body>
</html>
```



A.1.4 Πλαίσιο κάμερας

Το παρακάτω τμήμα κώδικα, είναι υπεύθυνο για την εμφάνιση της εικόνας της κάμερας του συστήματος. Εκτός από εντολές HTML περιέχει και εντολές Javascript, οι οποίες ρυθμίζουν το μέγεθος, το είδος και τη δυνατότητα της ψηφιακής μεγέθυνσης της εικόνας, καθώς και τη διαδρομή προέλευσής της.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-
1">
<title>Untitled Document</title>
<style type="text/css">
<!--
body {
    background-color: #99CCFF;
}
-->
</style></head>

<body>
    <table width="100%" height="100%" align="center">
        <td width="100%" height="100%" align="center"><p>
            <SCRIPT language=JavaScript>
                <!--
                    if ((navigator.appName == "Microsoft Internet
Explorer") && (navigator.platform != "MacPPC"))
                    {
                        document.write("<OBJECT ID=\"VAMctrl\" WIDTH=330
HEIGHT=306");
                        document.write(" CLASSID=CLSID:A93B47FD-9BF6-4DA8-97FC-
9270B9D64A6C");
                        document.write("
CODEBASE=\"/plugin/h263ctrl.cab#version=2,0,0,6\">");
                        document.write("<PARAM NAME=\"url\"
VALUE=\"http://195.130.115.43:8080/cgi-bin/video.vam\">");
                        document.write("<PARAM NAME=\"VSize\" VALUE=\"CMS\">");
                        document.write("<PARAM NAME=\"RemoteIP\"
VALUE=\"167880896\">");
                        document.write("<PARAM NAME=\"RemotePort\" VALUE=\"5001\">");
                        document.write("<PARAM NAME=\"RemoteID\" VALUE=\"11238\">");
                        document.write("<PARAM NAME=\"DisplayTimeFormat\"
VALUE=\"1\">");
                        document.write("<PARAM NAME=\"DigitalZoomEdit\"
VALUE=\"true\">");
                        document.write("<PARAM NAME=\"Deblocking\" VALUE=\"true\">");
                        document.write("<PARAM NAME=\"Language\" VALUE=\"EN\">");
                        document.write("</OBJECT>");
                    }
                -->
            </SCRIPT></p></td></table>
</body>
</html>
```



A.1.5 Πλαίσιο διαύλου I²C

Το παρακάτω τμήμα κώδικα HTML δημιουργεί το πλαίσιο διαχείρισης του διαύλου I²C. Σε αυτό, περιέχονται οι επιλογές που παρέχει ο δίαυλος και τα πλαίσια κειμένου από τα οποία εισάγει ο χρήστης δεδομένα.

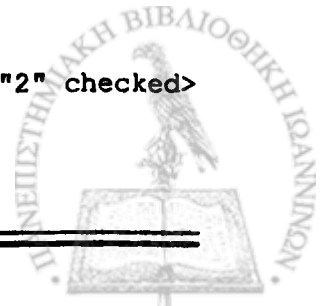
```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-
1">
<title>twi stzo</title>
<style type="text/css">
<!--
body {
    background-color: #99CCFF;
}
-->
</style></head>

<body>

    <td width="100%" height="100%">
        <form method="get" action="/cgi-bin/twicell5.cgi">
            <table width="100%" height="100%" border="0"
cellpadding="0" cellspacing="0">
                <tr>
                    <td colspan="2" align="center"><b>I2C bus</b></td>
                </tr>
                <tr>
                    <td align="center">Address:</td>
                    <td align="center">Data:</td>
                </tr>
                <tr>
                    <td align="center"><input name="a" type="text"
id="a" value="0000000" size="10" maxlength="7"></td>
                    <td align="center"><input name="b" type="text"
id="b" value="00000000" size="11" maxlength="8"></td>
                </tr>
                <tr>
                    <td align="left"><p>
                        <label>
                            <input type="radio" name="c" value="1">
                            Read</label>
                        <br>
                        <label>
                            <input type="radio" name="c"
value="0">
                            Write</label>
                        <br>
                        <label>
                            <input name="c" type="radio" value="2" checked>
                            None</label>
                    </p></td>

```



```

<td align="left"><p>
  <label>
    <input type="radio" name="d" value="2">
    Start</label>
  <br>
  <label>
    <input type="radio" name="d" value="3">
    Stop</label>
  <br>
    <label>
    <input name="d" type="radio" value="4">
    Manual</label>
  <br>
    <label>
    <input name="d" type="radio" value="0" checked>
    None</label>
</p></td>
</tr>
<tr>
  <td colspan="2" align="center"><input
name="twiButton" type="submit" id="twiButton" value="Send"></td>
</tr>
</table>
</form>
</td>
</body>
</html>

```

A.1.6 Πλαίσιο μετατροπέα ADC

Το παρακάτω τμήμα κώδικα HTML, δημιουργεί το πλαίσιο διαχείρισης του μετατροπέα ADC. Σε αυτό περιέχονται τα κανάλια του μετατροπέα, το πλαίσιο κειμένου για τα δεδομένα και για την τιμή offset.

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-
1">
<title>Untitled Document</title>
<style type="text/css">
<!--
body {
  background-color: #99CCFF;
}
-->
</style></head>

<body>
<td valign="top">
<form name="form2" method="get" action="/cgi-bin/adccell3.cgi">
  <table width="100%" height="100%" border="0"
cellpadding="0" cellspacing="0">
    <tr>
      <td colspan="3" align="center"><b>ADC Bus</b></td>
    </tr>
  </table>

```



```

        <td align="center">Channel:</td>
        <td align="center">Data:</td>
        <td align="center">Offset:</td>
    </tr>
    <tr>
        <td align="center"><select name="e" id="e">
            <option value="0">ch1</option>
            <option value="1">ch2</option>
            <option value="2">ch3</option>
            <option value="3">ch4</option>
            <option value="4">ch5</option>
            <option value="5">ch6</option>
            <option value="6">ch7</option>
            <option value="7">ch8</option>
        </select></td>
        <td align="center"><input name="f" type="text"
id="f" value="00000000" size="11" maxlength="8"></td>
        <td align="center"><input name="g" type="text"
id="g" value="00000000" size="11" maxlength="8"></td>
    </tr>
    <tr>
        <td align="center">Offset:<input name="h"
type="checkbox" id="h" value="1"></td>
        <td align="center"><input name="read" type="submit"
id="read" value="Read"></td>
        <td>&nbsp;</td>
    </tr>
</table>
</form></td>

</body>
</html>

```

A.1.6 Πλαίσιο θύρας GPI/O

Το τμήμα κώδικα που παρουσιάζεται παρακάτω, δημιουργεί το πλαίσιο διαχείρισης της θύρας GPI/O. Περιέχει την επιλογή εισόδου εξόδου για κάθε bit της θύρας, καθώς και πλαίσιο κειμένου για τα δεδομένα προς και από την θύρα.

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-
1">
<title>Untitled Document</title>
<style type="text/css">
<!--
body {
    background-color: #99CCFF;
}
-->
</style></head>

<body>
<td><form name="form3" method="get" action="/cgi-bin/gpiocell2.cgi">
        <table width="100%" height="100%"border="0"
cellspacing="0" cellpadding="0">

