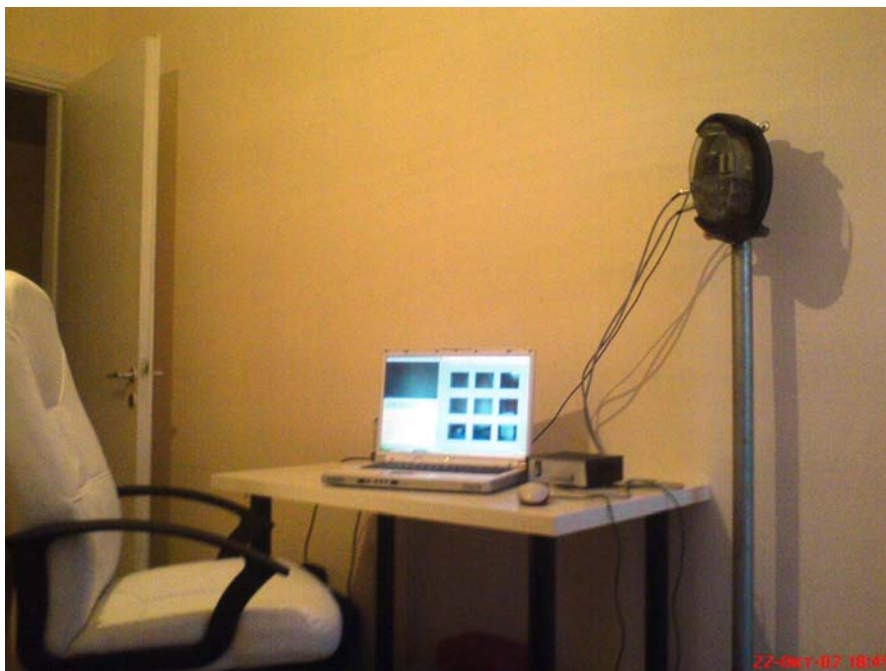




**ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ (Τ.Ε.Ι.) ΚΡΗΤΗΣ**  
**Παράρτημα Χανίων – Τμήμα Ηλεκτρονικής**

## **ΚΙΝΗΜΑΤΙΚΟΣ ΕΛΕΓΧΟΣ ΨΗΦΙΑΚΗΣ ΚΑΜΕΡΑΣ**

**ΜΠΕΛΑΔΑΚΗΣ Μιχαήλ**  
**ΤΡΙΑΝΤΑΦΥΛΛΟΣ Δημήτριος**



### **ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ**

υποβληθείσα για την επί μέρους πλήρωση των προϋποθέσεων του  
Πτυχίου του Ηλεκτρονικού Μηχανικού

Επιβλέπων Καθηγητής: Γ. Ν. Φουσκιτάκης

**Χανιά, Νοέμβριος 2007**

## **ΕΥΧΑΡΙΣΤΙΕΣ**

Ευχαριστούμε:

τον επιβλέποντα καθηγητή Φουσκιτάκη Γεώργιο,  
τον Ριγάκη Ηρακλή για την πολύτιμη βοήθεια του στην κατασκευή της  
πλακέτας και στη σύνταξη του κώδικα assembly,  
τον Χουρδάκη Μανώλη για τη βοήθεια που μας προσέφερε στο Matlab  
τον Γαρεδάκη Ανδρέα για την κατασκευή του βραχίονα,  
τον Φραγκόπουλο Αντώνη για την κατασκευή της βάσης του βραχίονα,  
τους Καμπέλη Νίκο και Κακουράκη Ερμιόνη για την βοήθεια τους στην  
σύνταξη του κειμένου της παρούσας πτυχιακής,  
και τέλος τους γονείς, τους φίλους και τους καθηγητές μας που μας  
στήριξαν κατά τη διάρκεια των σπουδών μας.

## **ΠΕΡΙΛΗΨΗ**

Η παρούσα πτυχιακή αποτελεί εφαρμογή της ρομποτικής. Σχεδιάστηκε και κατασκευάστηκε ένας βραχίονας δύο βαθμών ελευθερίας για τον κινηματικό έλεγχο μιας ψηφιακής κάμερας. Χρησιμοποιήθηκαν DC βηματικοί κινητήρες και ο κινηματικός έλεγχος της κάμερας υλοποιείται μέσω του ηλεκτρονικού υπολογιστή και ενός μικροελεγκτή, ο οποίος προγραμματίστηκε σε γλώσσα Assembly, ενώ ο χρήστης χειρίζεται την κάμερα μέσω του λογισμικού του Matlab, στο πλαίσιο του οποίου αναπτύχθηκαν οι απαραίτητες συναρτήσεις για τον έλεγχό της.

## **ABSTRACT**

The goal of this thesis is the development of a (two degrees of freedom) robotic mechanism for the motion control of a digital camera. Two DC stepper motors are utilized for the camera motion, controlled via a Personal Computer and a properly developed microcontroller that has been programmed via assembly. Matlab routines have been also developed for the motion control of the camera.

## ΠΕΡΙΕΧΟΜΕΝΑ

<i><b>ΕΥΧΑΡΙΣΤΙΕΣ</b></i> .....	- 2 -
<i><b>ΠΕΡΙΛΗΨΗ</b></i> .....	- 3 -
<i><b>ΠΕΡΙΕΧΟΜΕΝΑ</b></i> .....	- 4 -
<i><b>ΕΙΣΑΓΩΓΗ</b></i> .....	- 7 -
<i><b>1.ΠΕΡΙΓΡΑΦΗ ΤΗΣ ΠΕΙΡΑΜΑΤΙΚΗΣ ΔΙΑΤΑΞΗΣ</b></i> .....	- 8 -
1.1 Η Κάμερα.....	- 8 -
1.2 Οι Βηματικοί Κινητήρες .....	- 9 -
1.3 Το Μηχανικό Σύστημα Στήριξης της Κάμερας .....	- 9 -
1.4 Κύκλωμα Ελέγχου Κίνησης .....	- 11 -
1.5 Οι Ρουτίνες Ελέγχου στο Matlab .....	- 16 -
1.6 Οικονομοτεχνική Μελέτη.....	- 17 -
<i><b>2.ΕΓΧΕΙΡΙΔΙΟ ΧΡΗΣΤΗ</b></i> .....	- 18 -
2.1 Εγκατάσταση και Διασύνδεση.....	- 18 -
2.2 Ρυθμίσεις της Κάμερας.....	- 21 -
2.3 Γνωριμία με τις συναρτήσεις .....	- 26 -
<i><b>ΣΥΜΠΕΡΑΣΜΑΤΑ</b></i> .....	- 29 -
<i><b>ΠΑΡΑΡΤΗΜΑ Α'</b></i> .....	- 30 -
<i><b>ΠΑΡΑΡΤΗΜΑ Β'</b></i> .....	- 40 -

## ΚΑΤΑΛΟΓΟΣ ΣΧΗΜΑΤΩΝ

<b>Σχήμα 1.1:</b>	Ψηφιακή κάμερα της εταιρείας Crypto- Budget II Web Camera.....	- 8 -
<b>Σχήμα 1.2:</b>	Η ψηφιακή κάμερα τοποθετημένη στον βηματικό κινητήρα.....	- 8 -
<b>Σχήμα 1.3:</b>	Πίσω όψη του Βηματικού κινητήρα της εταιρείας Astrosyn.....	- 9 -
<b>Σχήμα 1.4:</b>	Μηχανισμός κίνησης του βραχίονα.....	-10-
<b>Σχήμα 1.5:</b>	Ο βραχίονας τοποθετημένος στο κουτί εξωτερικού φωτισμού .....	-10-
<b>Σχήμα 1.6:</b>	Εξωτερικό προστατευτικό γυαλί.....	-10-
<b>Σχήμα 1.7:</b>	Σχηματικό τροφοδοτικού του Ελεγκτή Κίνησης.....	-11-
<b>Σχήμα 1.8:</b>	Σχηματικό κυκλώματος μικροελεγκτή & βαθμίδας οδήγησης κινητήρα....	-13-
<b>Σχήμα 1.9:</b>	PCB της πλακέτας με τα εξαρτήματα.....	-14-
<b>Σχήμα 1.10:</b>	Σχέδιο των αγωγών της πλακέτας.....	-15-
<b>Σχήμα 1.11:</b>	Η πλακέτα συνδεδεμένη και τοποθετημένη στο κουτί.....	-15-
<b>Σχήμα 2.1:</b>	Καλώδιο τροφοδοσίας κατασκευής.....	-18-
<b>Σχήμα 2.2:</b>	Τα καλώδια DIN-5.....	-18-
<b>Σχήμα 2.3:</b>	Καλώδιο RS-232.....	-18-
<b>Σχήμα 2.4:</b>	Η πίσω όψη του κουτιού της κατασκευής.....	-19-
<b>Σχήμα 2.5:</b>	Σύνδεση του καλωδίου DIN-5 με την υποδοχή “UP” στο κουτί.....	-19-
<b>Σχήμα 2.6:</b>	Σύνδεση του καλωδίου DIN-5 με την πάνω υποδοχή του βραχίονα.....	-19-
<b>Σχήμα 2.7:</b>	Σύνδεση του καλωδίου DIN-5 με την υποδοχή “DOWN” στο κουτί.....	-19-
<b>Σχήμα 2.8:</b>	Σύνδεση του καλωδίου DIN-5 με την κάτω υποδοχή του βραχίονα.....	-20-
<b>Σχήμα 2.9:</b>	Το καλώδιο RS-232 συνδεδεμένο στη θύρα της κατασκευής.....	-20-
<b>Σχήμα 2.10:</b>	Η κατασκευή πλήρως συνδεδεμένη.....	-20-
<b>Σχήμα 2.11:</b>	Ο βραχίονας τοποθετημένος στη θέση «0».....	-21-
<b>Σχήμα 2.12:</b>	Χρήση της movecam.m για την τοποθέτηση της κάμερας στο σημείο «0».....	-22-
<b>Σχήμα 2.13:</b>	Εγγραφή κατά την χρήση της movecam.m για την τοποθέτηση της κάμερας στο σημείο «0».....	-22-
<b>Σχήμα 2.14:</b>	Άνοιγμα των ρυθμίσεων της κάμερας με το scnroom.m.....	-23-
<b>Σχήμα 2.15:</b>	Αρχική ρύθμιση της κάμερας .....	-23-
<b>Σχήμα 2.16:</b>	Επιλογή του «Source» από το παράθυρο «SYS Video For Matlab».....	-23-
<b>Σχήμα 2.17:</b>	Επιλογή της ψηφιακής κάμερας του βραχίονα.....	-24-
<b>Σχήμα 2.18:</b>	Τελική ρύθμιση της κάμερας.....	-24-
<b>Σχήμα 2.19:</b>	Αποτέλεσμα μετά το τέλος του scnroom.m.....	-25-
<b>Σχήμα 2.20:</b>	Αποτέλεσμα από την χρήση της συνάρτησης movecam.m.....	-26-
<b>Σχήμα 2.21:</b>	Αποτέλεσμα από την χρήση της συνάρτησης goto.m.....	-27-
<b>Σχήμα 2.22:</b>	Η χρήση της goto.m εκμεταλλευόμενοι τις φωτογραφίες του figure.....	-28-

## ΚΑΤΑΛΟΓΟΣ ΠΙΝΑΚΩΝ

<b>Πίνακας 1.1:</b> Τεχνικά χαρακτηριστικά των βηματικών κινητήρων.....	- 9 -
<b>Πίνακας 1.2:</b> Χρησιμοποιούμενα εξαρτήματα τροφοδοτικού.....	-11-
<b>Πίνακας 1.3:</b> Χρησιμοποιούμενα εξαρτήματα DSP μικτοελεγκτή&οδηγών κινητήρων.....	-13-
<b>Πίνακας 1.4:</b> Κόστος χρησιμοποιηθέντων εξαρτημάτων.....	-17-

## ΕΙΣΑΓΩΓΗ

Η παρούσα εργασία αφορά στο σχεδιασμό και κατασκευή ενός συστήματος που επιτρέπει στο χρήστη να χειρίζεται μια ψηφιακή κάμερα και να εποπτεύει ένα δεδομένο χώρο μέσω ηλεκτρονικού υπολογιστή. Ο μηχανισμός περιλαμβάνει μια ψηφιακή κάμερα, το βραχίονα κίνησης, δύο (2) DC βηματικούς κινητήρες, ένα μικροελεγκτή αλλά και το λογισμικό που καθιστά τον έλεγχο του συστήματος προσιτό και φιλικό προς το χρήστη.

Τα στάδια ανάπτυξης και υλοποίησης του συστήματος έχουν ως εξής: Αρχικά κατασκευάστηκε ο (μηχανικός) βραχίονας αποτελούμενο από μηχανισμό περιστροφής ως προς τον κατακόρυφο και τον εγκάρσιο άξονα μέσω δύο DC βηματικών κινητήρων, αντίστοιχα, στον οποίο τοποθετήθηκε μία ψηφιακή USB κάμερα. Εν συνεχεία κατασκευάστηκε κατάλληλο αναπτυξιακό για τον μικροελεγκτή τύπου 8051, προκειμένου να επιτευχθεί ο προγραμματισμός του επεξεργαστή καθώς και η φυσική του επικοινωνία με τον Η/Υ για τον κινηματικό έλεγχο της κάμερας.

Η κίνηση του βραχίονα επιτυγχάνεται μέσω βηματικών κινητήρων διά των οποίων παρέχεται η δυνατότητα υπολογισμού και ταυτόχρονου ελέγχου (και γνώσης) της θέσης αυτών. Ως εκ τούτου, απεφέχθη η χρήση αισθητήρων για τον προσδιορισμό και έλεγχο της θέσης της κάμερας.

Η λειτουργία του μικροελεγκτή συνίσταται στη λήψη των δεδομένων από τον Η/Υ που περιέχουν τις απαραίτητες πληροφορίες για τον τρόπο κίνησης των κινητήρων. Ο μικροελεγκτής μεταφράζει τα δεδομένα αυτά σε κίνηση και αφού την εκτελέσει, υπολογίζει και επιστρέφει (στον Η/Υ) τη νέα θέση της κάμερας.

Ο κινηματικός έλεγχος υλοποιείται μέσω συναρτήσεων (functions) που αναπτύχθηκαν στο περιβάλλον του λογισμικού Matlab. Παρέχεται η δυνατότητα, από τις δημιουργηθείσες συναρτήσεις, της λύσης του ευθέως και του αντίστροφου κινηματικού προβλήματος. Οι συγκεκριμένες συναρτήσεις σχεδιάστηκαν λαμβάνοντας υπ' όψη το κριτήριο της φιλικότητας προς το χρήστη.

ΗΣ αρχή λειτουργία και χρήσης της κάμερας παρατίθεται δύο κεφάλαια: Στο πρώτο γίνεται μια συνολική περιγραφή της πειραματικής διάταξης, ενώ στο δεύτερο, δίνονται λεπτομερείς οδηγίες προς το χρήστη για τη σωστή εγκατάσταση και λειτουργία της.

# 1. ΠΕΡΙΓΡΑΦΗ ΤΗΣ ΠΕΙΡΑΜΑΤΙΚΗΣ ΔΙΑΤΑΞΗΣ

## 1.1 Η Κάμερα

Στην κατασκευή χρησιμοποιήθηκε μια απλή Web κάμερα της εταιρείας crypto και συγκεκριμένα το μοντέλο Budget II Web Camera, Σχήμα 1.1. Από την κάμερα αφαιρέθηκε η βάση της προκειμένου να τοποθετηθεί εκ νέου στον ένα βηματικό κινητήρα.



Σχήμα 1.1: Ψηφιακή κάμερα της εταιρείας Crypto: Budget II Web Camera.

Η κάμερα έχει στηριχθεί στον κινητήρα μέσω δύο (2) περικοχλίων με στόχο αυτή να ακολουθεί με ακρίβεια την κίνηση του ρότορα, Σχήμα 1.2.



Σχήμα 1.2: Η ψηφιακή κάμερα τοποθετημένη στο βηματικό κινητήρα.



## 1.2 Οι Βηματικοί Κινητήρες

Ο μηχανισμός κίνησης της κάμερας αποτελείται από δύο βηματικούς κινητήρες της εταιρείας Astrosyn, οι EE939D2, Σχήμα 1.3.