```

```

<!--DWLayoutTable-->
<tr>
  <td colspan="9" align="center"><b>GP I/O
Bus</b></td>
  <td>&nbsp;</td>
</tr>
<tr align="center">
  <td>&nbsp;</td>
  <td>0</td>
  <td>1</td>
  <td>2</td>
  <td>3</td>
  <td>4</td>
  <td>5</td>
  <td>6</td>
  <td>7</td>
  <td>&nbsp;</td>
</tr>
<tr>
  <td align="center">In: <br>
    Out:</td>
  <td align="center"><p>
    <input name="i0" type="radio" value="00"
checked>
    <br>
    <input type="radio" name="i0" value="01">
    <br>
    </p></td>
  <td align="center"><p>
    <input name="i1" type="radio" value="00"
checked>
    <br>
    <input type="radio" name="i1" value="02">
    <br>
    </p></td>
  <td align="center"><p>
    <input name="i2" type="radio" value="00"
checked>
    <br>
    <input type="radio" name="i2" value="04">
    <br>
    </p></td>
  <td align="center"><p>
    <input name="i3" type="radio" value="00"
checked>
    <br>
    <input type="radio" name="i3" value="08">
    <br>
    </p></td>
  <td align="center"><p>
    <input name="i4" type="radio" value="00"
checked>
    <br>
    <input type="radio" name="i4" value="10">
    <br>
    </p></td>
  <td align="center"><p>
    <input name="i5" type="radio" value="00"
checked>
    <br>
    <input type="radio" name="i5" value="20">

```

```

        <br>
        </p></td>
        <td align="center"><p>
            <input name="i6" type="radio" value="00"
checked>
            <br>
            <input type="radio" name="i6" value="40">
            <br>
        </p></td>
        <td align="center"><p>
            <input name="i7" type="radio" value="00"
checked>
            <br>
            <input type="radio" name="i7" value="80">
            <br>
        </p></td>
        <td>&nbsp;</td>
    </tr>
    <tr>
        <td>&nbsp;</td>
        <td colspan="4" align="center">
            <label>
                <input name="j" type="radio" value="0" checked>
                Read</label>
                <label>
                    <input type="radio" name="j" value="1">
                    Write</label></td>
        <td colspan="4" align="center">Data: <input
name="k" type="text" id="k" value="00000000" size="11"
maxlength="8"></td>
        <td>&nbsp;</td>
    </tr>
    <tr>
        <td height="31">&nbsp;</td>
        <td colspan="8" align="center">
            <input name="write" type="submit"
id="write" value=" send "></td>
    </tr>
</table>
</form></td>
</body>
</html>

```

A.1.7 Πλαίσιο ελέγχου σκανδαλισμού της κάμερας Α

Το πλαίσιο αυτό, απλά καλεί κάθε 5 sec το CGI αρχείο ελέγχου του σκανδαλισμού της κάμερας.

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="refresh" content="5;URL=../../../../cgi-
bin/camtr08.cgi">
<title>camera trigger</title>
</html>

```



A.1.8 Πλαίσιο ελέγχου σκανδαλισμού της κάμερας Β

Το πλαίσιο αυτό, εμφανίζει σε ξεχωριστό παράθυρο την εικόνα της διεύθυνσης <http://195.130.115.43:8080/cgi-bin/video.jpg?size=3> και καλεί το αρχείο CGI ελέγχου σκανδαλισμού της κάμερας.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"><head>
<meta http-equiv="refresh" content="10;URL=../../../../cgi-
bin/camtr08.cgi" />
<title>camera trigger on</title>

<SCRIPT TYPE="text/javascript">
<!--
function popup(mylink, windowname)
{
if (! window.focus)return true;
var href;
if (typeof(mylink) == 'string')
    href=mylink;
else
    href=mylink.href;
window.open(href, windowname, 'width=800,height=600,scrollbars=yes');
return false;
}
//-->
</SCRIPT>
</head>

<body onLoad="popup('http://195.130.115.43:8080/cgi-
bin/video.jpg?size=3', 'ad')">
</body>
</html>
```

A.2 Ο κώδικας της γλώσσας C με χρήση του προτύπου CGI.

A.2.1 Ο κώδικας του διαύλου I²C

Το παρακάτω τμήμα κώδικα CGI, επεξεργάζεται τα δεδομένα που λαμβάνει από το αντίστοιχο πλαίσιο HTML, και αφού ελέγξει ποια συνθήκη αυτά ικανοποιούν, αποστέλλει την κατάλληλη ακολουθία bit στην σειριακή θύρα. Στη συνέχεια λαμβάνει από αυτήν, την απάντηση του μικροελεγκτή και την παρουσιάζει στο πλαίσιο HTML που ξαναδημιουργείται από τον κώδικα CGI.

```
#include <stdio.h>
#include <stdlib.h>
#include <windows.h>
#include <time.h>
#include <string.h>

int main( )
```

```
{
    char *data, a1[8], b1[8], c1[4], d1[4];
    char t1[12]="w";
    char t2[20]="w";
    char t3[12]="w";
    long int a, b, c, d;
    //λήψη δεδομένων από το HTML πλαίσιο
    data = getenv("QUERY_STRING");
    sscanf (data, "a=%ld&b=%ld&c=%ld&d=%ld", &a, &b, &c, &d);
    //επεξεργασία των δεδομένων με συνθήκες
    sprintf(a1, "%d", a);
    sprintf(b1, "%d", b);
    sprintf(c1, "%d", c);
    sprintf(d1, "%d", d);

    strcat(t1, a1);
    strcat(t1, c1);
    strcat(t2, a1);
    strcat(t2, c1);
    strcat(t2, b1);
    strcat(t3, a1);
    strcat(t3, d1);

    HANDLE fileHandle;
    DCB dcb;
    COMMTIMEOUTS cmt;
    DWORD write=-1, read=-1;
    char dataw[20]="";
    char datar[20]="";
    int size=10;
    int sl=-1;

    if (c==0 || c==1)
    switch(c)
    {
        case 0:
            strcpy(dataw, t2) ;
```



```
        break;
    case 1:
        strcpy(dataw,t1) ;
        break;
}

else if (d==2 || d==3)

switch(d)
{
    case 2:
        strcpy(dataw,t3) ;
        break;
    case 3:
        strcpy(dataw,t3) .;
        break;
}

//αρχικοποίηση και λειτουργία της σειριακής θύρας
fileHandle=CreateFile("COM1",GENERIC_READ|GENERIC_WRITE,0
,0,OPEN_EXISTING,0,0);

if(fileHandle==INVALID_HANDLE_VALUE)
{
    printf("Cant open serial port\n");
    exit(-1);
}

FillMemory(&dcb, sizeof(dcb), 0);
if(!BuildCommDCB("9600,n,8,1",&dcb))
{
    printf("error in BuildCommDCB...\n");
    exit(-1);
}

if(!SetCommState(fileHandle, &dcb))
{
    printf("error in SetCommState...\n");
    exit(-1);
}
```

```

if(!SetupComm(fileHandle,1024,1024))
{
    printf("error in SetupComm...\n");
    exit(-1);
}

cmt.ReadIntervalTimeout=1000;
cmt.ReadTotalTimeoutMultiplier=1000;
cmt.ReadTotalTimeoutConstant=1000;
cmt.WriteTotalTimeoutConstant=1000;
cmt.WriteTotalTimeoutMultiplier=1000;

if(!SetCommTimeouts(fileHandle,&cmt))
{
    printf("error in SetCommTimeOuts...\n");
    exit(-1);
}

WriteFile(fileHandle,dataw,strlen(dataw),&write,NULL);
sl=sleep(2);
ReadFile(fileHandle,datar,size,&read,NULL);