**Σχήμα 1.3:** Πίσω όψη του βηματικού κινητήρα της εταιρείας Astrosyn.

Τα τεχνικά χαρακτηριστικά του βηματικού κινητήρα EE939D2 παρουσιάζονται στον Πίνακα 1.1 που ακολουθεί:

**Πίνακας 1.1:** Τεχνικά χαρακτηριστικά των βηματικών κινητήρων.

Τάση λειτουργίας V	12-24 Volt
Ρεύμα λειτουργίας I	0.16 Amps
Γωνία βηματισμού	1.8 Deg
Αντίσταση Τυλιγμάτων R	75 Ohm

Η χρήση βηματικών κινητήρων παρουσιάζει δύο σημαντικά πλεονεκτήματα: Πρώτον, η οδήγηση τους απαιτεί απλό ηλεκτρονικό κύκλωμα με μικρό αριθμό εξαρτημάτων, με ταυτόχρονη δυνατότητα επιλογής της φοράς περιστροφής. Δεύτερον, δεν απαιτείται κύκλωμα ελέγχου της θέσης τους διότι αυτή προσδιορίζεται από το βήμα τους και το συνολικό αριθμό παλμών τάσης κατά την οδήγησή τους (κάθε παλμός τάσης έχει ως αποτέλεσμα την περιστροφή του κινητήρα κατά ένα βήμα=1.8°).

## 1.3 Το Μηχανικό Σύστημα Στήριξης της Κάμερας

Στα Σχήματα 1.4 και 1.5 φαίνεται ο τρόπος με τον οποίο έχουν συνδεθεί οι δύο κινητήρες για να επιτευχθεί η ανεξάρτητη περιστροφή της κάμερας σε δύο άξονες. Στον κινητήρα που βρίσκεται χαμηλά και ελέγχει την περιστροφή ως προς τον κατακόρυφο άξονα, στηρίχθηκε μία βάση αλουμινίου σε σχήμα “L”, με δύο περικόχλια. Πάνω στη βάση αυτή στηρίχθηκε ο δεύτερος κινητήρας, στον οποίο έχει τοποθετηθεί η ψηφιακή κάμερα. Έτσι ο “κάτω” κινητήρας περιστρέφει την κάμερα κατά τον κατακόρυφο άξονα και ο “επάνω” κινητήρας περιστρέφει την κάμερα κατά τον εγκάρσιο άξονα.

Η χρήση γραναζιών δεν κρίθηκε απαραίτητη αφ’ ενός διότι η κατασκευή είναι αρκετά ελαφριά για τους συγκεκριμένους κινητήρες, και αφ’ ετέρου διότι το βήμα των κινητήρων είναι αρκετά μικρό παρέχοντας σχετικά υψηλή ακρίβεια ελέγχου της περιστροφής.



**Σχήμα 1.4 :** Μηχανισμός κίνησης του βραχίονα.

Ο μηχανισμός κίνησης της κάμερας τοποθετήθηκε μέσω μεταλλικής βάσης, σε πλαίσιο φωτιστικού εξωτερικού χώρου, Σχήμα 1.5, και επεξεργάστηκε το εξωτερικό προστατευτικό γυαλί, έτσι ώστε το σημείο στο οποίο “βλέπει” η κάμερα να είναι διάφανο, Σχήμα 1.6. Με τον τρόπο αυτό η κάμερα προστατεύεται από εξωτερικούς παράγοντες.



**Σχήμα 1.5:** Ο βραχίονας τοποθετημένος στο πλαίσιο φωτιστικού εξωτερικού χώρου.



**Σχήμα 1.6 :** Εξωτερικό προστατευτικό γυαλί.

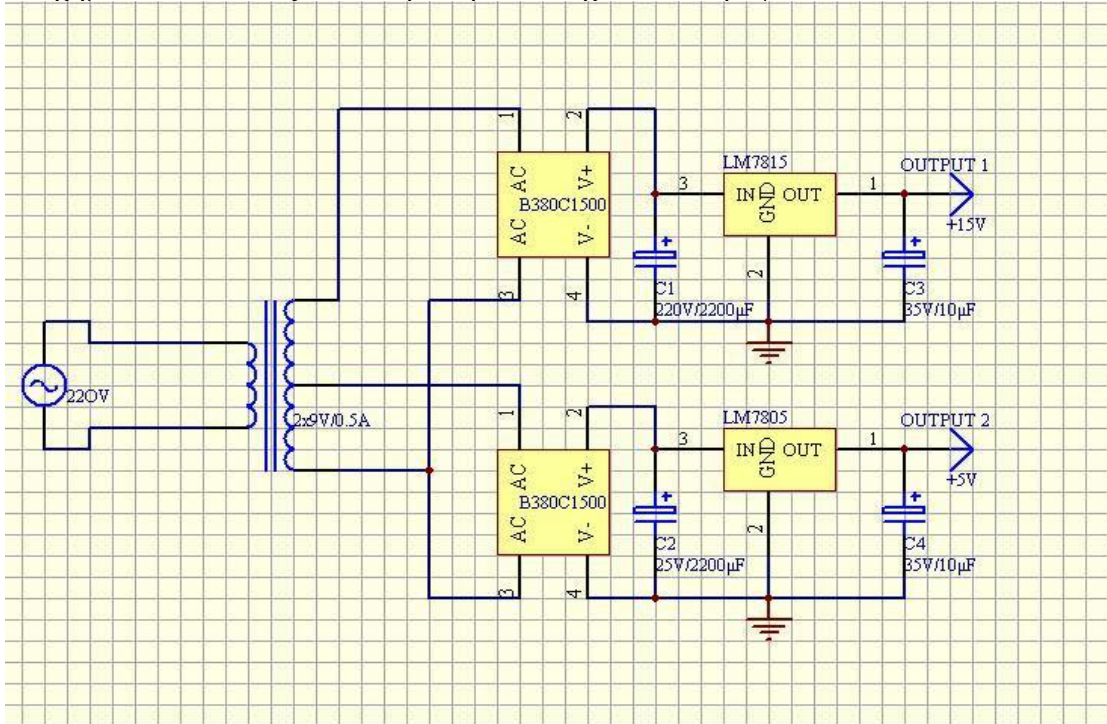
#### 1.4 Κύκλωμα Ελέγχου Κίνησης

Το κύκλωμα διακρίνεται σε τρεις βαθμίδες:

1. Τροφοδοτικό,
2. Βαθμίδα επικοινωνίας με τον Η/Υ - κύκλωμα του μικροελεγκτή,
3. Βαθμίδα οδήγησης των κινητήρων.

##### 1. Τροφοδοτικό

Στο Σχήμα 1.7 απεικονίζεται το ηλεκτρονικό σχέδιο του τροφοδοτικού.



Σχήμα 1.7: Σχηματικό διάγραμμα τροφοδοτικού του ελεγκτή κίνησης.

Πίνακας 1.2 : Χρησιμοποιηθέντα εξαρτήματα τροφοδοτικού.

A/A	Περιγραφή	Κωδικός	Χαρακτηριστικό Μέγεθος
1	Μετασχηματιστής	-	
2	Ηλεκτρολυτικός πυκνωτής C1	-	220 V / 2200 µf
3	Ηλεκτρολυτικός πυκνωτής C2	-	25 V / 2200 µf
4	Ηλεκτρολυτικοί πυκνωτές C3-4	-	35 V / 10 µf
5	Γέφυρες διόδων	B380C1500	-
6	Σταθεροποιητής 5 Volt	LM7805	-
7	Σταθεροποιητής 15 Volt	LM7815	-

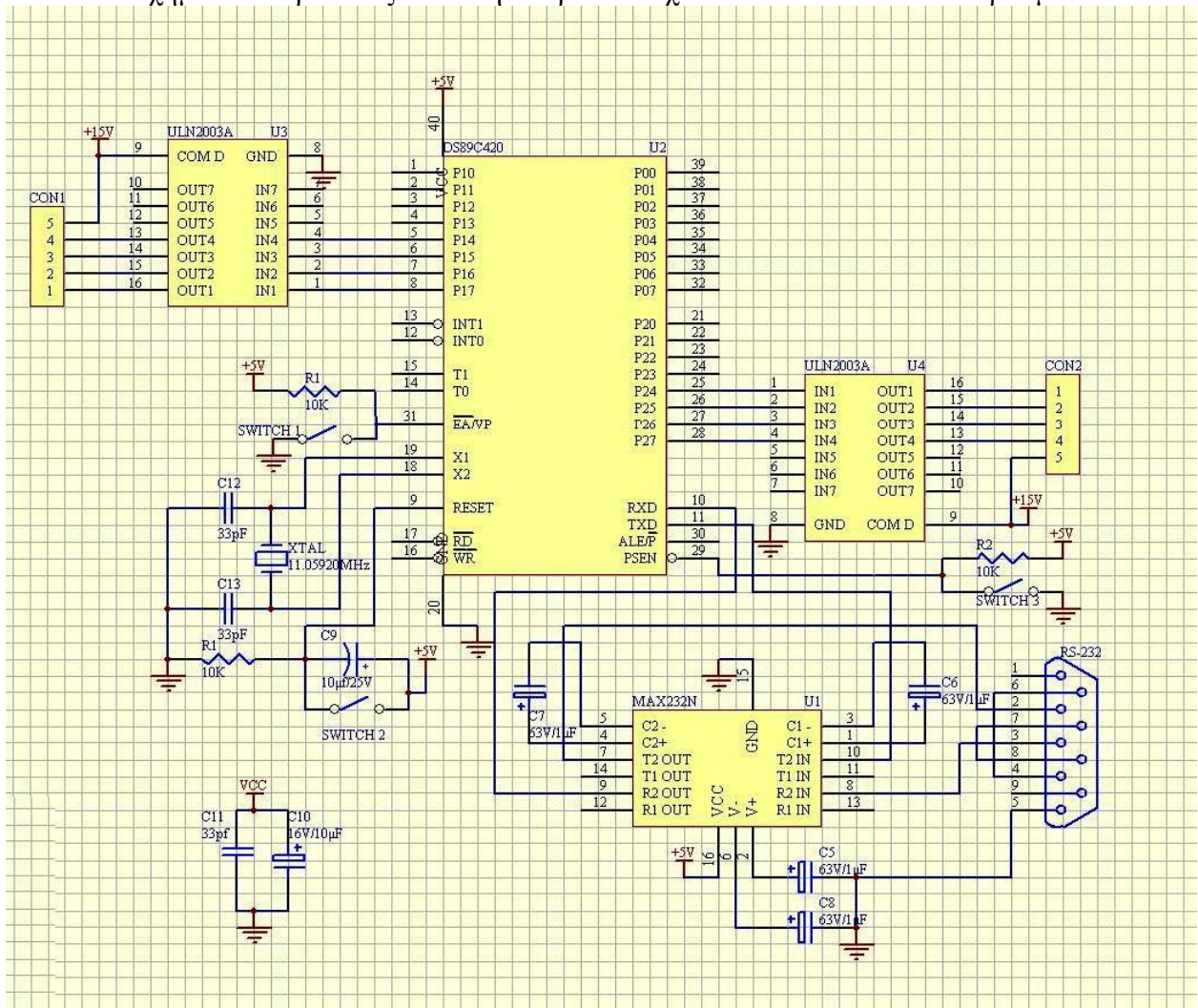
Για την τροφοδοσία του κυκλώματος απαιτούνται δύο διαφορετικές τιμές τάσης. Η πρώτη τάση έχει τιμή 5 Volt και εφαρμόζεται στο κύκλωμα του μικροελεγκτή και η δεύτερη κυμαίνεται από 12-24 Volt και εφαρμόζεται στο κύκλωμα οδήγησης των κινητήρων. Για το κύκλωμα οδήγησης των κινητήρων επιλέχθηκε τάση 15 Volt. Προκειμένου να αποτραπεί η επιβάρυνση των σταθεροποιητών, χρησιμοποιήθηκε μετασχηματιστής μέσης λήψης.

Η μισή μετασχηματισμένη τάση (9 Volt AC) ανορθώνεται πλήρως από την “κάτω” γέφυρα διόδων, B380C1500 και εξομαλύνεται από τον ηλεκτρολυτικό πυκνωτή C2, πριν σταθεροποιηθεί από το σταθεροποιητή τάσης LM7805 στα 5 Volt. Τέλος ο πυκνωτής C4 γειώνει ενδεχόμενο θόρυβο από το σταθεροποιητή και τελικά στην έξοδο, OUTPUT2, παρέχεται τάση 5 Volt DC.

Ανάλογη είναι η λειτουργία για την έξοδο OUTPUT1, με τη διαφορά ότι στην αντίστοιχη γέφυρα εφαρμόζεται ολόκληρη η μετασχηματισμένη τάση, 19 Volt AC. Στην συγκεκριμένη περίπτωση χρησιμοποιήθηκε ο σταθεροποιητής LM7815, ο οποίος σταθεροποιεί την ανορθωμένη (από τη γέφυρα), και εξομαλυμένη (από τον πυκνωτή C1), τάση, στα 15 Volt DC. Ο πυκνωτής C3 λειτουργεί αντιστοίχως με τον πυκνωτή C4.

## 2.3 Κύκλωμα Μικροελεγκτή-Βαθμίδα Οδήγησης των Κινητήρων

Στο Σχήμα 1.8 παρουσιάζεται το ηλεκτρονικό σχέδιο των δύο τελευταίων βαθμίδων.



Σχήμα 1.8: Σχηματικό διάγραμμα κυκλώματος μικροελεγκτή και βαθμίδας οδήγησης κινητήρων.

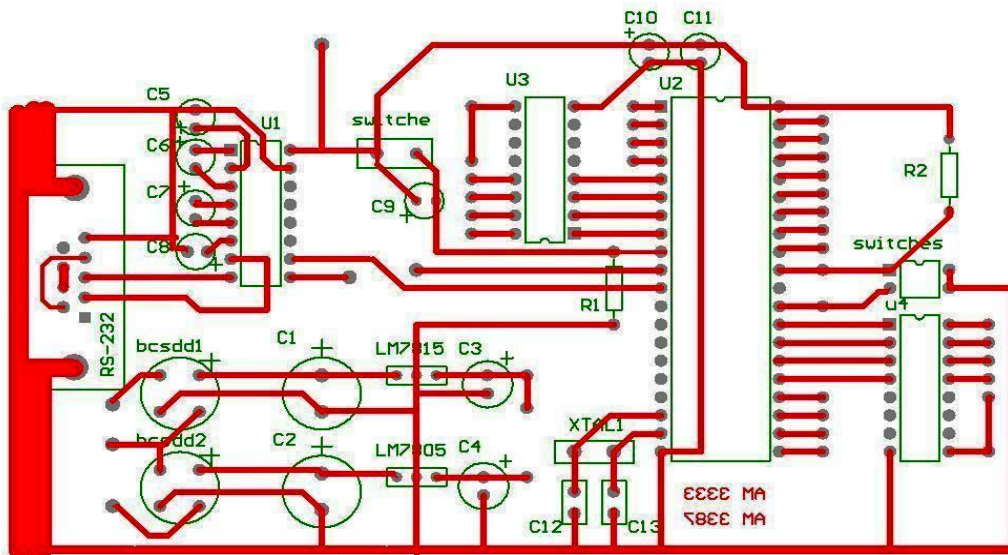
Πίνακας 1.3 : Χρησιμοποιούμενα εξαρτήματα DSP μικροελεγκτή & οδηγών κινητήρων.

A/A	Περιγραφή	Κωδικός	Χαρακτηριστικό Μέγεθος
1	Μικροελεγκτής	DS89C420	-
2	Διαμορφωτής τάσης	MAX232N	-
3	Οδηγός κινητήρων	ULN2003	-
4	Ταλαντωτής(κρύσταλλος)	XTAL	11.0592 MHz
5	Αντιστάσεις R1-R2	-	10 KOhm
6	Ηλεκτρολυτικοί πυκνωτές C5-C8	-	63 V/1 μf
7	Ηλεκτρολυτικός πυκνωτής C9	-	25 V/10 μf
8	Ηλεκτρολυτικός Πυκνωτής C10	-	16V/10μf
9	Πυκνωτές C11-C13	-	33pf

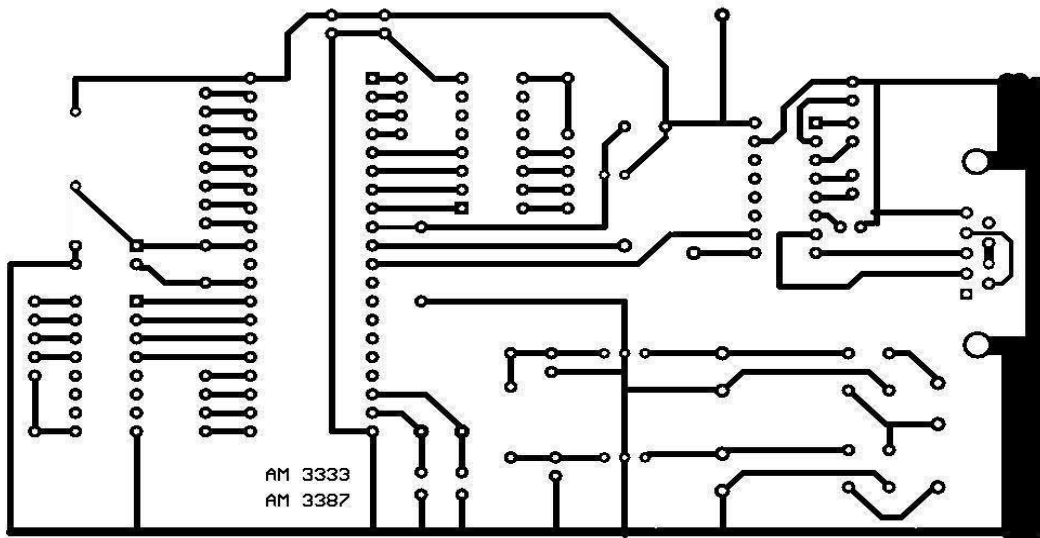
Η βαθμίδα οδήγησης των κινητήρων αποτελείται από ένα ULN2003(U3-U4) για κάθε κινητήρα. Το ULN2003 περιέχει ψηφιακούς διακόπτες και τροφοδοτείται με τάση 15Volt. Όταν εφαρμοστεί τάση σε κάποια από τις εισόδους του, κλείνει ο διακόπτης και μεταφέρονται τα 15 Volt στην αντίστοιχη έξοδο. Το ένα ULN2003 συνδέθηκε με τη θύρα (Port) 1 και το άλλο με τη θύρα (Port) 2 του μικροελεγκτή.

Ο χρησιμοποιούμενος μικροελεγκτής είναι τύπου 8051, DS89C420(U2) της εταιρείας Microelectronics. Η επικοινωνία πραγματοποιείται μέσω του ολοκληρωμένου MAX-232(U1) το οποίο μετατρέπει τα σειριακά δεδομένα που λαμβάνει από τον H/Y, από -3 Volt “OFF” και 3 Volt “ON”, σε σειριακά δεδομένα 0 Volt “OFF” και 5 Volt “ON”, τα οποία μεταδίδονται στον μικροελεγκτή. Ο διακόπτης switch-2 κάνει reset στον μικροελεγκτή, ενώ οι switch-1 και switch-3, όταν είναι κλειστοί, ο DS89C420 μπαίνει σε κατάσταση προγραμματισμού, συνεπώς με το ίδιο κύκλωμα δύναται να επαναπρογραμματιστεί ο μικροελεγκτής.

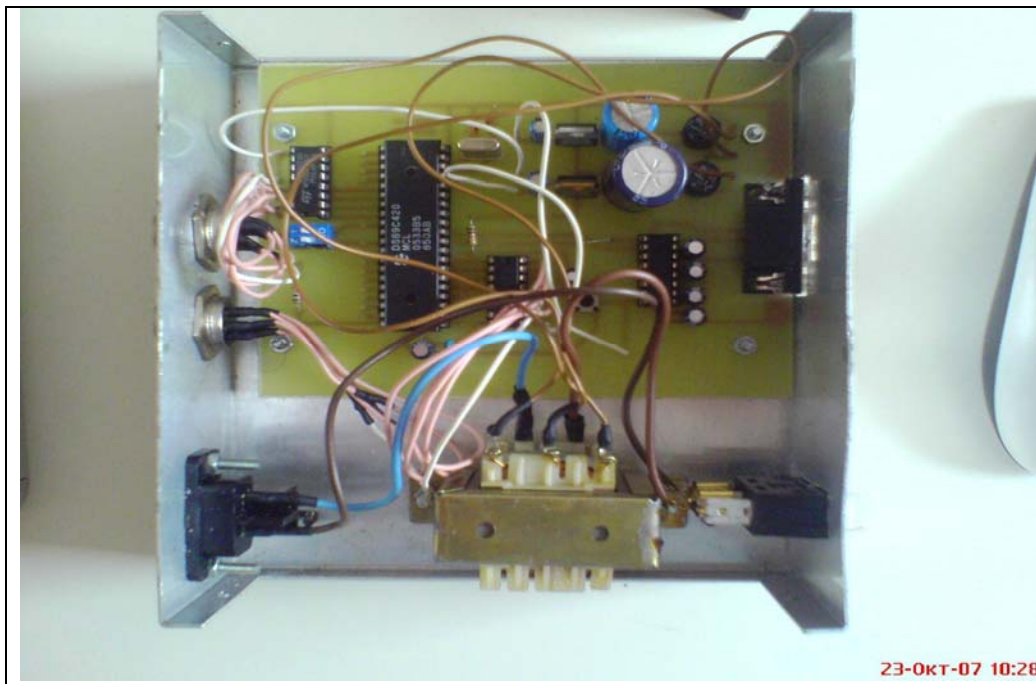
Στα Σχήματα 1.9, 1.10 και 1.11 εμφανίζονται το PCB της πλακέτας με τα εξαρτήματα, οι αγωγοί, και η πλακέτα συνδεδεμένη και τοποθετημένη σε κουτί, αντίστοιχα.



Σχήμα 1.9: PCB της πλακέτας με τα εξαρτήματα.



Σχήμα 1.10: Σχέδιο των αγωγών της πλακέτας.



Σχήμα 1.11: Η πλακέτα συνδεδεμένη και τοποθετημένη στο κουτί.

## 1.5 Οι Ρουτίνες Ελέγχου στο Matlab

Για τον εύκολο κινηματικό έλεγχο της ψηφιακής κάμερας δημιουργήθηκαν πέντε συναρτήσεις (functions/m-files) στο Matlab. Στη συνέχεια ακολουθεί σύντομη περιγραφή τους (ο κώδικας των προαναφερθέντων αρχείων παρουσιάζεται στο Παράρτημα Β').

- **scnroom.m [scan room]**

Χρησιμοποιείται για τη ρύθμιση των παραμέτρων της κάμερας. Μετά το τέλος των ρυθμίσεων σαρώνει το χώρο στον οποίο είναι τοποθετημένη η κατασκευή και εμφανίζει σε ένα παράθυρο εννέα φωτογραφίες από ισάριθμα προεπιλεγμένα σημεία του χώρου. Περισσότερες λεπτομέρειες για τον τρόπο χρήσης του, όπως και για κάθε M-file, αναπτύσσονται στο επόμενο κεφάλαιο.

- **setopos.m [set 0 position]**

M-file καλούμενο από το χρήστη με το οποίο μεταφέρονται δεδομένα από τον H/Y στον μικροελεγκτή, για να δηλώσουν σε αυτόν ότι η κάμερα βρίσκεται στη θέση "0".

- **movecam(γωνία1,γωνία2) [move camera]**

Η movecam.m είναι συνάρτηση (function), η οποία έχει δύο ορίσματα εισόδου και σκοπός της είναι να περιστρέφει την κάμερα στους δύο άξονες κίνησης (κατακόρυφος και εγκάρσιος).

Τα ορίσματα εισόδου "γωνία1" και "γωνία2" είναι οι γωνίες κίνησης για τον κατακόρυφο και εγκάρσιο άξονα αντιστοίχως, σε μοίρες. Για αρνητικές τιμές της πρώτης γωνίας, η κάμερα θα κινηθεί αριστερά, ενώ για θετικές τιμές κινείται δεξιά. Αντιστοίχως, αρνητικές τιμές της γωνίας 2 στρέφουν την κάμερα προς τα κάτω και θετικές τα πάνω.

- **goto(γωνία1,γωνία2) [go to]**

Η goto.m είναι μία συνάρτηση (function) η οποία έχει δύο ορίσματα εισόδου και σκοπός της είναι να επιτρέπει στο χρήστη τον έλεγχο της θέσης της κάμερας.

Τα ορίσματα εισόδου "γωνία1" και "γωνία2", καθορίζουν τις συντεταγμένες (σε μοίρες) της κάμερας. Η γωνία1 είναι η γωνία της κάμερας σε σχέση με τη θέση μηδέν στον οριζόντιο άξονα. Επομένως για αρνητικές τιμές της γωνίας1 η κάμερα θα μετακινηθεί σε θέση αριστερά της θέσης μηδέν ενώ για θετικές τιμές, δεξιά. Αντίστοιχα η γωνία2 έχει το ρόλο της τεταγμένης και για αρνητικές τιμές αυτής, η κάμερα θα στραφεί προς τα κάτω από τη θέση μηδέν και για θετικές τιμές, προς τα επάνω από τη θέση μηδέν.

- **nomove.m [no move]**

Το συγκεκριμένο M-file καλείται από τη συνάρτηση movecam.m στην περίπτωση που τα ορίσματα εισόδου της movecam είναι εκτός εύρους κίνησης της κάμερας και πληροφορεί τον μικροελεγκτή ότι δεν απαιτείται καμία ενέργεια.



### 1.6 Οικονομοτεχνική Μελέτη

Στον Πίνακα 1.4 παρουσιάζεται το κόστος των εξαρτημάτων που χρησιμοποιήθηκαν και το συνολικό κόστος των εξαρτημάτων της κατασκευής της πτυχιακής εργασίας.

**Πίνακας 1.4:** Κόστος χρησιμοποιηθέντων εξαρτημάτων.

<b>ΕΞΑΡΤΗΜΑΤΑ</b>	<b>ΤΙΜΗ(ευρώ)/μονάδα</b>	<b>ΠΟΣΟΤΗΤΑ</b>	<b>ΣΥΝΟΛΟ</b>
<b>ΒΗΜΑΤΙΚΟΙ ΚΙΝΗΤΗΡΕΣ</b>	20	2	40
<b>ΚΑΜΕΡΑ</b>	15	1	15
<b>ΚΟΥΤΙ ΚΑΜΕΡΑΣ</b>	15	1	15
<b>ΠΡΟΣΤΑΤΕΥΤΙΚΟ ΤΖΑΜΙ</b>	70	1	70
<b>ΒΑΣΗ ΚΑΜΕΡΑΣ (ΣΩΛΗΝΑ)</b>	20	1	20
<b>ΠΥΚΝΩΤΕΣ</b>	0.25	13	3.25
<b>ΑΝΤΙΣΤΑΣΕΙΣ</b>	0.20	2	0.40
<b>ULN2003</b>	1.50	2	3.0
<b>MAX232</b>	1.45	1	1.45
<b>DS89C420</b>	13.5	1	13.5
<b>SWITCHES DIP-2</b>	0.85	1	0.85
<b>ΒΙΣΜΑ DIN-5</b>	0.65	4	2.6
<b>ΚΑΛΩΔΙΟ DIN-5</b>	1.0	2	2.0
<b>ΘΥΡΑ RS-232</b>	0.55	1	0.55
<b>ΚΑΛΩΔΙΟ RS-232</b>	2.5	1	2.5
<b>ΚΟΥΤΙ ΠΛΑΚΕΤΑΣ</b>	10	1	10
<b>ΚΕΝΤΡΙΚΟΣ ΔΙΑΚΟΠΤΗΣ</b>	0.35	1	0.35
<b>ΣΥΝΟΛΙΚΟ ΚΟΣΤΟΣ ΚΑΤΑΣΚΕΥΗΣ : 200.45 ευρώ</b>			

## 2. ΕΓΧΕΙΡΙΔΙΟ ΧΡΗΣΤΗ

Τοποθετήστε αρχικά την κάμερα σε ένα σημείο του χώρου που επιθυμείτε.

### 2.1 Εγκατάσταση και Διασύνδεση.

Είδη καλωδίων:

- Ένα (1) καλώδιο τροφοδοσίας, Σχήμα 2.1,
- Δύο (2) καλώδια DIN-5, Σχήμα 2.2,
- Ένα (1) καλώδιο RS-232, Σχήμα 2.3.



Σχήμα 2.1 : Καλώδιο τροφοδοσίας κατασκευής.



Σχήμα 2.2: Καλώδια DIN-5.



Σχήμα 2.3: Καλώδιο RS-232.

## ΔΙΑΣΥΝΔΕΣΗ

Τοποθετήστε το καλώδιο τροφοδοσίας στην υποδοχή στο πίσω μέρος του κουτιού της κατασκευής, Σχήμα 2.4.



**Σχήμα 2.4:** Η πίσω όψη του κουτιού της κατασκευής.

Τοποθετήστε την μια πλευρά του ενός καλωδίου DIN-5 στην υποδοχή "UP" που βρίσκεται αριστερά από την υποδοχή της τροφοδοσίας και την άλλη πλευρά του DIN-5 στην πάνω υποδοχή του κουτιού της κάμερας, όπως φαίνεται στα Σχήματα 2.5 και 2.6, παρακάτω.



**Σχήμα 2.5:** Σύνδεση του καλωδίου DIN-5 με την υποδοχή "UP" στο κουτί.

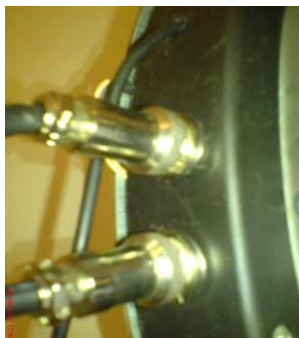


**Σχήμα 2.6 :** Σύνδεση του καλωδίου DIN-5 με την πάνω υποδοχή του βραχίονα.

Επαναλάβετε το προηγούμενο βήμα για το δεύτερο καλώδιο DIN-5 για τις υποδοχές DOWN του κουτιού του μικροελεγκτή και την κάτω υποδοχή της κάμερας.



**Σχήμα 2.7:** Σύνδεση του καλωδίου DIN-5 με την υποδοχή "DOWN" στο κουτί.



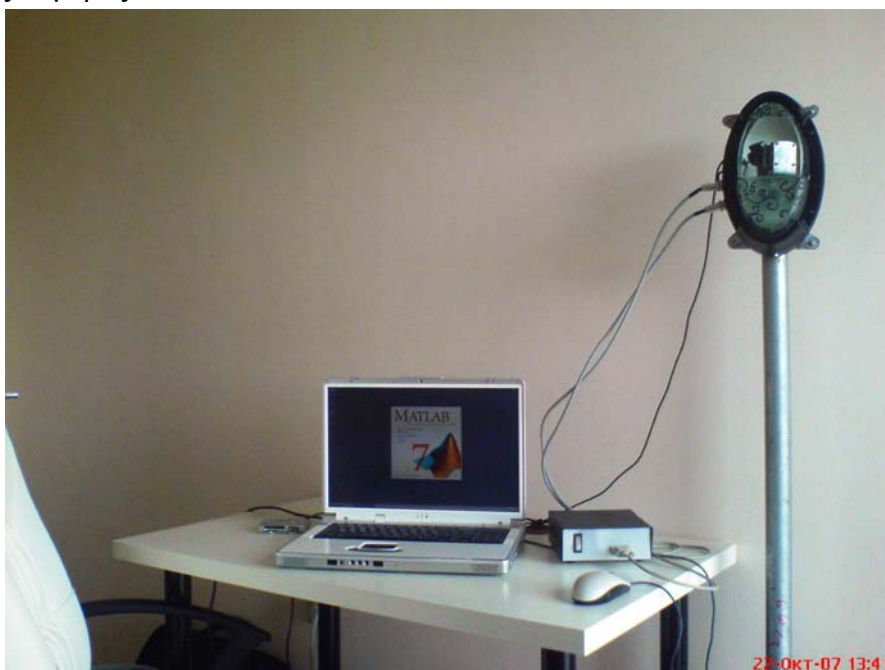
**Σχήμα 2.8:** Σύνδεση του καλωδίου DIN-5 με την κάτω υποδοχή του βραχίονα.

Στο μπροστά μέρος της κατασκευής υπάρχει ένας διακόπτης, (αριστερά) ON/OFF και δεξιά μια θύρα RS-232. Συνδέστε τη θύρα RS-232 του υπολογιστή σας με την αντίστοιχη θύρα της κατασκευής.



**Σχήμα 2.9:** Το καλώδιο RS-232 συνδεδεμένο στη θύρα της κατασκευής.

Συνδέστε το καλώδιο της τροφοδοσίας με το δίκτυο της Δ.Ε.Η. και την κάμερα σε μια θύρα USB του υπολογιστή της. Αφού βεβαιωθείτε για τις παραπάνω συνδέσεις, ανοίξτε το MATLAB και βάλτε τον διακόπτη στη θέση ON. Είστε έτοιμοι να προχωρήσετε στις ρυθμίσεις της κάμερας.