```

//Δημιουργία του πλαισίου HTML

```

printf ("content type: text/html\n\n");
printf ("<html>\n");
printf ("<head>\n");
printf ("<title>twicell stzo</title>\n");
printf ("<style type=text/css>\n");
printf ("<!-- body {background-color:#0099ff;} -->
</style></head>\n");
printf ("<body>\n");
printf ("<td width=200 heigth=120>\n");
printf ("<form name=form1 method=get action=/cgi-
bin/twicell4.cgi>\n");
printf ("<table width=200 height=120 border=0 cellpadding=0
cellspacing=0>\n");
printf ("<tr>\n");
printf ("<td colspan=2 align=center><b>%s\n",datar);
printf ("</b></td>\n");
printf ("</tr>\n");

```



```
printf("<tr>\n");
printf("<td align=center>Address:</td>\n");
printf("<td align=center>Data:</td>\n");
printf("</tr>\n");
printf("<tr>\n");
printf("<td align=center><input name=a type=text id=a
value=0 size=10 maxlength=7></td>\n");
printf("<td align=center><input name=b type=text id=b
value=0 size=11 maxlength=8></td>\n");
printf("</tr>\n");
printf("<tr>\n");
printf("<td align=center><p>\n");
printf("<label>\n");
printf("<input type=radio name=c value=1>\n");
printf("Read</label>\n");
printf("<br>\n");
printf("<label>\n");
printf("<input type=radio name=c value=0>\n");
printf("Write</label>\n");
printf("<br>\n");
printf("</p></td>\n");
printf("<td align=center><p>\n");
printf("<label>\n");
printf("<input type=radio name=d value=2>\n");
printf("Start</label>\n");
printf("<br>\n");
printf("<label>\n");
printf("<input type=radio name=d value=3>\n");
printf("Stop</label>\n");
printf("<br>\n");
printf("</p></td>\n");
printf("</tr>\n");
printf("<tr>\n");
printf("<td colspan=2 align=center><input name=twiButton
type=submit id=twiButton value=Send></td>\n");
printf("</tr>\n");
printf("</table>\n");
printf("</form>\n");
printf("</body>\n");
printf("</html>\n");
```



```
        return(0);  
    }
```

A.2.2 Ο κώδικας του μετατροπέα ADC

Το ακόλουθο τμήμα κώδικα CGI, λειτουργεί όπως και το προηγούμενο.

```
#include <stdio.h>  
#include <stdlib.h>  
#include <windows.h>  
#include <time.h>  
#include <string.h>  
  
int main ()  
{  
    //λήψη δεδομένων από το πλαίσιο HTML  
    char *data;  
        char e1[8]="";  
        char f1[8]="";  
        char g1[4]="";  
        char s1[12]="d";  
        char s2[20]="d";  
    long e, f, g;  
  
    data = getenv("QUERY_STRING");  
    sscanf (data, "e=%ld&f=%ld&g=%ld", &e, &f, &g);  
  
    //επεξεργασία των δεδομένων με συνθήκες  
    sprintf(e1, "%d", e);  
    sprintf(f1, "%d", f);  
    sprintf(g1, "%d", g);  
  
    HANDLE fileHandle;  
        DCB dcb;  
        COMMTIMEOUTS cmt;  
        DWORD write=-1, read=-1;  
        char dataw[20]="";  
        char datar[20]="";  
        int size=10;
```



```
int s1=-1;

if (g==1)

    {
    strcat(s2,e1);
    strcat(s2,g1);
    strcat(s2,f1);
    strcpy(dataw,s2) ;
    }

else

    {
    char g1[4]="0";
    strcat(s1,e1);
    strcat(s1,g1);
    strcpy(dataw,s1) ;
    }
```

//αρχικοποίηση και λειτουργία της σειριακής θύρας

```
fileHandle=CreateFile("COM1",GENERIC_READ|GENERIC_WRITE,0,0,OPEN_EXISTING,0,0);
```

```
if(fileHandle==INVALID_HANDLE_VALUE)
```

```
{
    printf("Cant open serial port\n");
    exit(-1);
}
```

```
FillMemory(&dcb, sizeof(dcb), 0);
```

```
if(!BuildCommDCB("9600,n,8,1",&dcb))
```

```
{
    printf("error in BuildCommDCB...\n");
    exit(-1);
}
```

```
if(!SetCommState(fileHandle,&dcb))
```

```
{
```



```

        printf("error in SetCommState...\n");
        exit(-1);
    }

    if(!SetupComm(fileHandle,1024,1024))
    {
        printf("error in SetupComm...\n");
        exit(-1);
    }

    cmt.ReadIntervalTimeout=1000;
    cmt.ReadTotalTimeoutMultiplier=1000;
    cmt.ReadTotalTimeoutConstant=1000;
    cmt.WriteTotalTimeoutConstant=1000;
    cmt.WriteTotalTimeoutMultiplier=1000;

    if(!SetCommTimeouts(fileHandle,&cmt))
    {
        printf("error in SetCommTimeOuts...\n");
        exit(-1);
    }

    WriteFile(fileHandle,dataw,strlen(dataw),&write,NULL);
    sl=sleep(2);
    ReadFile(fileHandle,datar,size,&read,NULL);

```

//Δημιουργία του πλαισίου HTML

```

printf ("content type: text/html\n\n");
printf("<html>\n");
printf("<head>\n");
printf("<title>ADCcell</title>\n");
printf("<style type=text/css>\n");
printf      ("<!--      body      {background-color:#0099ff;}      --
></style></head>\n");
printf("<body>\n");
printf("<td align=center>\n");
printf      ("<form      name=form2      method=get      action=/cgi-
bin/adccell3.cgi>\n");

```



```
printf ("<table width=279 height=120 border=0 cellpadding=0
cellspacing=0>\n");
printf ("<tr>\n");
printf ("<td colspan=3 align=center><b>%s\n",datar);
printf ("</b></td></tr>\n");
printf ("<tr>\n");
printf ("<td align=center>Channel:</td>\n");
printf ("<td align=center>Data:</td>\n");
printf ("<td align=center>Offset:</td>\n");
printf ("</tr>\n");
printf ("<tr>\n");
printf ("<td align=center><select name=e id=channel>\n");
printf ("<option value=0>ch1</option>\n");
printf ("<option value=1>ch2</option>\n");
printf ("<option value=2>ch3</option>\n");
printf ("<option value=3>ch4</option>\n");
printf ("<option value=4>ch5</option>\n");
printf ("<option value=5>ch6</option>\n");
printf ("<option value=6>ch7</option>\n");
printf ("<option value=7>ch8</option>\n");
printf ("</select></td>\n");
printf ("<td align=center><input name=dat type=text id=dat value=%s
size=11 maxlength=8></td>\n",datar);
printf ("<td align=center><input name=f type=text id=offsetdat
value=0000 size=11 maxlength=8></td>\n");
printf ("</tr>\n");
printf ("<tr>\n");
printf ("<td>&nbsp;</td>\n");
printf ("<td align=center><input name=read type=submit id=read
value=Read></td>\n");
printf ("<td align=center><input name=g type=checkbox id=offset
value=1></td>\n");
printf ("</tr>\n");
printf ("</table>\n");
printf ("</form></td>\n");
printf ("</body>\n");
printf ("</html>\n");
```

```
return(0);
```

```
}
```