**Σχήμα 2.10:** Η κατασκευή πλήρως συνδεδεμένη.

## 2.2 Ρυθμίσεις της Κάμερας

Αφού ανοίξετε το MATLAB, τοποθετήστε το CD με τους οδηγούς της κατασκευής στον υπολογιστή σας. Εγκαταστήστε τον οδηγό της κάμερας που θα βρείτε στον φάκελο “CAM DRIVER”. Αντιγράψτε όλα τα αρχεία που περιέχει ο φάκελος MATLAB FILES, στον φάκελο work του MATLAB.

### Τοποθέτηση της κάμερας στο σημείο “0”

Η θέση “0” είναι προεπιλεγμένη ώστε η κάμερα να “βλέπει” σε απολύτως ευθεία θέση, Σχήμα 2.11.



**Σχήμα 2.11:** Ο βραχίονας τοποθετημένος στη θέση “0”.

Μπορείτε να τοποθετήσετε την κάμερα στο σημείο “0” με δύο τρόπους:

(α) Χειροκίνητα:

Βάλτε τον διακόπτη στη θέση OFF, ξεβιδώστε το πλαίσιο που συγκρατεί το προστατευτικό γυαλί της κάμερας. Βγάλτε προσεκτικά το γυαλί και γυρίστε την κάμερα στη θέση “0”, βάλτε πάλι τον διακόπτη στη θέση ON, τοποθετήστε το γυαλί και το πλαίσιο στη θέση τους και βιδώστε το πλαίσιο.

(β) Μέσω του MATLAB:

Καλέστε στο command window του matlab τη συνάρτηση movecam και βάλτε ως ορίσματα εισόδου της γωνίες που νομίζετε ότι πρέπει να κινηθεί η κάμερα έως όπου πάει στο σημείο “0”. Τρέξτε την παραπάνω συνάρτηση μέχρι η κάμερα να φτάσει στο σημείο “0”, Σχήμα 2.12.

Κατά τη χρήση της movecam θα λαμβάνετε μηνύματα μέσω του Matlab, για την θέση στη οποία βρίσκεται η κάμερα. Στο στάδιο αυτό μη λάβετε υπ’ όψη σας, τις συντεταγμένες της κάμερας στην περίπτωση που αυτή δεν έχει “μηδενιστεί”, αρχικά.

```

MATLAB
File Edit Debug Desktop Window Help
C:\MATLAB7\work
Shortcuts How to Add What's New

>> movecam(-50,30)
H CAMERA VRISKETAI STH 8ESH [-48.6,28.8 ]
>> movecam(-2,0)
H CAMERA VRISKETAI STH 8ESH [-50.4,28.8 ]
>> movecam(-5,5)
H CAMERA VRISKETAI STH 8ESH [-54.0,32.4 ]
>> |

```

**Σχήμα 2.12:** Χρήση της movecam.m για την τοποθέτηση της κάμερας στο σημείο “0”.

**Προσοχή:** Αν η συνάρτησή σας βγάλει error και δε σας αφήνει να κινήσετε την κάμερα στις γωνίες που δώσατε, τρέξτε το M-file setopos (set “0” position), για να μηδενιστούν οι μετρητές του μικροελεγκτή, και μετά συνεχίστε, Σχήμα 2.13.

```

MATLAB
File Edit Debug Desktop Window Help
C:\MATLAB7\work
Shortcuts How to Add What's New

>> movecam(-50,30)
H CAMERA VRISKETAI STH 8ESH [-48.6,28.8 ]
>> movecam(-2,0)
H CAMERA VRISKETAI STH 8ESH [-50.4,28.8 ]
>> movecam(-5,5)
H CAMERA VRISKETAI STH 8ESH [-54.0,32.4 ]
>> movecam(-40,0)
??? Error using ==> movecam
MEIΩSTE THN GWNIA ARISTERA

>> setopos|

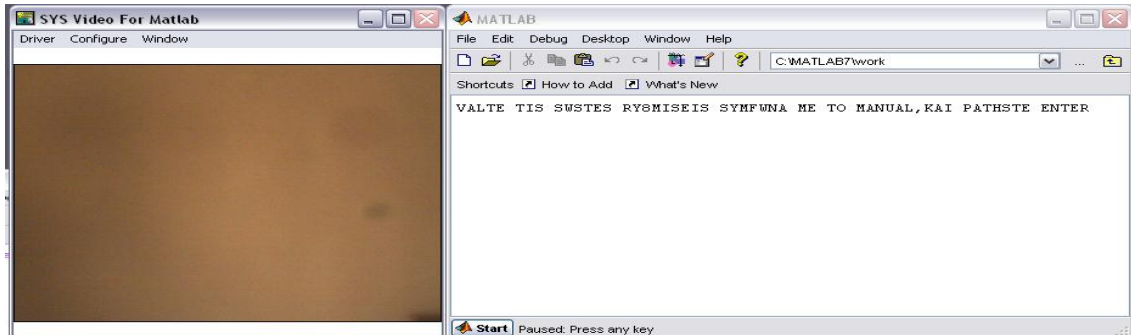
```

**Σχήμα 2.13:** Error κατά την χρήση της movecam.m για την τοποθέτηση της κάμερας στο σημείο “0”.

Ελέγξτε την θέση της κάμερας και αν βρίσκεται στη θέση “0”, τρέξτε το M-file “setopos”

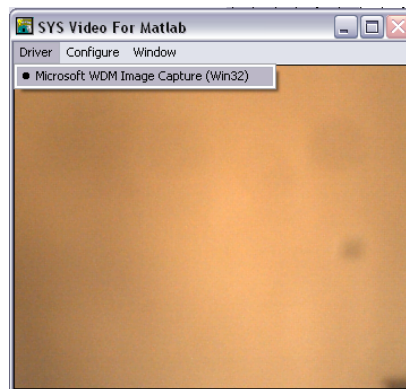
## Ρυθμίσεις του οδηγού της κάμερας και εμφάνιση φωτογραφιών των 9 επιλεγμένων σημείων

Καλέστε το M-file scnrroom (scan room). Το Matlab θα ανοίξει το παράθυρο “SYS Video For Matlab”, όπως φαίνεται στο Σχήμα 2.14.



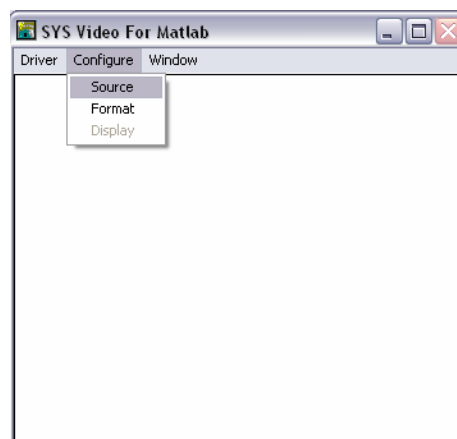
**Σχήμα 2.14:** Άνοιγμα των ρυθμίσεων της κάμερας με το scnrroom.m.

Από τη γραμμή εργαλείων του “SYS Video For Matlab” κάντε “κλικ” στο “Driver” και αμέσως μετά “κλικ” στο “Microsoft WDM Image Capture (Win32)”.



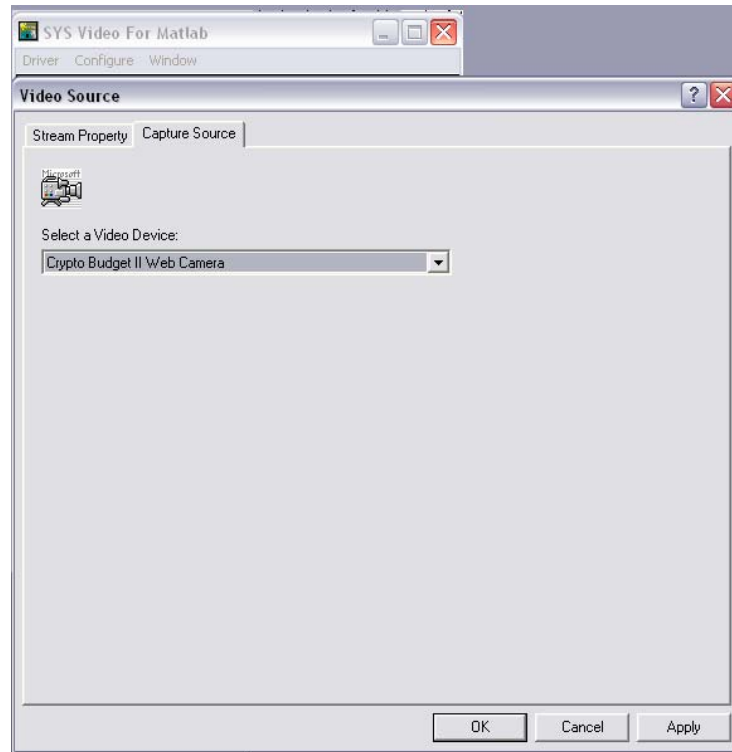
**Σχήμα 2.15:** Αρχική ρύθμιση της κάμερας.

Από το μενού “Configure” κάντε “κλικ” στο Source, Σχήμα 3.6, για να εμφανιστεί το παράθυρο Video Source.



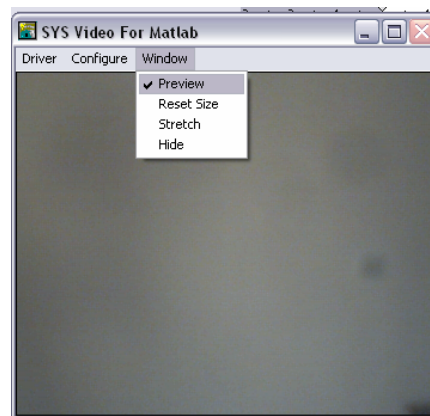
**Σχήμα 2.16:** Επιλογή του «Source» από το παράθυρο “SYS Video For Matlab”.

Από το παράθυρο Video Source επιλέξτε την καρτέλα “Capture Source”. Στη συνέχεια, στο πεδίο “Select a Video Device” επιλέξτε την “Crypto Budget II Web Camera”, Σχήμα 2.17.



**Σχήμα 2.17:** Επιλογή της ψηφιακής κάμερας του βραχίονα.

Τέλος, από το μενού Window επιλέξτε τη επιλογή Preview, Σχήμα 2.18.

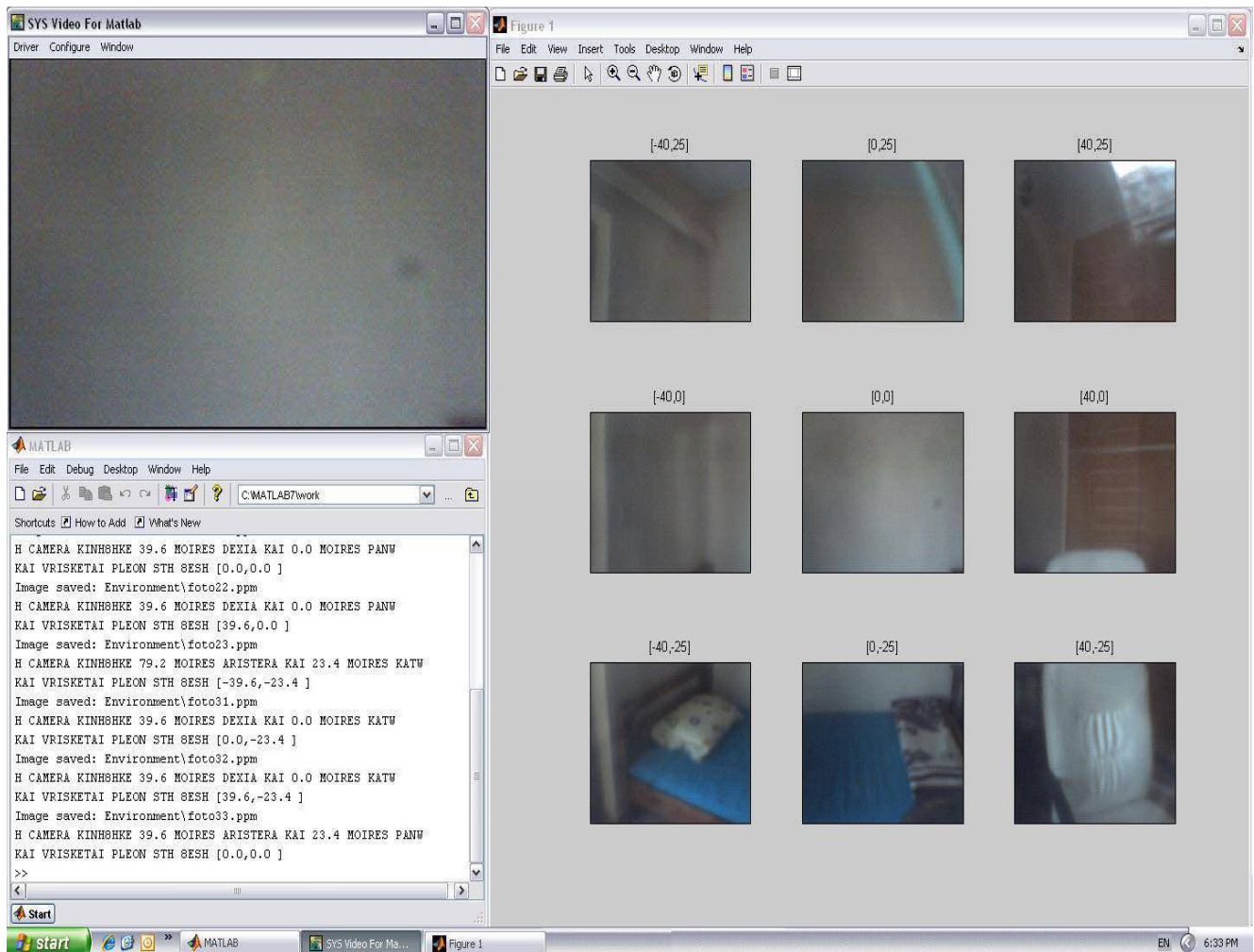


**Σχήμα 2.18:** Τελική ρύθμιση της κάμερας.

Τελειώνοντας με τις ρυθμίσεις της κάμερας, θα πρέπει αυτή να λειτουργεί από το “SYS Video For Matlab”. Πατήστε δύο φορές Enter στο Command Window, αν είστε βέβαιοι για τις παραπάνω ρυθμίσεις, ώστε να συνεχίσετε.



Όταν τελειώσει το scnroom θα έχετε στην οθόνη σας το “command window”, το “SYS Video For Matlab” και ένα γράφημα με τις εννέα φωτογραφίες των προεπιλεγμένων σημείων του χώρου, Σχήμα 2.19. Τα σημεία αυτά μπορούν να αλλάξουν, αλλάζοντας τις συντεταγμένες που περιέχει ο πίνακας A και τρέχοντας εκ νέου την scnroom (χωρίς να αλλάξετε τις ρυθμίσεις).



Σχήμα 2.19: Αποτέλεσμα μετά το τέλος του scnroom.m.

## 2.3 Γνωριμία με τις συναρτήσεις

### movecam (γωνία1, γωνία2)

Η movecam κινεί την κάμερα σύμφωνα με τις γωνίες που του εισάγουμε ως ορίσματα εισόδου:

Γωνία1: γωνία σε μοίρες που θα στραφεί η κάμερα κατά τον κατακόρυφο άξονα.

Γωνία1>0 : η κάμερα θα στραφεί δεξιά.

Γωνία1<0 : η κάμερα θα στραφεί αριστερά.