A.2.3 Ο κώδικας της θύρας GPI/O

Το ακόλουθο τμήμα κώδικα CGI, λειτουργεί όπως και το προηγούμενο.

```

#include <stdio.h>
#include <stdlib.h>
#include <windows.h>
#include <time.h>
#include <string.h>

int main ()
{
    //λήψη δεδομένων από το πλαίσιο HTML
    char *data, i8[8], j1[2], k1[8];
    char p1[22]="g"
    char p2[14]="g"
    long i0, i1, i2, i3, i4, i5, i6, i7, j, k, l;

    data = getenv("QUERY_STRING");
    sscanf
    (data, "i0=%ld&i1=%ld&i2=%ld&i3=%ld&i4=%ld&i5=%ld&i6=%ld&i7=%ld&j=%ld&
    k=%ld", &i0, &i1, &i2, &i3, &i4, &i5, &i6, &i7, &g, &k);
    l=i0+i1+i2+i3+i4+i5+i6+i7;

    sprintf(e1, "%d", e);
    sprintf(g1, "%d", g);
    sprintf(h1, "%d", h);

    HANDLE fileHandle;
    DCB dcb;
    COMMTIMEOUTS cmt;
    DWORD write=-1, read=-1;
    char dataw[20]="";
    char datar[20]="";
    int size=10;
    int sl=-1;

    //επεξεργασία των δεδομένων με συνθήκες
    if (j==0 || j==1)

    switch(j)

```



```
{  
  
    case 0:  
        strcat(t3,a1);  
        strcat(t3,d1);  
        strcpy(dataw,p2) ;  
        break;  
  
    case 1:  
        strcat(t3,a1);  
        strcat(t3,d1);  
        strcpy(dataw,p1) ;  
        break;  
  
    default:  
        break;  
  
    //αρχικοποίηση και λειτουργία της σειριακής θύρας  
    fileHandle=CreateFile("COM1",GENERIC_READ|GENERIC_WRITE,0,0,OPEN_Existing,0,0);  
  
    if(fileHandle==INVALID_HANDLE_VALUE)  
    {  
        printf("Cant open serial port\n");  
        exit(-1);  
    }  
  
    FillMemory(&dcb,sizeof(dcb),0);  
    if(!BuildCommDCB("9600,n,8,1",&dcb))  
    {  
        printf("error in BuildCommDCB...\n");  
        exit(-1);  
    }  
  
    if(!SetCommState(fileHandle,&dcb))  
    {  
        printf("error in SetCommState...\n");  
        exit(-1);  
    }  
  
    if(!SetupComm(fileHandle,1024,1024))  
    {  
        printf("error in SetupComm...\n");  
        exit(-1);  
    }  
}
```



```

}

cmt.ReadIntervalTimeout=1000;
cmt.ReadTotalTimeoutMultiplier=1000;
cmt.ReadTotalTimeoutConstant=1000;
cmt.WriteTotalTimeoutConstant=1000;
cmt.WriteTotalTimeoutMultiplier=1000;

if(!SetCommTimeouts(fileHandle,&cmt))
{
    printf("error in SetCommTimeOuts...\n");
    exit(-1);
}

WriteFile(fileHandle,dataw,strlen(dataw),&write,NULL);
sl=sleep(2);
ReadFile(fileHandle,datar,size,&read,NULL);

```

//Δημιουργία του πλαισίου HTML

```

printf ("content type: text/html\n\n");
printf("<html>\n");
printf("<head>\n");
printf("<title>stratos</title>\n");
printf("<style type=text/css>\n");
printf      ("<!--      body      {background-color:#0099ff;}      --
></style></head>\n");
printf("<body>\n");
printf ("<td width=273 height=120><form name=form3 method=get
action=/cgi-bin/gpiocell1.cgi>\n");
printf ("<table width=273 height=120 border=0 cellpadding=0
cellpadding=0>\n");
printf("<tr>\n");
printf("<td colspan=9 align=center><b>%ld</b></td>\n",m);
printf("</tr>\n");
printf("<tr align=center>\n");
printf("<td>&nbsp;   </td>\n");
printf("<td>0</td>\n");
printf("<td>1</td>\n");
printf("<td>2</td>\n");

```



```
printf ("<td>3</td>\n");
printf ("<td>4</td>\n");
printf ("<td>5</td>\n");
printf ("<td>6</td>\n");
printf ("<td>7</td>\n");
printf ("</tr>\n");
printf ("<tr>\n");
printf ("<td align=center>In: <br>\n");
printf ("Out:</td>\n");
printf ("<td align=center><p>\n");
printf ("<input type=radio name=i0 value=0>\n");
printf ("<br>\n");
printf ("<input type=radio name=i0 value=1>\n");
printf ("<br>\n");
printf ("</p></td>\n");
printf ("<td align=center><p>\n");
printf ("<input type=radio name=i1 value=0>\n");
printf ("<br>\n");
printf ("<input type=radio name=i1 value=1>\n");
printf (" <br>\n");
printf ("</p></td>\n");
printf ("<td align=center><p>\n");
printf ("<input type=radio name=i2 value=0>\n");
printf ("<br>\n");
printf ("<input type=radio name=i2 value=1>\n");
printf ("<br>\n");
printf ("</p></td>\n");
printf ("<td align=center><p>\n");
printf ("<input type=radio name=i3 value=0>\n");
printf ("<br>\n");
printf ("<input type=radio name=i3 value=1>\n");
printf ("<br>\n");
printf ("</p></td>\n");
printf ("<td align=center><p>\n");
printf ("<input type=radio name=i4 value=0>\n");
printf ("<br>\n");
printf ("<input type=radio name=i4 value=1>\n");
printf ("<br>\n");
printf ("</p></td>\n");
printf ("<td align=center><p>\n");
printf ("<input type=radio name=i5 value=0>\n");
```



```

printf ("  
\n");
printf ("\n");
printf ("  
\n");
printf ("\n");
printf ("  
\n");
printf ("\n");
printf ("  
\n");
printf ("\n");
printf ("  
\n");
printf ("\n");
printf ("  
\n");
printf ("&nbsp;</td>\n");
printf ("&nbsp;</td>\n");
printf (" &nbsp;</td>\n"); printf (" | | | |
```



```
printf("</html>\n");

return(0);
}
```

A.2.4 Ο κώδικας ελέγχου της κάμερας

Το ακόλουθο τμήμα κώδικα CGI, αποστέλλει στον μικροελεγκτή το χαρακτήρα «t» και αναμένει την απάντηση του μικροελεγκτή. Αν η απάντηση είναι '0', τότε καλεί το εαυτό του (camtr08.cgi) και η διαδικασία συνεχίζεται. Αν ο μικροελεγκτής απαντήσει με την τιμή '1' τότε καλείται το αρχείο camtron01.html.

```
#include <stdio.h>
#include <stdlib.h>
#include <windows.h>
#include <time.h>
#include <string.h>
```