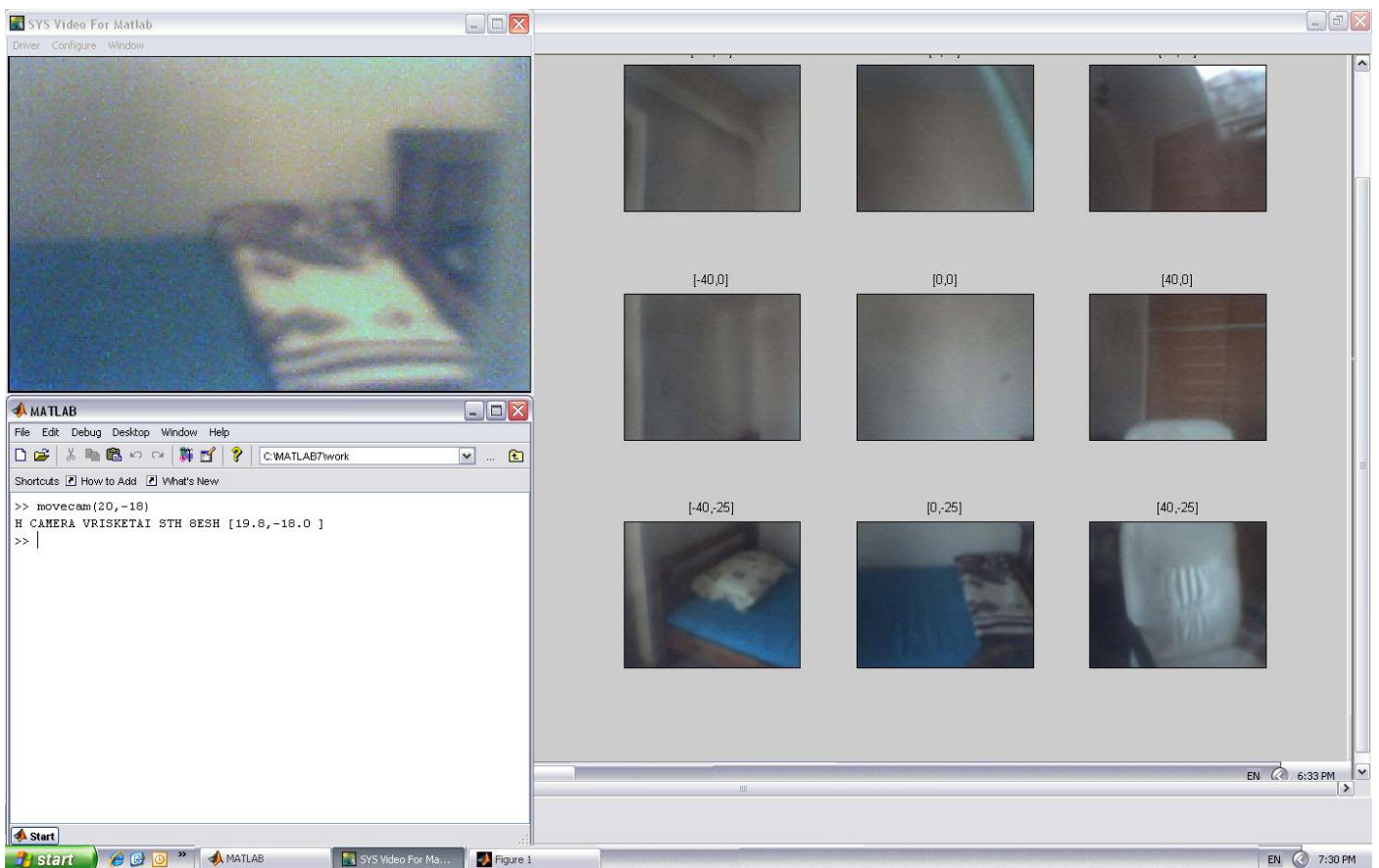
Γωνία2: γωνία σε μοίρες που θα στραφεί η κάμερα κατά τον εγκάρσιο άξονα.

Γωνία2>0 : η κάμερα θα κινηθεί πάνω.

Γωνία2<0 : η κάμερα θα κινηθεί κάτω.

Η συνάρτηση αυτή επιλύει το ευθύ κινηματικό πρόβλημα. Ο χρήστης ορίζει πόσο θα κινηθεί ο κάθε κινητήρας και επιστρέφει τις συντεταγμένες που βρίσκεται η κάμερα μετά την κίνησή της.

Η κάμερα στο Σχήμα 2.19, βρίσκεται στη θέση [0,0]. Αν τρέξετε την movecam(20,-18), η κάμερα θα αναμένετε να στραφεί 20 μοίρες δεξιά και 18 μοίρες κάτω, επειδή όμως η κατασκευή έχει τη δυνατότητα να στρέφεται σε γωνίες ακέραια πολλαπλάσια του 1.8, θα έχετε ως αποτέλεσμα το Σχήμα 2.20.



Σχήμα 2.20: Αποτέλεσμα από την χρήση της συνάρτησης movecam.m.

### goto (γωνία1,γωνία2)

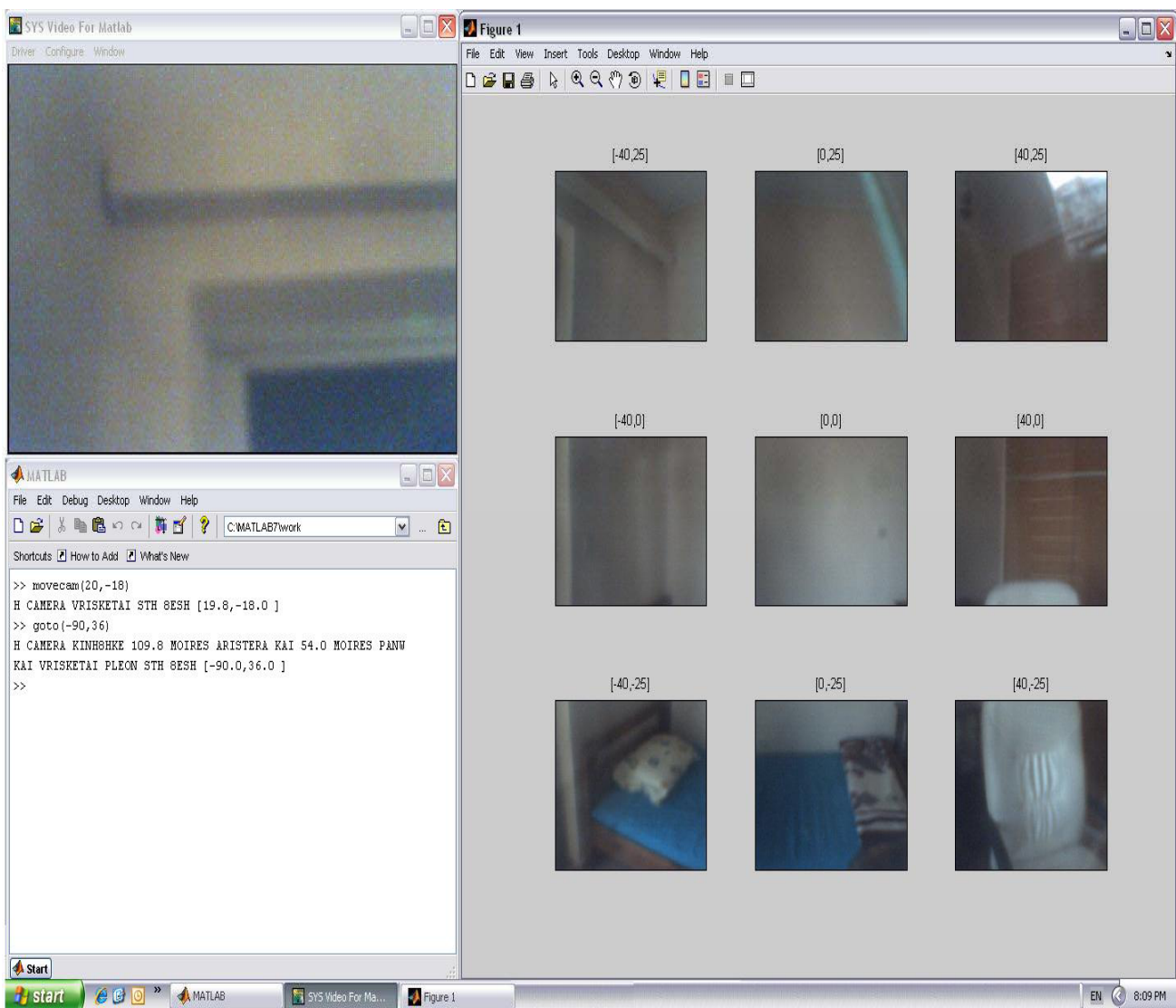
Η goto μετακινεί την κάμερα στις συγκεκριμένες συντεταγμένες που εισάγουμε ως ορίσματα εισόδου:

Γωνία1: συντεταγμένες σε μοίρες στον κατακόρυφο άξονα.

Γωνια2: συντεταγμένες σε μοίρες στον εγκάρσιο άξονα.

Η συνάρτηση αυτή, σε αντίθεση με την movecam, επιλύει το αντίστροφο κινηματικό πρόβλημα. Ο χρήστης ορίζει τη θέση (σε μοίρες) στην οποία επιθυμεί να μετακινηθεί η κάμερα και μετά την κίνηση, επιστρέφει εκτός από τις συντεταγμένες, τις γωνίες στροφής ως προς κάθε άξονα.

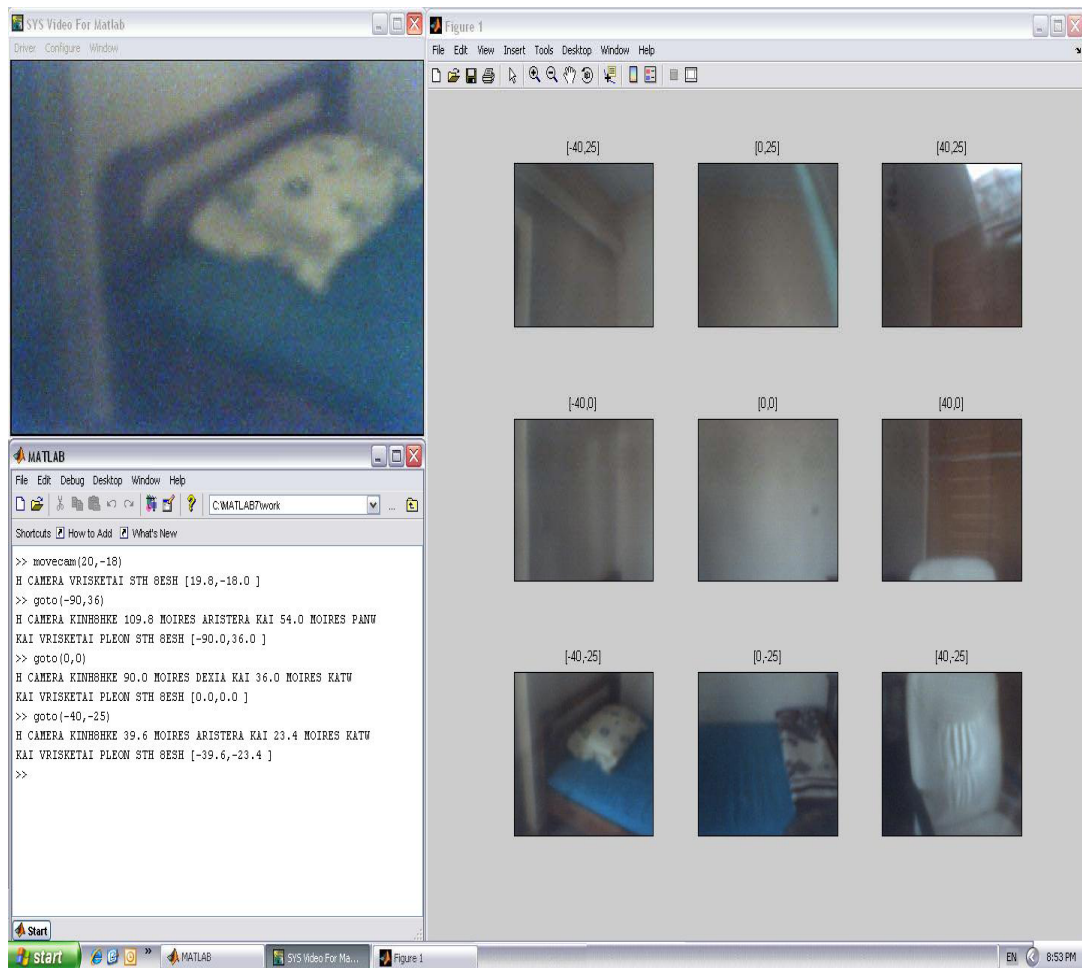
Στο Σχήμα 2.20 η κάμερα βρίσκεται στη θέση [19.8,-18.0]. Αν τρέξετε την goto(-90,36), η κάμερα θα μετακινηθεί στο σημείο που απέχει 90 μοίρες αριστερά και 36 μοίρες πάνω από το σημείο [0, 0], Σχήμα 2.21.



Σχήμα 2.21: Αποτέλεσμα από την χρήση της συνάρτησης goto.m.

Η goto παρέχει επίσης τη δυνατότητα να εκμεταλλευτείτε το figure με τις φωτογραφίες από το scnroom. Πάνω από κάθε φωτογραφία αναγράφονται οι συντεταγμένες της συγκεκριμένης θέσης, έτσι αφ' ενός έχετε καλύτερη αντίληψη του χώρου που θέλετε να εστιάσετε, αφ' ετέρου δε, με την goto μπορείτε να εστιάσετε σημαντικά για εσάς σημεία, χωρίς να ψάχνετε συντεταγμένες ή γωνίες κίνησης.

Στο Σχήμα 2.21 η κάμερα βρίσκεται στο σημείο  $[-90.0,36.0]$ . Αν θέλετε να κοιτάξετε στη φωτογραφία που βρίσκεται κάτω αριστερά στο figure, αρκεί να τρέξετε την goto με τις συντεταγμένες της συγκεκριμένης φωτογραφίας, Σχήμα 2.22.



Σχήμα 2.22: Η χρήση της goto.m εκμεταλλευόμενοι τις φωτογραφίες του figure.

**Προσοχή:** πριν κλείσετε το Matlab τρέξτε στο command window την goto(0,0) για να μην χρειαστεί την επόμενη φορά να τοποθετήσετε πάλι την κάμερα στο σημείο "0".

## ΣΥΜΠΕΡΑΣΜΑΤΑ

Στην παρούσα εργασία σχεδιάστηκε και κατασκευάστηκε ένα σύστημα που επιτρέπει στο χρήστη να χειρίζεται μια ψηφιακή κάμερα και να εποπτεύει ένα δεδομένο χώρο μέσω ηλεκτρονικού υπολογιστή. Η κατασκευή δύναται να χρησιμοποιηθεί για την πλήρη εποπτεία χώρου από απόσταση, με τη δυνατότητα της χρήσης μιας μόνο κάμερας. Επίσης, ως εφαρμογή της ρομποτικής μπορεί να χρησιμοποιηθεί από τους σπουδαστές του Τ.Ε.Ι Κρήτης στα πλαίσια του Εργαστηριακού Μαθήματος της ρομποτικής, για τη βέλτιστη κατανόηση των πρακτικών εφαρμογών της ρομποτικής.

Όπως κάθε κατασκευή, έτσι και η συγκεκριμένη αντιμετωπίζει περιορισμούς που οφείλονται στα χρησιμοποιούμενα υλικά. Ένας από τους περιορισμούς είναι το περιορισμένο εύρος κίνησης, το οποίο οφείλεται στο πλαίσιο όπου έχει τοποθετηθεί ο βραχίονας κίνησης της κάμερας (κατασκευαστικού περιορισμοί). Επίσης χρησιμοποιήθηκε USB κάμερα για την ελάττωση του κόστους της κατασκευής, με συνέπεια να υπάρχει περιορισμός μεταξύ του σταθμού ελέγχου του συστήματος και της κάμερας.

Οι παραπάνω περιορισμοί αντιμετωπίζονται με ευκολία με απλή αντικατάσταση των υλικών που αναφέραμε. Έτσι τοποθετώντας το βραχίονα κίνησης σε πλαίσιο το οποίο περιβάλλεται εξ ολοκλήρου από διαφανές υλικό και χρησιμοποιώντας ασύρματη κάμερα, το εύρος κίνησης της κάμερας γίνεται απεριόριστο. Με αύξηση του συνολικού κόστους, μπορεί να αντικατασταθεί η USB κάμερα με μια IP ασύρματη, με την οποία επιλύεται το πρόβλημα της απόστασης από το σταθμό ελέγχου.

Ο βραχίονας δύναται να επεκταθεί σε βραχίονα 3<sup>ων</sup> βαθμών ελευθερίας, με την κατάλληλη προσαρμογή ενός ακόμα βηματικού κινητήρα. Το κύκλωμα ελέγχου της κάμερας έχει τη δυνατότητα οδήγησης ενός ακόμα κινητήρα, αφού δε χρησιμοποιείται μια θύρα του μικροελεγκτή. Για την υλοποίηση της παραπάνω βελτίωσης είναι απαραίτητη η διαμόρφωση του κώδικα assembly για τη χρήση της θύρας, όπως και η αλλαγή των παραμέτρων των συναρτήσεων του Matlab.

Οι επεκτάσεις και οι βελτιώσεις της παρούσας πτυχιακής εργασίας, πέραν των προαναφερθέντων, υπόκεινται σε αντικειμενικούς παράγοντες όπως τα τεχνικά χαρακτηριστικά, οι επιδόσεις, η εμφάνιση και η χρηστικότητα σε σχέση με το κόστος κατασκευής που έχει ιδιαίτερο ενδιαφέρον όταν αυτό εξετάζεται σε επίπεδο οικονομίας κλίμακας.

## **ΠΑΡΑΡΤΗΜΑ Α΄**

## Κώδικας ASSEMBLY του Μικροελεγκτή

```
1.      DEFSEG START,ABSOLUTE
2.      SEG START
3.      ORG 0000H
4.      SJMP MAIN
5.
6.
7.
8.  MAIN:MOV    P1,#10001000B    ;P1<-#10001000B
9.      MOV     P2,#10001000B    ;P2<-#10001000B
10.     MOV     43H,#00H        ;43H<-#00H
11.     MOV     44H,#00H        ;43H<-#00H
12.
13.
14.     MOV     TMOD,#20h        ;TMOD<-#20H (Timer1 ->Mode 2)
15.     MOV     TL1,#-3         ;TL1<- #-3
16.     MOV     TH1,#-3         ;TH1<- #-3 (B.D.=9600)
17.     MOV     SCON,#50h       ;SCON<-#50H
18.     SETB    TR1             ;TR1<-1
19.
20.
21.  WtIns: ACALL GtIns          ;Κάλεσε τη ρουτίνα "GtIns"
22.     MOV     A,39H           ;A<- 39H
23.     CJNE    A,#0FFH,SETOP    ;Αν ο A δεν είναι ίσος με 255 κάνει άλμα
                                ;στο "SETOP"
24.     ACALL   SNDPOS          ;Κάλεσε τη ρουτίνα "SNDPOS"
25.     ACALL   GtMvs          ;Κάλεσε τη ρουτίνα "GtMvs"
26.     MOV     A,40H           ;A<- 40H
27.     JNB     ACC.3,Chck2     ;αν το bit τρία του A είναι '0' κάνει
                                ;άλμα στο Chck2
28.     ACALL   CntMtr1        ;Κάλεσε τη ρουτίνα "CntMtr1"
29.
30.
31.  Chck2: MOV     A,40H         ;A<- 40H
32.     JNB     Acc.7,Chck3     ;αν το bit επτά του A είναι '0' κάνει
                                ;άλμα στο Chck3
33.     ACALL   CntMtr2        ;Κάλεσε τη ρουτίνα "CntMtr2"
34.  Chck3: ACALL   SNDPOS          ;Κάλεσε τη ρουτίνα "SNDPOS"
35.     LJMP    WtIns          ;Κάνε άλμα στο "WtIns"
36.
37.
38.  SETOP: MOV    43H,#00H      ;43H<- #00H
39.     MOV     44H,#00H        ;44H<- #00H
40.     LJMP    WtIns          ;Κάνε άλμα στο "WtIns"
41.
42.
43.
44.  GtIns: CLR     RI           ;RI<- '0'
45.     JNB     RI,$           ;Περίμενε μέχρι να γίνει '1' ο RI
46.     MOV     39H,SBUF        ;αποθήκευσε τα δεδομένα από την RS-323
                                ;στο 39H
47.     CLR     RI           ;RI<- '0'
48.     RET                     ;RETURN
49.
50.  GtMvs: CLR     RI           ;RI<- '0'
51.     JNB     RI,$           ;Περίμενε μέχρι να γίνει '1' ο RI
52.     MOV     40H,SBUF        ;αποθήκευσε τα δεδομένα από την RS-323
                                ;στο 40H
53.     CLR     RI           ;RI<- '0'
54.
```

```

55.      JNB   RI,$           ;Περίμενε μέχρι να γίνει '1' ο RI
56.      MOV   41H,SBUF      ;αποθήκευσε τα δεδομένα από την RS-323
                               ;στο 41H
57.      CLR   RI           ;RI<- '0'
58.
59.      JNB   RI,$           ;Περίμενε μέχρι να γίνει '1' ο RI
60.      MOV   42H,SBUF      ;αποθήκευσε τα δεδομένα από την RS-323
                               ;στο 42H
61.      CLR   RI           ;RI<- '0'
62.      RET                    ;RETURN
63.
64.
65. CntMtr1: MOV   A,40H      ;A<- 40H
66.      JNB   ACC.2,ST12    ;αν το bit δύο του A είναι '0' κάνε
                               ;άλμα στο "ST12"
67. ST11:  ACALL DELAY       ;κάλεσε τη ρουτίνα " DELAY "
68.      MOV   A,P1         ;A<- P1
69.      RL    A            ;κάνε στον A μια ολίσθηση αριστερά
70.      MOV   P1,A         ;A<- P1
71.      DEC   44H         ;44H<- 44H -1
72.      DJNZ  41H,ST11     ;μείωσε τον 41H κατά ένα και αν δεν
                               ;είναι '0' κάνε άλμα στο "ST11"
73.      RET                    ;RETURN
74.
75. ST12:  ACALL DELAY       ;κάλεσε τη ρουτίνα " DELAY "
76.      MOV   A,P1         ;A<- P1
77.      RR    A            ;κάνε στον A μια ολίσθηση δεξιά
78.      MOV   P1,A         ;A<- P1
79.      INC   44H         ;44H<- 44H +1
80.      DJNZ  41H,ST12     ;μείωσε τον 41H κατά ένα και αν δεν
                               ;είναι '0' κάνε άλμα στο "ST12"
81.      RET                    ;RETURN
82.
83.
84.
85.
86. CntMtr2: MOV   A,40H
87.      JNB   ACC.6,ST22    ;αν το bit έξι του A είναι '0' κάνε
                               ;άλμα στο "ST22"
88. ST21:  ACALL DELAY       ;κάλεσε τη ρουτίνα " DELAY "
89.      MOV   A,P2         ;A<- P2
90.      RL    A            ;κάνε στον A μια ολίσθηση αριστερά
91.      MOV   P2,A         ;A<- P2
92.      DEC   43H         ;43H<- 43H -1
93.      DJNZ  42H,ST21     ;μείωσε τον 42H κατά ένα και αν δεν
                               ;είναι '0' κάνε άλμα στο "ST21"
94.      RET                    ;RETURN
95.
96.
97. ST22:  ACALL DELAY       ;κάλεσε τη ρουτίνα " DELAY "
98.      MOV   A,P2         ;A<- P2
99.      RR    A            ;κάνε στον A μια ολίσθηση δεξιά
100.     MOV   P2,A         ;A<- P2
101.     INC   43H         ;43H<- 43H +1
102.     DJNZ  42H,ST22     ;μείωσε τον 42H κατά ένα και αν δεν
                               ;είναι '0' κάνε άλμα στο "ST22"
103.     RET                    ;RETURN
104.
105. SNDPOS: ACALL INVRT      ;κάλεσε τη ρουτίνα "INVRT "
106.     CLR   TI           ;TI<- '0'
107.     MOV   SBUF,53H     ;στείλε στην RS-232 τα δεδομένα του 53H

```



```

108.      JNB    TI,$           ;Περίμενε μέχρι να γίνει '1' ο TI
109.      CLR    TI           ;TI<- '0'
110.      MOV    SBUF,54H     ;στείλε στην RS-232 τα δεδομένα του 54H
111.      JNB    TI,$           ;Περίμενε μέχρι να γίνει '1' ο TI
112.      CLR    TI           ;TI<- '0'
113.      MOV    SBUF,55H     ;στείλε στην RS-232 τα δεδομένα του 55H
114.      JNB    TI,$           ;Περίμενε μέχρι να γίνει '1' ο TI
115.      CLR    TI           ;TI<- '0'
116.      MOV    SBUF,56H     ;στείλε στην RS-232 τα δεδομένα του 56H
117.      JNB    TI,$           ;Περίμενε μέχρι να γίνει '1' ο TI
118.      CLR    TI           ;TI<- '0'
119.      RET                    ;RETURN
120.
121.DELAY:
122.      MOV    R6,#200      ;R6<- #200
123.L1:   MOV    R7,#250      ;R7<- #250
124.      DJNZ  R7,$           ;μείωνε τον R7 κατά ένα μέχρι να
125.      ;μηδενιστεί
126.      DJNZ  R6,L1         ;μείωσε τον R6 κατά ένα και αν δεν
127.      ;είναι '0' κάνει άλμα στο "L1"
128.      RET                    ;RETURN
129.
130.INVRT: MOV    R5,#2        ;R6<- #4
131.      MOV    R0,#52H      ;R6<- #52H
132.      MOV    R1,#42H      ;R1<- #42H
133.STRT: INC    R1           ;R1<- R1 +1
134.      INC    R0           ;R0<- R0 +1
135.      MOV    A,@R1        ;μετέφερε τα δεδομένα της μνήμης που
136.      ANL    A,#0F0H      ;δείχνει ο R1, στον A
137.MN0:  CJNE  A,#00H,MN1    ;A<- A AND 11110000
138.      ;αν ο A δεν είναι ίσος με το #00H κάνει
139.      ;άλμα στο "MN1"
140.MN1:  MOV    A,#30H      ;A<- #30H
141.      LJMP  NEXT         ;κάνε άλμα στο "NEXT"
142.      CJNE  A,#10H,MN2    ;αν ο A δεν είναι ίσος με το #10H κάνει
143.      ;άλμα στο "MN2"
144.      MOV    A,#31H      ;A<- #31H
145.      LJMP  NEXT         ;κάνε άλμα στο "NEXT"
146.MN2:  CJNE  A,#20H,MN3    ;αν ο A δεν είναι ίσος με το #20H κάνει
147.      ;άλμα στο "MN3"
148.      MOV    A,#32H      ;A<- #32H
149.      LJMP  NEXT         ;κάνε άλμα στο "NEXT"
150.      CJNE  A,#30H,MN4    ;αν ο A δεν είναι ίσος με το #30H κάνει
151.      ;άλμα στο "MN4"
152.      MOV    A,#33H      ;A<- #33H
153.      LJMP  NEXT         ;κάνε άλμα στο "NEXT"
154.      CJNE  A,#40H,MN5    ;αν ο A δεν είναι ίσος με το #40H κάνει
155.      ;άλμα στο "MN5"
156.      MOV    A,#34H      ;A<- #34H
157.      LJMP  NEXT         ;κάνε άλμα στο "NEXT"
158.MN5:  CJNE  A,#50H,MN6    ;αν ο A δεν είναι ίσος με το #50H κάνει
159.      ;άλμα στο "MN6"
160.      MOV    A,#35H      ;A<- #35H
161.      LJMP  NEXT         ;κάνε άλμα στο "NEXT"
162.      CJNE  A,#60H,MN7    ;αν ο A δεν είναι ίσος με το #60H κάνει
163.      ;άλμα στο "MN7"
164.      MOV    A,#36H      ;A<- #36H
165.      LJMP  NEXT         ;κάνε άλμα στο "NEXT"
166.      CJNE  A,#70H,MN8    ;αν ο A δεν είναι ίσος με το #70H κάνει
167.      ;άλμα στο "MN8"

```

```

160.      MOV    A,#37H      ;A<- #37H
161.      LJMP   NEXT      ;κάνε άλμα στο "NEXT"
162.MN8:  CJNE  A,#80H,MN9  ;αν ο A δεν είναι ίσος με το #80H κάνει
                    ;άλμα στο "MN9"
163.      MOV    A,#38H      ;A<- #38H
164.      LJMP   NEXT      ;κάνε άλμα στο "NEXT"
167.MN9:  CJNE  A,#90H,MNA  ;αν ο A δεν είναι ίσος με το #90H κάνει
                    ;άλμα στο "MNA"
168.      MOV    A,#39H      ;A<- #39H
169.      LJMP   NEXT      ;κάνε άλμα στο "NEXT"
170.MNA:  CJNE  A,#0A0H,MNB ;αν ο A δεν είναι ίσος με το #0A0H κάνει
                    ;άλμα στο "MNB"
171.      MOV    A,#41H      ;A<- #41H
172.      LJMP   NEXT      ;κάνε άλμα στο "NEXT"
173.MNB:  CJNE  A,#0B0H,MNC ;αν ο A δεν είναι ίσος με το #0B0H κάνει
                    ;άλμα στο "MNC"
174.      MOV    A,#42H      ;A<- #42H
175.      LJMP   NEXT      ;κάνε άλμα στο "NEXT"
176.MNC:  CJNE  A,#0C0H,MND ;αν ο A δεν είναι ίσος με το #0C0H κάνει
                    ;άλμα στο "MND"
177.      MOV    A,#43H      ;A<- #43H
178.      LJMP   NEXT      ;κάνε άλμα στο 2"NEXT"
179.MND:  CJNE  A,#0D0H,MNE ;αν ο A δεν είναι ίσος με το #0D0H κάνει
                    ;άλμα στο "MNE"
180.      MOV    A,#44H      ;A<- #44H
181.      LJMP   NEXT      ;κάνε άλμα στο "NEXT"
182.MNE:  CJNE  A,#0E0H,MNF ;αν ο A δεν είναι ίσος με το #0E0H κάνει
                    ;άλμα στο "MNF"
183.      MOV    A,#45H      ;A<- #45H
184.      LJMP   NEXT      ;κάνε άλμα στο "NEXT"
185.MNF:  MOV    A,#46H      ;A<- #46H
186.
187.NEXT: MOV    @R0,A      ;μετέφερε τα δεδομένα του A στην μνήμη
                    ;που δείχνει ο R0
188.      INC    R0          ;R0<- R0 +1
189.      MOV    A,@R1      ;μετέφερε τα δεδομένα της μνήμης που
                    ;δείχνει ο R1, στον A
190.      ANL    A,#0FH      ;A<- A AND 00001111
191.NN0:  CJNE  A,#0H,NN1  ;αν ο A δεν είναι ίσος με το #00H κάνει
                    ;άλμα στο "NN1"
192.      MOV    A,#30H      ;A<- #30H
193.      LJMP   FIN        ;κάνε άλμα στο "FIN"
194.NN1:  CJNE  A,#1H,NN2  ;αν ο A δεν είναι ίσος με το #1H κάνει
                    ;άλμα στο "NN2"
195.      MOV    A,#31H      ;A<- #31H
196.      LJMP   FIN        ;κάνε άλμα στο "FIN"
197.NN2:  CJNE  A,#2H,NN3  ;αν ο A δεν είναι ίσος με το #2H κάνει
                    ;άλμα στο "NN3"
198.      MOV    A,#32H      ;A<- #32H
199.      LJMP   FIN        ;κάνε άλμα στο "FIN"
200.NN3:  CJNE  A,#3H,NN4  ;αν ο A δεν είναι ίσος με το #3H κάνει
                    ;άλμα στο "NN4"
201.      MOV    A,#33H      ;A<- #33H
202.      LJMP   FIN        ;κάνε άλμα στο "FIN"
203.NN4:  CJNE  A,#4H,NN5  ;αν ο A δεν είναι ίσος με το #4H κάνει
                    ;άλμα στο "NN5"
204.      MOV    A,#34H      ;A<- #34H
205.      LJMP   FIN        ;κάνε άλμα στο "FIN"
206.NN5:  CJNE  A,#5H,NN6  ;αν ο A δεν είναι ίσος με το #5H κάνει
                    ;άλμα στο "NN6"
207.      MOV    A,#35H      ;A<- #35H

```

```

208.      LJMP  FIN           ;κάνε άλμα στο "FIN"
209.NN6:  CJNE  A,#6H,NN7   ;αν ο A δεν είναι ίσος με το #6H κάνει
                          ;άλμα στο "NN7"
210.      MOV   A,#36H      ;A<- #36H
211.      LJMP  FIN           ;κάνε άλμα στο "FIN"
212.NN7:  CJNE  A,#7H,NN8   ;αν ο A δεν είναι ίσος με το #7H κάνει
                          ;άλμα στο "NN8"
213.      MOV   A,#37H      ;A<- #37H
214.      LJMP  FIN           ;κάνε άλμα στο "FIN"
215.NN8:  CJNE  A,#8H,NN9   ;αν ο A δεν είναι ίσος με το #8H κάνει
                          ;άλμα στο "NN9"
216.      MOV   A,#38H      ;A<- #38H
217.      LJMP  FIN           ;κάνε άλμα στο "FIN"
218.NN9:  CJNE  A,#9H,NNA   ;αν ο A δεν είναι ίσος με το #9H κάνει
                          ;άλμα στο "NNA"
219.      MOV   A,#39H      ;A<- #39H
220.      LJMP  FIN           ;κάνε άλμα στο "FIN"
221.NNA:  CJNE  A,#0AH,NNB  ;αν ο A δεν είναι ίσος με το #0AH κάνει
                          ;άλμα στο "NNB"
222.      MOV   A,#41H      ;A<- #41H
223.      LJMP  FIN           ;κάνε άλμα στο "FIN"
224.NNB:  CJNE  A,#0BH,NNC  ;αν ο A δεν είναι ίσος με το #0BH κάνει
                          ;άλμα στο "NNC"
225.      MOV   A,#42H      ;A<- #42H
226.      LJMP  FIN           ;κάνε άλμα στο "FIN"
227.NNC:  CJNE  A,#0CH,NND  ;αν ο A δεν είναι ίσος με το #0CH κάνει
                          ;άλμα στο "NND"
228.      MOV   A,#43H      ;A<- #43H
229.      LJMP  FIN           ;κάνε άλμα στο "FIN"
230.NND:  CJNE  A,#0DH,NNE  ;αν ο A δεν είναι ίσος με το #0DH κάνει
                          ;άλμα στο "NNE"
231.      MOV   A,#44H      ;A<- #44H
232.      LJMP  FIN           ;κάνε άλμα στο "FIN"
233.NNE:  CJNE  A,#0EH,NNF  ;αν ο A δεν είναι ίσος με το #0EH κάνει
                          ;άλμα στο "NNF"
234.      MOV   A,#45H      ;A<- #45H
235.      LJMP  FIN           ;κάνε άλμα στο "FIN"
236.NNF:  MOV   A,#46H      ;A<- #46H
237.
238.
238.FIN:
239.      MOV   @R0,A        ;μετέφερε τα δεδομένα του A στην μνήμη
                          ;που δείχνει ο R0
240.      DJNZ  R5,JM        ;μείωσε τον R5 κατά ένα και αν δεν
                          ;είναι '0' κάνει άλμα στο "JM"
241.      RET                ;RETURN
242.JM:   LJMP  STRT        ;κάνε άλμα στο "STRT"
243.      RET                ;RETURN
244.
245.      END                ;ΤΕΛΟΣ

```

### Χρήση των θέσεων μνήμης του μικροελεγκτή:

**39 H:** στη μνήμη αυτή αποθηκεύεται το πρώτο byte που παίρνει ο μικροελεγκτής από τη σειριακή και καθορίζει αν θα κινηθεί η κάμερα ή θα μηδενίσει τη θέση της.