//αρχικοποίηση και λειτουργία της σειριακής θύρας

```
int main ()
{
HANDLE fileHandle;
DCB dcb;
COMMTIMEOUTS cmt;
DWORD write=-1,read=-1;
char dataw[4]="t";
char datar[4]="";
int size=10;
int sl=-1;

fileHandle=CreateFile("COM1",GENERIC_READ|GENERIC_WRITE,0,0,OPEN_EXISTING,0,0);

if(fileHandle==INVALID_HANDLE_VALUE)
{
printf("Cant open serial port\n");
exit(-1);
}
```



```
FillMemory(&dcb, sizeof(dcb), 0);
if(!BuildCommDCB("9600,n,8,1",&dcb))
{
    printf("error in BuildCommDCB...\n");
    exit(-1);
}

if(!SetCommState(fileHandle, &dcb))
{
    printf("error in SetCommState...\n");
    exit(-1);
}

if(!SetupComm(fileHandle, 1024, 1024))
{
    printf("error in SetupComm...\n");
    exit(-1);
}

cmt.ReadIntervalTimeout=1000;
cmt.ReadTotalTimeoutMultiplier=1000;
cmt.ReadTotalTimeoutConstant=1000;
cmt.WriteTotalTimeoutConstant=1000;
cmt.WriteTotalTimeoutMultiplier=1000;

if(!SetCommTimeouts(fileHandle, &cmt))
{
    printf("error in SetCommTimeOuts...\n");
    exit(-1);
}

WriteFile(fileHandle, dataw, strlen(dataw), &write, NULL);
sl=sleep(2);
ReadFile(fileHandle, datar, size, &read, NULL);

datar[4]=0;
```

//επεξεργασία των δεδομένων με συνθήκες

```
if (datar==0)
```



```

{
printf ("content type: text/html\n\n");
printf ("<html>\n");
printf ("<head>\n");
printf ("<meta http-equiv=refresh content=5;URL=../../../cgi-
bin/camtr08.cgi>\n");
printf ("<title>titlecell</title>\n");
printf ("</head>\n");
printf ("</html>\n");

}
else
{

printf ("content type: text/html\n\n");
printf ("<html>\n");
printf ("<head>\n");
printf ("<meta http-equiv=refresh content=5 ;url=../../tests/tests/
camtron01.html>\n");
printf ("<title>titlecell</title>\n");
printf ("</head>\n");
printf ("</html>\n");

}

return(0);
}

```

A.3 Ο κώδικας της γλώσσας Assembly.

Ο κώδικας που παρουσιάζεται παρακάτω, αφορά τη λειτουργία του μικροελεγκτή. Περιέχει τις επιμέρους λειτουργίες του διαύλου I2C, του μετατροπέα ADC και της θύρας GPI/O. Επίσης ρυθμίζει και ελέγχει την σειριακή επικοινωνία του μικροελεγκτή, και τον WatchDog timer.

```

.include "m16def.inc"

.def temp=r16 ;GENERAL PURPOSE
.def temp1=r17 ;GENERAL PURPOSE
.def datadc=r18 ;ADC
.def datin=r19 ;USART
.def datout=r20 ;USART
.def slad=r21 ;TWI

```



```

.def          twidat1=r22 ;TWI
.def          twidat2=r23 ;TWI
.def          data=r24    ;TWI data register
.def          hex=r25     ;HEX
.def          hex1=r26    ;HEX
.def          hex2=r27    ;HEX
.def          hex3=r28    ;HEX

.equ          start=$f8
.equ          mr_slad_ack=$40
.equ          mt_slad_ack=$f8
.equ          mt_data_ack=$28
.equ          mr_data_ack=$50

.org          $0000

                ldi          temp,low(ramend)
                out          spl,temp
                ldi          temp,high(ramend)
                out          sph,temp

                ldi
temp, (1<<DDD2) | (1<<DDD3) | (1<<DDD4) | (1<<DDD5) | (1<<DDD6) | (1<<DDD7
)
                out          ddrd,temp;set    portd,7    as    output
(indication led)
                ldi          temp,0x48;setting the scl at 100KHz
                out          twbr,temp
                ldi
temp, (1<<DDC2) | (1<<DDC3) | (1<<DDC4) | (1<<DDC5) | (1<<DDC6) | (1<<DDC7
)
                out          ddrc,temp;set the pullup resistors for
the unused pins
                ;cbi          portd,6;status led
;----- WatchDog timer -----
                wdr
                in          temp,WDTCR
                ori
temp, (1<<WDTOE) | (1<<WDE) | (1<<WDP0) | (0<<WDP1) | (1<<WDP2)

```



```

out          WDTCR,temp;enable wdt and set for
0.5sec cycle

```

```

;----- USART setup -----

```

```

USART_setup: ldi          temp1,0x00
              ldi          temp,0x67; 9.6Kbps for 16MHz osc
              out          ubrrh,temp1
              out          ubrrl,temp
              ldi          temp1,(0<<U2X)|(0<<MPCM)
              out          UCSRA,temp1
              ldi
              temp1,(0<<RXCIE)|(0<<TXCIE)|(0<<UDRIE)|(1<<RXEN)|(1<<TXEN)|(0<<
UCSZ2)
              out          UCSRB,temp1
              ldi
              temp1,(1<<URSEL)|(0<<UMSEL)|(0<<UPM1)|(0<<UPM0)|(0<<USBS)|(1<<U
CSZ1)|(1<<UCSZ0)|(0<<UCPOL)
              out          UCSRC,temp1

              rjmp         incom

```

```

;-----USART_Transmit-----

```

```

U_T:
buffer
sbis         UCSRA,UDRE ; Wait for empty transmit
buffer
rjmp        U_T
out         UDR,datout ; Put data into buffer to
be send
ret

```

```

;----- USART Receive -----

```

```

U_R:        wdr          ;reset the wdt
sbis         ucscr, rxc
rjmp        U_R
in          datin, udr
ret

```

```

;----- ADC setup -----

```

```

adc_s:      rcall        asciitohex
mov         temp1,hex ;the channel's address

```

```

ori      hex,$20      ;for 8-bits resolution and
the XX channel

out      admux,hex

ldi

temp,(1<<aden)|(1<<adsc)|(1<<adps2)|(1<<adps1)|(1<<adps0);enabl
e adc,start single conversion and set prs /128

out      adcsr,temp

ret

```

```

;----- ADC convert -----

```

```

adc_end:  in      temp,adcsr

sbrs    temp,adif

rjmp    adc_end

in      datadc,ADCH

eeprom_r: sbic    EECR,EWE ;read from eeprom

rjmp    eeprom_r ;the offset value

ldi     temp,$00

out     EEARH,temp ;of the channel XX

out     EEARL,temp1

sbi     EECR,EERE

in     temp1,EEDR

sub     datadc,temp1

mov     datout,datadc

rcall   u_t

ret

```

```

;----- delay 50ms for 16Mhz osc -----

```

```

delay:   push    temp

push    temp1

loop3:   ldi     temp,$fe;for 2 loops Ctotal=8+(4+3a)b

loop1:   ldi     temp1,$57

loop2:   DEC     temp1

brne    loop2

DEC     temp

brne    loop1

pop     temp1

pop     temp

ret

```

```

;----- ascii to hex -----

```



```

asciitohex: rcall u_r      ;first byte check
             cpi          datin,$23
             breq error1
             mov          hex1,datin
             subi hex1,$30
             rcall u_r      ;second byte check
             cpi          datin,$23
             breq end1
             mov          hex2,datin
             subi hex2,$30
             rcall u_r      ;third byte check
             cpi          datin,$23
             breq end2
             mov          hex3,datin
             subi hex3,$30
             rcall u_r      ;"#".end symbol check
             cpi          datin,$23
             brne error1
             ldi          temp,$64
             mul          hex1,temp
             mov          hex1,r0
             ldi          temp,$0a
             mul          hex2,temp
             mov          hex2,r0
             mov          hex,hex1
             add          hex,hex2
             add          hex,hex3
             ret

end1:        mov          hex,hex1
             ret

end2:        ldi          temp,$0a
             mul          hex1,temp
             mov          hex1,r0
             mov          hex,hex1
             add          hex,hex2
             ret

error1:      ldi          datout,$ff
             rcall u_t
             ret