**40 H:** εδώ αποθηκεύεται το δεύτερο byte από τη σειριακή, το οποίο έχει το ρόλο του CONTROLBYTE. Καθορίζει ποιος κινητήρας θα κινηθεί και τι κίνηση θα εκτελέσει.

**41 H και 42 H:** στις δύο αυτές μνήμες αποθηκεύονται οι δύο τελευταίοι αριθμοί που παίρνει ο μικροελεγκτής από τη σειριακή και είναι τα βήματα που πρέπει να κάνει ο κινητήρας 1 και ο κινητήρας 2, αντίστοιχα.

**43 H και 44 H:** οι μνήμες αυτές έχουν το ρόλο των μετρητών. Μετράνε τις κινήσεις των κινητήρων από την αρχική τους θέση, ώστε να γνωρίζουμε πάντα τη θέση που βρίσκεται η κάμερα.

**53 H-56 H:** εδώ αποθηκεύονται τα δεδομένα που στέλνουμε στην RS-232 του H/Y. Η κάθε μνήμη περιέχει έναν από τους χαρακτήρες ASCII από 1-9 και από A-F. Οι δύο πρώτες θέσεις μνήμης μας δίνουν τη θέση της κάμερας στον κατακόρυφο άξονα και οι άλλες δύο τη θέση στον εγκάρσιο.

### **Οι ρουτίνες:**

#### **GtIns**

(γραμμές 44-48)

Ο μικροελεγκτής περιμένει να λάβει ένα byte το οποίο αποθηκεύεται στη μνήμη 39H. Για το byte αυτό, υπάρχουν δύο περιπτώσεις: ή να είναι ο αριθμός FF hex, δηλαδή όλα τα bit να είναι "1", ή να είναι οποιοσδήποτε άλλος αριθμός.

Η μνήμη 39 H ελέγχεται στη γραμμή 23 και αν είναι ίση με τον αριθμό FF hex συνεχίζεται κανονικά η ροή του προγράμματος, καλώντας τη ρουτίνα "SNDPOS", διαφορετικά γίνεται άλμα στο label "SETOP" όπου μηδενίζονται οι μνήμες 43 H και 44 H και γίνεται επιστροφή στο label "WtIns".

#### **SNDPOS**

(γραμμές 105-119)

Αρχικά καλεί τη ρουτίνα "INVRT", της οποίας η λειτουργία σχολιάζεται παρακάτω. Με την SNDPOS στέλνουμε στη σειριακή τις θέσεις μνήμης 53 H - 56 H, οι οποίες περιέχουν δεδομένα για τον υπολογισμό των συντεταγμένων της κάμερας.

#### **GtMvs**

(γραμμές 50-62)

Ο μικροελεγκτής σε αυτή τη ρουτίνα περιμένει να λάβει 3 byte τα οποία καθορίζουν το είδος της κίνησης των κινητήρων, με το πρώτο byte που αποθηκεύεται στην θέση 40H, και με τα δύο επόμενα, που αποθηκεύονται στις θέσεις 41 H και 42 H, και περιέχουν τα βήματα κίνησης κάθε κινητήρα.

#### **CntMtr1**

(γραμμές 65-81)

Η ρουτίνα αυτή καλείται μόνο όταν το bit τρία του byte που έχει αποθηκευτεί στη θέση 40 H είναι '1' (γραμμές 26-27). Η CntMtr1 ουσιαστικά περιέχει δύο υπορουτίνες: την "ST11" και την "ST12".

Η πρώτη τρέχει όταν το bit δύο του byte που έχει αποθηκευτεί στη θέση 40 H είναι '1' (γραμμές 65-66), και η δεύτερη όταν είναι '0'.

ST11: Αρχικά καλεί την ρουτίνα DELAY η οποία απλώς είναι μια χρονοκαυστήρηση. Μεταφέρεται ο αριθμός της πόρτας 1 στον Acc και τα ολισθαίνουμε αριστερά και μεταφέρουμε πάλι πίσω τα ολισθημένα δεδομένα στην πόρτα 1, το οποίο μεταφράζεται στην κάμερα σε ένα βήμα κάτω. Μετά μειώνουμε τα δεδομένα της μνήμης 44 H κατά "1". Αυτό θα τρέξει τόσες φορές, όσες ορίζει η μνήμη 41 H.

ST12: Αρχικά καλεί την ρουτίνα DELAY η οποία απλώς είναι μια χρονοκαθυστέρηση. Μεταφέρεται ο αριθμός της πόρτας 1 στον Acc και τα ολισθαίνουμε αριστερά και μεταφέρουμε πάλι πίσω τα ολισθημένα δεδομένα στην πόρτα 1, το οποίο μεταφράζεται στην κάμερα σε ένα βήμα πάνω. Μετά αυξάνονται τα δεδομένα της μνήμης 44 H κατά "1". Αυτό θα τρέξει τόσες φορές, όσες ορίζει η μνήμη 41 H.

### **CntMtr2**

(γραμμές 86-103)

Η ρουτίνα αυτή λειτουργεί ανάλογα με την CntMtr1, με τη διαφορά ότι τρέχει όταν είναι '1' το bit εφτά του byte που είναι αποθηκευμένο στην μνήμη 40 H. Επίσης οι αντίστοιχες υπορουτίνες ονομάζονται "ST21" και "ST22", η πρώτη κινεί την κάμερα αριστερά και τρέχει όταν το bit έξι του byte που έχει αποθηκευτεί στη θέση 40 H είναι '1' (γραμμές 86-87), και η δεύτερη για κίνηση της κάμερας δεξιά, όταν είναι '0'.

### **DELAY**

(γραμμές 122-127)

Αυτή η ρουτίνα απλώς δημιουργεί μια χρονοκαθυστέρηση ώστε να κρατάει τον παλμό πριν κάθε ολίσηση για κάποιο χρόνο. Αλλάζοντας τους αριθμούς στους καταχωρητές ή ακόμα αν προσθέσουμε και άλλον, μπορούμε να αλλάξουμε την ταχύτητα κίνησης της κάμερας.

### **INVRT**

(γραμμές 130-243)

Η ρουτίνα αυτή είναι η μεγαλύτερη του κώδικα και μάλλον η πιο σημαντική στην επικοινωνία του μικροελεγκτή με τον υπολογιστή. Κάθε byte που λαμβάνει ο υπολογιστής από την RS-232 το μεταφράζει σε χαρακτήρα ASCII, για το λόγο αυτό δεν μπορούσαμε να στείλουμε τους μετρητές και να ξέρουμε τι λαμβάνουμε. Έτσι έπρεπε να μετατρέψουμε τους μετρητές σε χαρακτήρες ASCII που να αντιστοιχούν σε δεκαεξαδικό αριθμό. Οι αντιστοιχία των χαρακτήρων φαίνεται στον παρακάτω πίνακα.

Για να γίνει καλύτερα κατανοητός ο τρόπος, με τον οποίο γίνεται η μετατροπή, παραθέτουμε ένα παράδειγμα:

Έστω ότι η κάμερα έχει κινηθεί από τη θέση [0,0] στη θέση [-9,0]. Επειδή οι συντεταγμένες είναι σε μοίρες και με κάθε βήμα ο κινητήρας στρέφεται 1.8 μοίρες, σημαίνει ότι η υπορουτίνα ST21, της CntMtr2 θα τρέξει πέντε φορές. Όταν επιστρέψει από την παραπάνω ρουτίνα, η θέση μνήμης 43H, από τον αριθμό 00H που είχε αρχικά, μετά από 5 μειώσεις (DEC 43H) θα έχει τον αριθμό 0FBH. Αυτό συμβαίνει επειδή αν μειώσουμε τα περιεχόμενα της μνήμης από 00H κατά μια μονάδα θα πάει στο 0FFH. Αν απλώς στείλουμε την μνήμη στον υπολογιστή, ο υπολογιστής θα καταλάβει ότι του στέλνουμε κάποιο χαρακτήρα ASCII που αντιστοιχεί σε αυτό το byte, το οποίο δεν μας είναι χρήσιμο. Σε αυτό το σημείο η ρουτίνα INVRT μας βγάζει από αυτό το αδιέξοδο, η οποία τρέχει κάθε φορά πριν την αποστολή των συντεταγμένων στον υπολογιστή.

<b>Χαρακτήρας ASCII</b>	<b>BYTE (HEX)</b>
'0'	30H
'1'	31H
'2'	32H
'3'	33H
'4'	34H
'5'	35H
'6'	36H
'7'	37H
'8'	38H
'9'	39H
'A'	41H
'B'	42H
'C'	43H
'D'	44H
'E'	45H
'F'	46H

Στην γραμμή 130 ο καταχωρητής R5 παίρνει την τιμή 2, όσες φορές χρειάζεται να τρέξει η ρουτίνα για να μετατρέψει και τους δύο μετρητές. Στις γραμμές 131 και 132 οι καταχωρητές R0 και R1 παίρνουν τις τιμές 52H και 42H αντίστοιχα, οι οποίοι θα λειτουργήσουν ως δείκτες σε θέσεις μνήμης. Αυξάνουμε τους δύο δείκτες κατά ένα έτσι ώστε ο R1 να δείχνει την μνήμη 43H που θέλουμε να μετατρέψουμε και ο R0 να δείχνει την μνήμη 53H, η οποία είναι η μνήμη που περιέχει το πρώτο δεκαεξαδικό ψηφίο που θέλουμε να στείλουμε, το F στο παράδειγμά μας.

Αντιγράφουμε τα δεδομένα της μνήμης που δείχνει ο R1, 43H, στον Acc και κάνουμε τη λογική πράξη AND με τον αριθμό 0F0H για να μηδενίσουμε τα 4 τελευταία σημαντικά bit του Acc και τα άλλα να μείνουν ως έχουν.

Ελέγχουμε αν ο Acc έχει τον αριθμό 00H, στο παράδειγμά μας έχει τον F0H, άρα θα κάνει άλμα στο MN1. Εκεί θα γίνει ο ίδιος έλεγχος αλλά αυτή τη φορά με τον αριθμό 10H και θα γίνει πάλι άλμα στο MN2. Αυτό θα γίνεται έως ότου βρεθεί ισότητα ή μέχρι να φτάσει στη γραμμή 182, όπως συμβαίνει στην περίπτωση μας, που κάνει το τελευταίο άλμα στο MNF, όπου πλέον δε χρειάζεται κανένας έλεγχος αφού η μόνη περίπτωση είναι να έχει τον αριθμό 0F0H, μεταφέρει τον αριθμό 46H στον Acc, δηλαδή όπως φαίνεται από τον πίνακα τον χαρακτήρα "F". Στην γραμμή 187 όπου βρίσκεται το label NEXT αντιγράφονται τα δεδομένα του Acc στη θέση μνήμης που δείχνει ο R0, την 53H.

Αυξάνουμε τον R0 κατά ένα ώστε να δείχνει πλέον την μνήμη 54H και αντιγράφουμε πάλι τα δεδομένα της 43H στον Acc αλλά αυτή τη φορά η πράξη AND γίνεται με τον 0FH, ώστε να μηδενίσουμε τα bit μετατρέψαμε πριν και να κρατήσουμε τα τέσσερα λιγότερο σημαντικά ανέπαφα. Γίνεται ο ανάλογος έλεγχος όπως και παραπάνω και μόλις φτάσει στο label MNB, όπου πλέον δε θα κάνει άλμα, μεταφέρει τον αριθμό 42H στον Acc, τον χαρακτήρα "B" και κάνει άλμα στο FIN για να μη συνεχίσει ανούσιους ελέγχους. Αντιγράφονται τα δεδομένα του Acc στη μνήμη 53H και αφού μειωθεί ο R5 κατά ένα και

πάει στο 1, κάνει άλμα στο JM, όπου εκεί κάνει άλμα στο STRT, επειδή η εντολή DJNZ δεν μπορεί να κάνει μεγάλα άλματα όπως η LJMP.

Η ρουτίνα θα τρέξει άλλη μία φορά και στο τέλος θα έχουμε στις μνήμες τους παρακάτω χαρακτήρες:

53H<- 'F'

54H<- 'B'

55H<- '0'

56H<- '0'

Η χρήση των χαρακτήρων που λαμβάνει ο υπολογιστής, φαίνεται στην ανάλυση των συναρτήσεων του MATLAB που ακολουθεί.