```

----- twi read data -----




```

twi_r:          push  temp
                ldi
                temp, (1<<twint)|(1<<twsta)|(1<<twen);start condition
                out   twcr,temp
;check for the twint flag set
                rcall tw_wait
;check status register, if different go to error
                in    temp,twsr
                andi  temp,0xf8
                cpi   temp,start;start=$f8
                ;rjmp error
;send the slave address+read bit-XXXXXXX1
                ori   slad,$01
                out   twdr,slad;slave address+read bit
                ldi   temp, (1<<twint)|(1<<twen);clear twint
to start transmission of address
                out   twcr,temp
                rcall tw_wait
;wait for address sending verifcation
                in    temp,twsr;check status register, if
different go to error
                andi  temp,0x40
                cpi   temp,mr_slad_ack;mr_slad_ack=$40
                ;brne error
;read first byte
                ldi   temp, (1<<twint)|(1<<twen)
                out   twcr,temp
                rcall tw_wait;check for the twint flag
                in    twidat1,twdr;data received
                out   portb,twidat1
;read second byte
                ldi   temp, (1<<twint)|(1<<twen)
                out   twcr,temp
                rcall tw_wait;check for the twint flag
                in    twidat2,twdr;data received
;returned an ack bit
                ldi   temp, (1<<twint)|(1<<twen)|(1<<twea)
                out   twcr,temp
                rcall tw_wait
                in    temp,twsr

```



```

        andi temp, mr_data_ack; data has been received and
ack returned

        ldi
temp, (1<<twint) | (1<<twen) | (1<<twsto); send stop, end of receiving
        out twcr, temp
        pop temp
        ret

;----- twi write data -----
twi_t:   push temp
        ldi temp, (1<<twint) | (1<<twsta) | (1<<twen)
        out twcr, temp
;check for the twint flag set
        rcall tw_wait
;check status register, if different go to error
        in temp, twsr
        andi temp, 0xf8
        cpi temp, start; start=$f8
        ;brne error
;send the slave address+write bit=XXXXXXX0
        andi sladr, $fe
        out twdr, sladr; slave address+write bit
        ldi temp, (1<<twint) | (1<<twen); clear twint
to start transmission of address
        out twcr, temp
;wait for address sending verification
        rcall tw_wait
;check status register, if different go to error
        in temp, twsr
        andi temp, 0xf8
        cpi temp, mt_sladr_ack
        ;brne error
;send first byte
        mov temp, data
        out twdr, temp
        ldi temp, (1<<twint) | (1<<twen)
        out twcr, temp
        rcall tw_wait
;check status register, if different go to error
        in temp, twsr
        andi temp, 0x28

```



```

        cpi          temp,mt_data_ack
        ;brne error
;send stop,end of sending
        ldi          temp,(1<<twint)|(1<<twen)|(1<<twsto)
        out          twcr,temp
        pop          temp
        ret

error:   sbi          portd,5;blink of the error led
        rcall delay
        cbi          portd,5
        ret

;----- check for the twint flag to be set (0) -----
tw_wait: in          temp,twcr
        sbrs temp,twint
        rjmp tw_wait
        ret

;----- ADC main routine -----
adcsb:   sbi          portd,7          ;function led
        rcall adc_s
        rcall asciitohex
        cpi          hex,$00
        breq adcread
        cpi          hex,$01
        breq adcoffset
        ldi          datout,$ff
        rcall u_t
        ret
adcread: rcall adc_end
        cbi          portd,7
        ret
adcoffset: rcall asciitohex
eeprom_w: sbic EECR,EWE
        rjmp eeprom_w
        ldi          temp,$00
        out          EEARH,temp
        out          EEARL,temp1
        out          EEDR,hex
        sbi          EECR,EEMWE

```

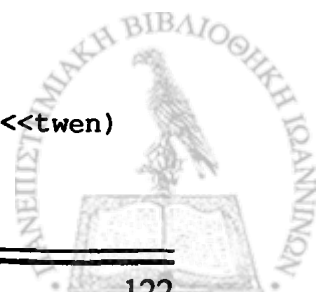


```

sbi      EECR,EWE
cbi      portd,7
ret

;----- twi main routine -----
twisub:  sbi      portd,7      ;function led
        rcall  asciitohex
        cpi      hex,$00      ;write data to twi
        breq   twiwrite
        cpi      hex,$01      ;read data from twi
        breq   twiread
        cpi      hex,$02      ;send a start
condition
        breq   twistart
        cpi      hex,$03      ;send a stop
condition
        breq   twistop
        cpi      hex,$04      ;manual mode
        breq   twiman
        ldi      datout,$ff   ;send an error
        rcall  u_t
        cbi      portd,7
        ret
twiwrite: rcall  asciitohex
        mov      slad,hex
        rcall  asciitohex ;read the data to be send
        mov      data,hex    ;send the data
        rcall  twi_t
        cbi      portd,7
        ret
twiread:  rcall  asciitohex
        mov      slad,hex
        rcall  twi_r
        mov      datout,twidat1
        rcall  u_t
        cbi      portd,7
        ret
twistart: rcall  asciitohex
        mov      slad,hex
        ldi      temp,(1<<twint)|(1<<twsta)|(1<<twen)
        out     twcr,temp

```



```

;check for the twint flag set
        rcall tw_wait
;check status register, if different go to error
        in          twsr
        andi temp,0xf8
        cpi          temp,start;start=$f8
        ;brne error
;send the slave address+write bit=XXXXXXX0
        andi sladr,$fe
        out          twdr,sladr;slave address+write bit
        ldi          temp,(1<<twint)|(1<<twen);clear twint
to start transmission of address
        out          twcr,temp
;wait for address sending verification
        rcall tw_wait
        cbi          portd,7
        ret
twistop: ldi          temp,(1<<twint)|(1<<twen)|(1<<twsto);send
stop,end of sending
        out          twcr,temp
        pop          temp
        cbi          portd,7
        ret
twiman:  rcall asciitohex
        cpi          hex,$00          ;manual write
bytes
        breq twiman_w
        cpi          hex,$01          ;manual read
bytes
        breq twiman_r
        ldi          datout,$ff ;send an error "FF"
        rcall u_t
        cbi          portd,7
        ret
twiman_w: rcall asciitohex
        mov          temp,hex
        out          twdr,temp ;send byte
        ldi          temp,(1<<twint)|(1<<twen)
        out          twcr,temp
        rcall tw_wait
;check status register, if different go to error

```



```

        in            temp,twsr
        andi temp,0x28
        cpi          temp,mt_data_ack
        ;brne error
        cbi          portd,7
        ret

twiman_r:  rcall asciitohex
           ldi          temp,(1<<twint)|(1<<twen);read byte
           out          twcr,temp
           rcall tw_wait;check for the twint flag
           in          twidat1,twdr;data received
           cbi          portd,7
           ret

```

;----- GP I/O main routine -----

```

gpio:      sbi          portd,7          ;function led
           rcall asciitohex
           out          ddrb,hex        ;receive the DDRB
           rcall asciitohex
           cpi          hex,$00        ;GPIO read
           breq gpread
           cpi          hex,$01        ;GPIO write
           breq gpwrite
           ldi          datout,$ff     ;send an error "FF"
           rcall u__t
           cbi          portd,7
           ret

gpread:    in          datout,pinb
           rcall u__t
           cbi          portd,7
           ret

gpwrite:   rcall asciitohex
           out          portb,hex
           cbi          portd,7
           ret

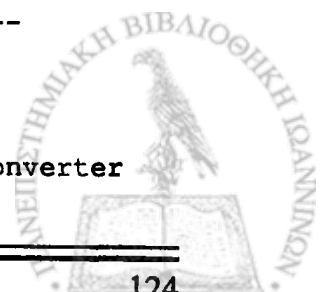
```

;----- main loop (check the incoming commands) -----

```

incom:     ;sbi          portd,6 status led
           rcall u__r
           cpi          datin,$64     ;d=$64 for adc converter

```



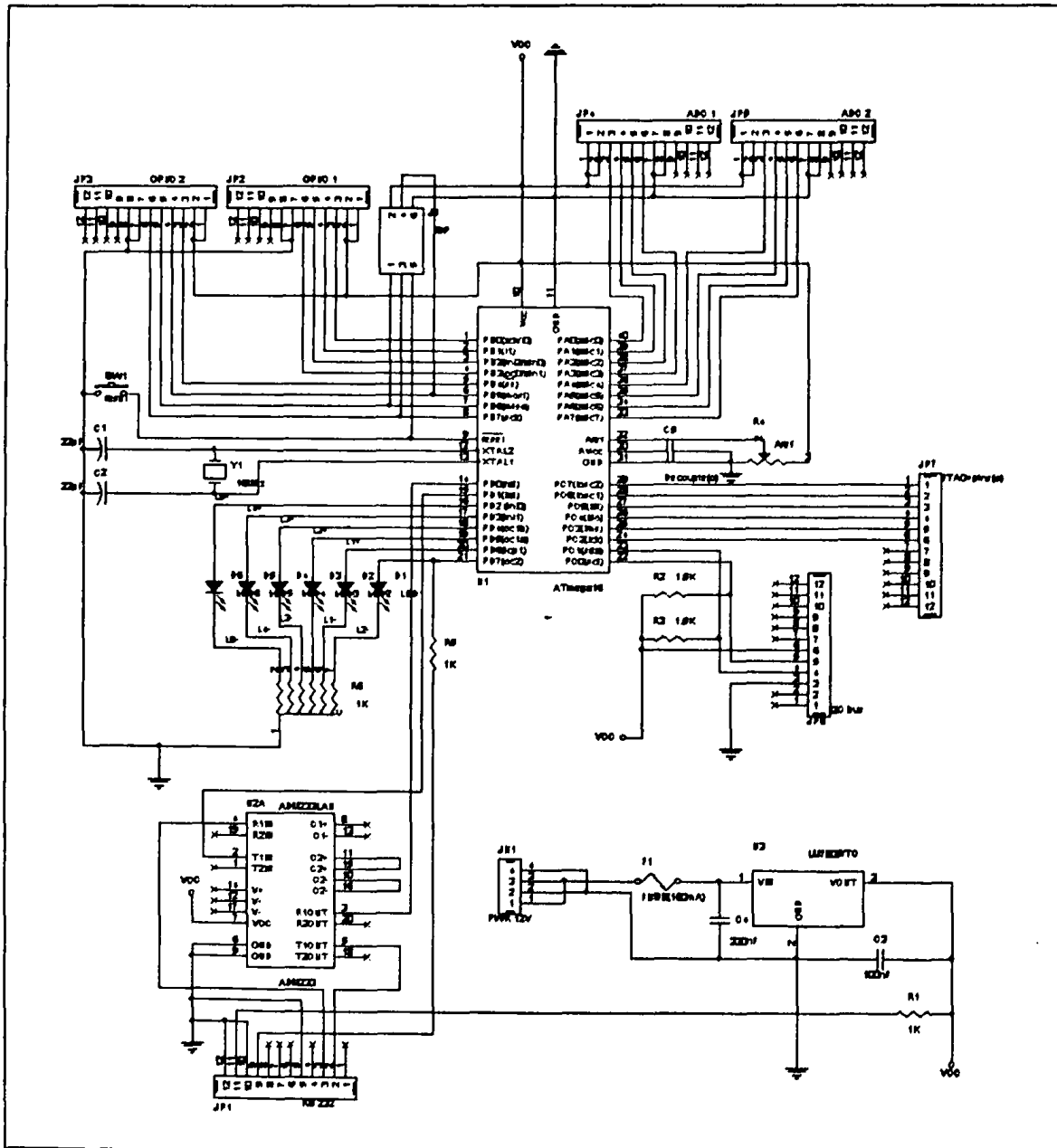
```
      breq  adcsubl
      cpi   datin,$77  ;w=$77 for twi interface
      breq  twisubl
      cpi   datin,$67  ;g=$67 for gp I/O interface
      breq  gpsubl
      wdr                               ;watchdog timer reset
      rjmp  incom
adcsubl:  rcall adcsub
          wdr                               ;watchdog timer reset
          rjmp  incom
twisubl:  rcall twisub
          wdr
          rjmp  incom
gpsubl:   rcall gpio
          wdr                               ;watchdog timer reset
          rjmp  incom
```



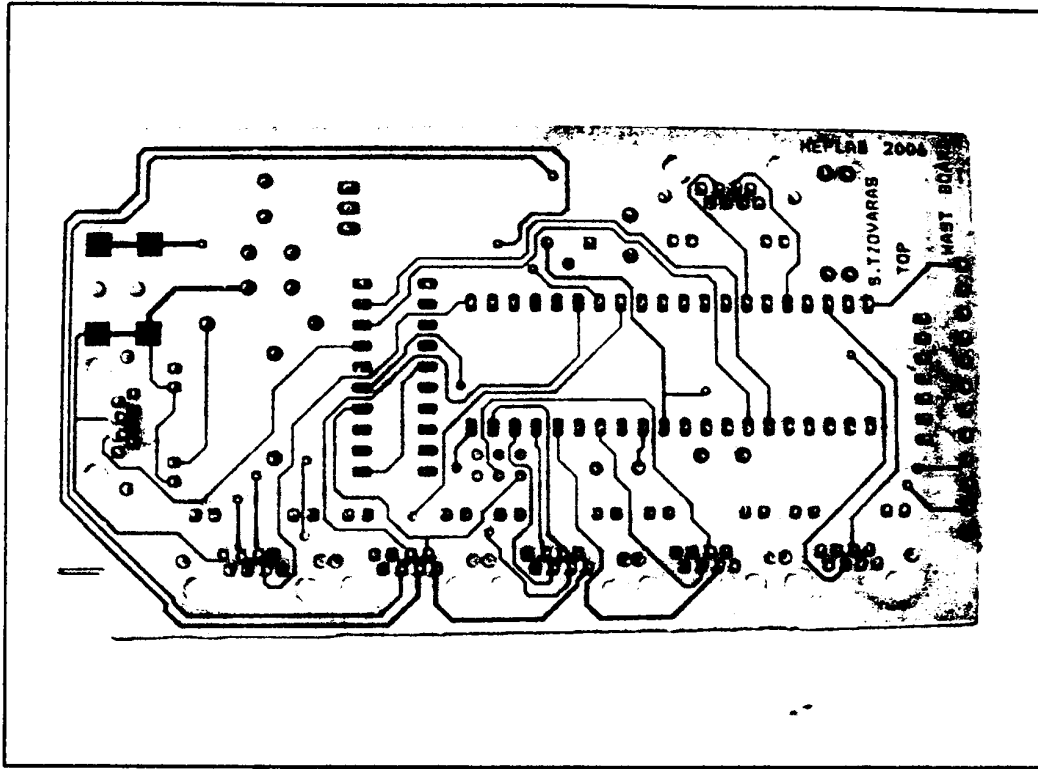
ΠΑΡΑΡΤΗΜΑ Β

Το υλικό του συστήματος

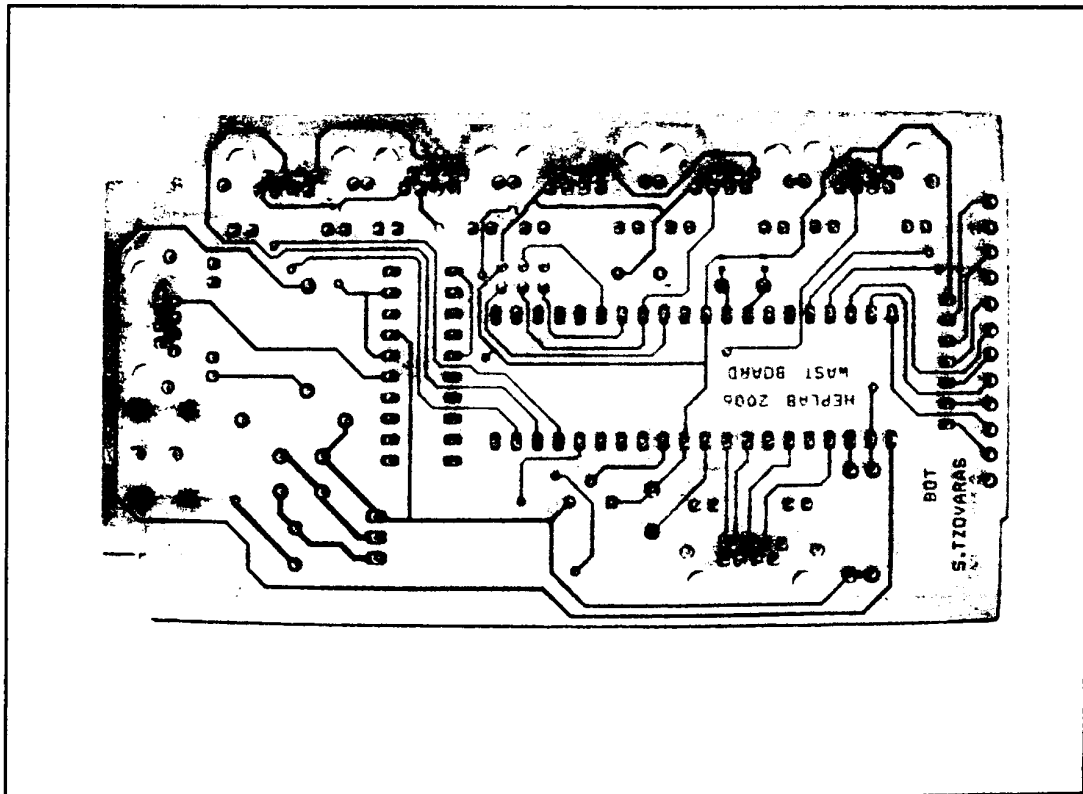
B.1 Ο σταθμός προσάρτησης αισθητηρίων και ενεργοποιητών.



Σχήμα Α: Το σχηματικό διάγραμμα του σταθμού

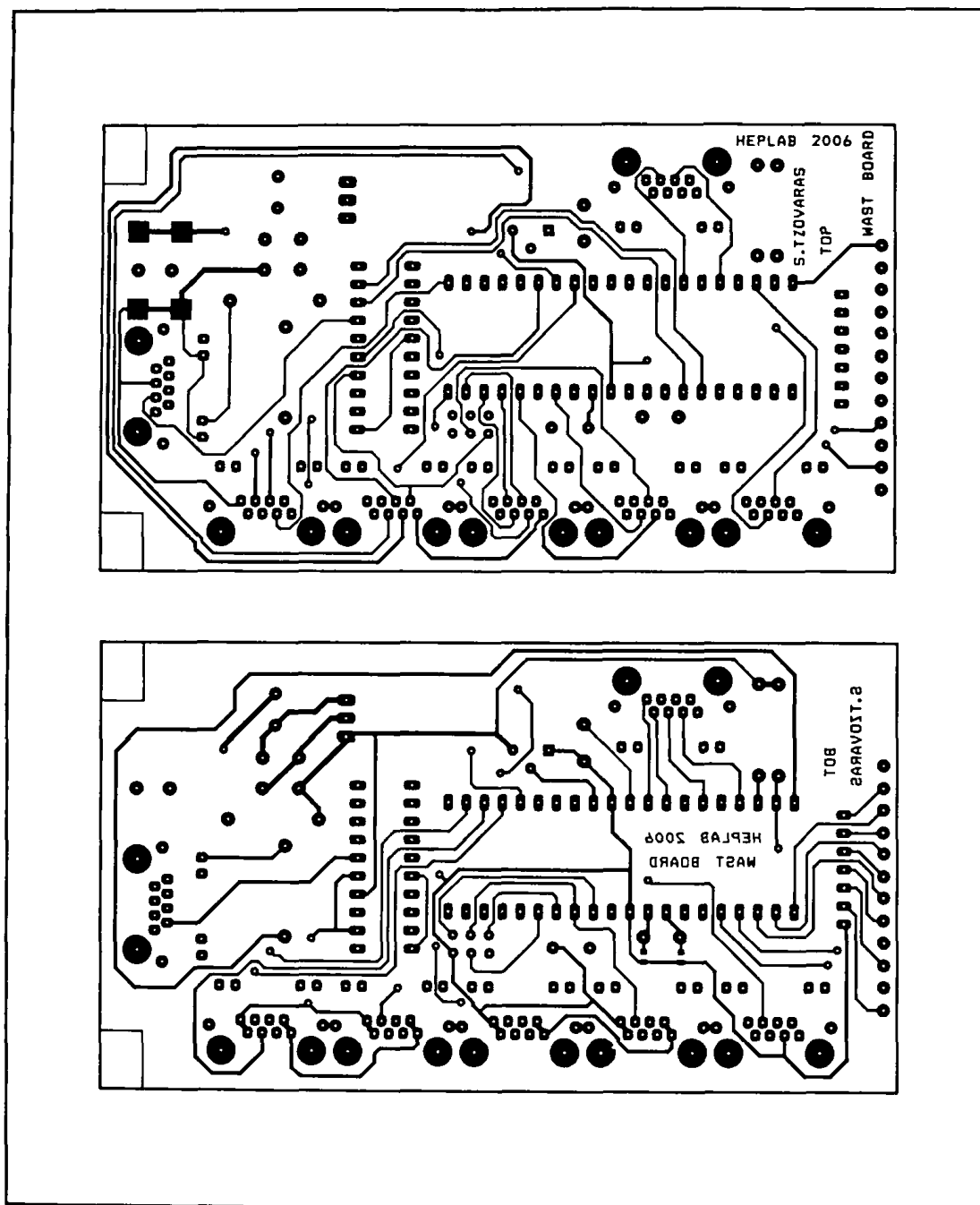


σχήμα Β: Πάνω όψη πλακέτας



σχήμα Γ: Κάτω όψη πλακέτας





σχήμα Α: Τα PCB's της πλακέτας