## **ΠΑΡΑΡΤΗΜΑ Β΄**



## Κώδικας στο MATLAB

### movecam

```
1. function movecam(gwnia1,gwnia2);
2. %movecam:METAKINEI THN CAMERA
3. %movecam(gwnia1,gwnia2)
4. %gwnia1:GWNIA KINHSHS ARISTERA-DEXIA SE MOIRES
5. %   ARISTERA:gwnia1<0
6. %   DEXIA:gwnia1>0
7. %gwnia2:GWNIA KINHSHS PANW-KATW SE MOIRES
8. %   KATW:gwnia1<0
9. %   PANW:gwnia1>0
10. %
11. %p.x.
12. %ETSW OTI H CAMERA VRISKETAI STH 8ESH [0,0]
13. %POS=movecam(-20,9);
14. %H CAMERA 8A KINH8EI APO THN 8ESH [0,0] 19.8** MOIRES ARISTERA
15. %KAI 9 MOIRES PANW
16. %
17. %COMMAND WINDOW:
18. %H CAMERA VRISKETAI STH 8ESH [ -19.8 , 9.0 ]
19. %
20. %**PROSOXH H CAMERA MPOREI NA KINH8EI MONO SE GWNIES
21. %POLLAPLASIA TOY 1.8
22. % EPEIDH TO VHMA TWN VHMATIKWN KINHTHRWN EINAI 1.8
23. % OI SYNTETAGMENES EINAI O ARI8MOS TWN VHMATWN POU APEXEI H CAMERA APO
24. % TH 8ESH [0,0]
25.
26.
27. times1=floor(sqrt((gwnia1/1.8)^2));
28. times2=floor(sqrt((gwnia2/1.8)^2));
29.
30. if times1==0&times2==0
31.     error('H CAMERA DEN MPOREI NA KANEI TOSO MIKRES KINHSEIS- AYXHSTE TIS GWNIES')
32.
33. end
34.
35. s = serial('COM1');
36.
37. set(s,'BaudRate',9600);
38.
39. fopen(s);
40. fwrite(s,bin2dec('11111111'),'char');
41. out=fscanf(s,'%c',[1,4]);
42. fclose(s);
43.
44. pos=[hex2dec(out(1:2)),hex2dec(out(3:4))];
45. if pos(1)>100||pos(2)>100
46.     if pos(1)>100
47.         pos(1)=pos(1)-256;
48.     end
49.     if pos(2)>100
50.         pos(2)=pos(2)-256;
51.     end
52. end
53.
54. UP_AV=20-pos(2);
55. DOWN_AV=20+pos(2);
56. RIGHT_AV=50-pos(1);
57. LEFT_AV=50+pos(1);
```

```

58.
59. if gwnia1>=0&gwnia2>=0
60.   controlbyte='10001000';
61.   if times1>RIGHT_AV||times2>UP_AV
62.     if times1>RIGHT_AV&times2<=UP_AV
63.       nomove
64.       error('MEIWSTE THN GWNIA DEKSIA')
65.     elseif times1<=RIGHT_AV&times2>UP_AV
66.       nomove
67.       error('MEIWSTE THN GWNIA PANW')
68.     else
69.       nomove
70.       error('MEIWSTE KAI TIS DYO GWNIES')
71.     end
72.   end
73. elseif gwnia1<0&gwnia2>=0
74.   controlbyte='11001000';
75.   if times1>LEFT_AV||times2>UP_AV
76.     if times1>LEFT_AV&times2<=UP_AV
77.       nomove
78.       error('MEIWSTE THN GWNIA ARISTERA')
79.     elseif times1<=LEFT_AV&times2>UP_AV
80.       nomove
81.       error('MEIWSTE THN GWNIA PANW')
82.     else
83.       nomove
84.       error('MEIWSTE KAI TIS DYO GWNIES')
85.     end
86.   end
87. elseif gwnia1>=0&gwnia2<0
88.   controlbyte='10001100';
89.   if times1>RIGHT_AV||times2>DOWN_AV
90.     if times1>RIGHT_AV&times2<=DOWN_AV
91.       nomove
92.       error('MEIWSTE THN GWNIA DEKSIA')
93.     elseif times1<=RIGHT_AV&times2>DOWN_AV
94.       nomove
95.       error('MEIWSTE THN GWNIA KATW')
96.     else
97.       nomove
98.       error('MEIWSTE KAI TIS DYO GWNIES')
99.     end
100.  end
101. else
102.   controlbyte='11001100'
103.   if times1>LEFT_AV||times2>DOWN_AV
104.     if times1>LEFT_AV&times2<=DOWN_AV
105.       nomove
106.       error('MEIWSTE THN GWNIA ARISTERA')
107.     elseif times1<=LEFT_AV&times2>DOWN_AV
108.       nomove
109.       error('MEIWSTE THN GWNIA KATW')
110.     else
111.       nomove
112.       error('MEIWSTE KAI TIS DYO GWNIES')
113.     end
114.   end
115. end
116. if times1= =0&times2~=0
117.   controlbyte=['0000',controlbyte(5:8)];

```

```

118.elseif times1~=0&times2==0
119. controlbyte=[controlbyte(1:4),'0000'];
120.end
121.
122.
123.
124.s = serial('COM1');
125.set(s,'BaudRate',9600);
126.fopen(s);
127.fwrite(s,bin2dec(controlbyte),'char');
128.fwrite(s,times2,'int8');
129.fwrite(s,times1,'int8');
130.out=fscanf(s,'%c',[1,4]);
131.fclose(s);
132.
133.POS=[hex2dec(out(1:2)),hex2dec(out(3:4))];
134.
135.if POS(1)>100||POS(2)>100
136. if POS(1)>100
137. POS(1)=POS(1)-256;
138. end
139. if POS(2)>100
140. POS(2)=POS(2)-256;
141. end
142.end
143.
144. NEWPOS=POS*1.8;
145. fprintf('H CAMERA VRISKETAI STH 8ESH [%2.1f,%2.1f ]\n',NEWPOS(1),NEWPOS(2));

```

**Γραμμές 2-24:** Σχόλια για τον κώδικα σχετικά με τη λειτουργία της συνάρτησης και τα ορίσματα εισόδου, τα οποία απευθύνονται στο χρήστη πληκτρολογώντας στο command window την εντολή >>help movecam

**Γραμμές 27-28:** Υπολογίζει πόσα βήματα πρέπει να κάνει ο κάθε κινητήρας για να κινηθεί η κάμερα, ανάλογα με τις μοίρες που εισάγει ο χρήστης.

**Γραμμές 30-33:** Ελέγχεται, αν ο χρήστης έχει θέσει πολύ μικρές γωνίες στροφής στους δύο άξονες κίνησης. Στην περίπτωση αυτή τερματίζεται η συνάρτηση και ενημερώνει το χρήστη να αυξήσει τις γωνίες.

**Γραμμές 35-42:** Ανοίγει την θύρα RS-232 του H/Y, με BR στα 9600 bps και στέλνει το πρώτο byte στον μικροελεγκτή, για να του δηλώσει ότι αφ' ενός η κάμερα θα κινηθεί και αφ' ετέρου ότι περιμένει να λάβει τις θέσεις της κάμερας όπου τις αποθηκεύει στο 'out', ένα πίνακα χαρακτήρων 1 × 4.

**Γραμμές 44-57:** Οι πρώτες δύο στήλες του out σχηματίζουν τη θέση της κάμερας στον κατακόρυφο άξονα και οι άλλες δύο τη θέση στον εγκάρσιο άξονα με δεκαεξαδικά ψηφία. Τα μετατρέπει σε δεκαδικά και αποθηκεύονται οι θέσεις στο pos (γραμμή 44). Ελέγχει αν κάποια από τις συντεταγμένες είναι πάνω από 100, που σημαίνει (επειδή η κάμερα δεν μπορεί να κινηθεί σε τόσο μεγάλο εύρος), ότι έχει λάβει θέση είτε αριστερά είτε κάτω (βλ. κώδικα assembly, ΠΑΡΑΡΤΗΜΑ Α'). Στην περίπτωση αυτή αφαιρεί από τον αριθμό που έλαβε το 256, ώστε να αποκτήσει πλέον τη σωστή θέση, με το σωστό πρόσημο (γραμμές 45-52). Τέλος υπολογίζει πόσο μπορεί να κινηθεί η κάμερα σε κάθε κατεύθυνση. Έχουν τεθεί ως όρια 50 βήματα (90 μοίρες) αριστερά ή δεξιά και 20 βήματα (36 μοίρες) πάνω ή κάτω από τη θέση [0,0].

**Γραμμές 59-120:** Σε αυτές τις γραμμές δημιουργείται το CONTROLBYTE, το οποίο καθορίζει την κίνηση της κάμερας. Παρακάτω παρουσιάζεται η σημασία κάθε bit του CONTROLBYTE:

**CONTROLBYTE:**

7	6	5	4	3	2	1	0
---	---	---	---	---	---	---	---

**Bit:**

**CONTROLBYTE.7:** 1: η κάμερα θα στραφεί στον κατακόρυφο άξονα  
0: η κάμερα δε θα στραφεί στον κατακόρυφο άξονα

**CONTROLBYTE.6:** 1: η κάμερα θα στραφεί αριστερά  
0: η κάμερα θα στραφεί δεξιά

**CONTROLBYTE.5:** αδιάφορο

**CONTROLBYTE.4:** αδιάφορο

**CONTROLBYTE.3:** 1: η κάμερα θα στραφεί στον εγκάρσιο άξονα  
0: η κάμερα δε θα στραφεί στον εγκάρσιο άξονα

**CONTROLBYTE.2:** 1: η κάμερα θα στραφεί προς τα κάτω  
0: η κάμερα θα στραφεί προς τα πάνω

**CONTROLBYTE.1:** αδιάφορο

**CONTROLBYTE.0:** αδιάφορο

**Γραμμή 59-72:** Ελέγχεται αν οι δυο γωνιές είναι θετικές. Σε αυτή την περίπτωση το CONTROLBYTE θα είναι '10001000' και ελέγχεται αν μπορεί η κάμερα να κάνει τις κινήσεις που της έχουν τεθεί. Σε περίπτωση που δεν μπορεί, βγαίνει από το πρόγραμμα με το ανάλογο μήνυμα αφού πρώτα καλέσει το nomove. Με την ίδια λογική ελέγχει στη συνέχεια και τις άλλες εκδοχές.

Μετά το τέλος της γραμμής 115 έχουν ελεγχθεί όλες οι περιπτώσεις εκτός από την περίπτωση η κάμερα να μην χρειάζεται να κινηθεί σε ένα από τους άξονες. Αυτό ελέγχεται στις γραμμές 116-120 και διαμορφώνεται ανάλογα το απαιτούμενο CONTROLBYTE.

**Γραμμές 124-145:** Ανοίγει την RS-232 και στέλνει το CONTROLBYTE όπως και το πλήθος των βημάτων που θα κινηθεί κάθε κινητήρας. Λαμβάνονται οι νέες θέσεις και αν χρειάζεται υλοποιείται η μετατροπή που πραγματοποιήθηκε με τις αρχικές θέσεις και μετά "τυπώνει" (στην οθόνη) ένα μήνυμα με τις νέες θέσεις.

## goto

```
1. function goto(posx,posy);
2. %GOTO:KINEI THN KAMERA STIS SYNTETAGMENES POY TOY ORIZOYME
3. %goto(posx,posy)
4. %posx:TETMHMENEH SE MOIRES
5. %posy:TETAGMENEH SE MOIRES
6. %
7. %P.X. ESTW OTI H CAMERA VRISKETAI STH 8ESH [0,0]
8. %>>goto(18,-36)
9. %LYNEI TO ANTISTROFO KINHMATIKO PROVLHMA WSTE
10. %NA YPOLOGISEI POSO PREPEI NA KINH8EI KA8E KINHTHRAS
11. %GIA NA FTASEI APO TIS SYNTETAGMENES POY VRISKETAI STIS SYNTETAGMENES
12. %POY TOY ORISAME
13. %COMMAND WINDOW:
14. %H CAMERA KINH8HKE 18 MOIRES DEXIA KAI 36 MOIRES KATW
15. %KAI VRISKETAI PLEON STH 8ESH [18.0,-36.0 ]
16. if posx<-90|posx>90|posy<-36|posy>36
17. error('H CAMERA MPOREI NA KINH8EI STON AXONA X APO -90 EWS 90 MOIRES \n KAI STON
18. AXONA Y APO -36 EWS 36 MOIRES')
19. end
20. s = serial('COM1');
21. set(s,'BaudRate',9600);
22. fopen(s);
23. fwrite(s,bin2dec('11111111'),'char');
24. out=fscanf(s,'%c',[1,4]);
25. fclose(s);
26. pos=[hex2dec(out(1:2)),hex2dec(out(3:4))];
27. if pos(1)>100||pos(2)>100
28.     if pos(1)>100
29.         pos(1)=pos(1)-256;
30.     end
31.     if pos(2)>100
32.         pos(2)=pos(2)-256;
33.     end
34. end
35.
36. gwnia1=posx-(pos(1)*1.8);
37. gwnia2=posy-(pos(2)*1.8);
38. times1=floor(sqrt((gwnia1/1.8)^2));
39. times2=floor(sqrt((gwnia2/1.8)^2));
40. if gwnia1>=0&gwnia2>=0
41.     controlbyte='10001000';
42.     s1='DEXIA';
43.     s2='PANW';
44. elseif gwnia1<0&gwnia2>=0
45.     controlbyte='11001000';
46.     s1='ARISTERA';
47.     s2='PANW';
48. elseif gwnia1>=0&gwnia2<0
49.     controlbyte='10001100';
50.     s1='DEXIA';
51.     s2='KATW';
52. else
53.     controlbyte='11001100';
54.     s1='ARISTERA';
55.     s2='KATW';
56. end
57. if times1==0&times2~=0
58.     controlbyte=['0000',controlbyte(5:8)];
```

```

59. elseif times1~=0&times2==0
60. controlbyte=[controlbyte(1:4),'0000'];
61. end
62. if times1==0&times2==0
63. nomove
64. fprintf('H CAMERA VRISKETAI HDH SE AYTH TH 8ESH \n')
65. return
66. end
67. s = serial('COM1');
68. set(s,'BaudRate',9600);
69. fopen(s);
70. fwrite(s,bin2dec(controlbyte),'char');
71. fwrite(s,times2,'int8');
72. fwrite(s,times1,'int8');
73. out=fscanf(s,'%c',[1,4]);
74. fclose(s);
75.
76. POS=[hex2dec(out(1:2)),hex2dec(out(3:4))];
77. if POS(1)>100||POS(2)>100
78. if POS(1)>100
79. POS(1)=POS(1)-256;
80. end
81. if POS(2)>100
82. POS(2)=POS(2)-256;
83. end
84. end
85. NEWPOS=POS*1.8;
86. fprintf('H CAMERA KINH8HKE %2.1f MOIRES %s KAI %2.1f MOIRES %s
87. \n',times1*1.8,s1,times2*1.8,s2);
88. fprintf('KAI VRISKETAI PLEON STH 8ESH [%2.1f,%2.1f]\n',NEWPOS(1),NEWPOS(2));

```

Η συνάρτηση goto λειτουργεί ακριβώς με τον ίδιο τρόπο όπως η movecam. Η διαφορά τους είναι ότι η movecam επιλύει το ευθύ κινηματικό πρόβλημα ενώ η goto επιλύει το αντίστροφο κινηματικό πρόβλημα .

Η διαφορές τους στον κώδικα βρίσκονται σε δύο σημεία: Το πρώτο σημείο βρίσκεται στην πρώτη if, η οποία ελέγχει αν οι συντεταγμένες που έχουν οριστεί, βρίσκονται εντός του εύρους κίνησης της κάμερας. Το δεύτερο είναι στις γραμμές 36-37 όπου μετά τη λήψη των συντεταγμένων της κάμερας, υπολογίζει τη γωνία που θα πρέπει να κινηθεί η κάμερα για να φτάσει στις θεθείσες συντεταγμένες. Τέλος, στο σχηματισμό του CONTROLBYTE ορίζονται δύο λέξεις (strings) s1 και s2 στα οποία αποθηκεύονται, ανάλογα με την περίπτωση, η κατεύθυνση προς την οποία εστράφη η κάμερα σε κάθε άξονα για να γίνει δυνατή η εκτύπωση (στην οθόνη) του σωστού μηνύματος μετά την κίνηση.

## scnroom

```
1. clc; %καθαρίζει το command window
2. vfm('show', true); %ανοίγουμε το driver της κάμερας
3. fprintf('VALTE TIS SWSTES RY8MISEIS SYMFWNA ME TO MANUAL,KAI PATHSTE ENTER \n')
4. pause %παύση για να ρυθμίσουμε την κάμερα
5. fprintf('AN EISTE SIGOYROS PATHSTE ENTER GIA NA SYNEXISETE \n')
6. pause
7. vfm('show',false); %κλείνουμε το driver της κάμερας
8.
9. A=[-40 25; %ο πίνακας με τις συντεταγμένες των θέσεων που μας ενδιαφέρουν
10. 0 25; %κάθε γραμμή περιέχει τις συντεταγμένες κάθε σημείου
11. 40 25; %πρώτη στήλη περιέχει τις τετμημένες και
12. -40 0; %η δεύτερη τις τεταγμένες
13. 0 0; %τις θέσεις της αλλάζουμε σύμφωνα
14. 40 0; %με το χώρο που θέλουμε να εμποτεύσουμε
15. -40 -25;
16. 0 -25;
17. 40 -25];
18.
19.
20.vfm('preview',true); %ανοίγουμε το driver της κάμερας
21.k=0; %μετρητής οποίος μας χρησιμεύει για να πάρουμε τις συντεταγμένες μας
22.for i=1:3 %χρησιμοποιούμε δύο for για να πάρουμε και να αποθηκεύσουμε τις
23. for j=1:3 %τις φωτογραφίες από τα σημεία που μας ενδιαφέρουν
24. k=k+1; %αυξάνουμε τον μετρητή κατά ένα
25. goto(A(k,1),A(k,2)); %στέλνουμε την κάμερα στο σημείο που μας ενδιαφέρει από τον πίνακα A
26. pause(1); %παύση για ένα δευτερόλεπτο, χρόνος ασφαλείας για να κινηθεί η κάμερα
27. I = vfm('grab', 1); %παίρνουμε την φωτογραφία
28. %την αποθηκεύουμε στον φάκελο Environment ως αρχείο ppm
29. filename = sprintf('%s%i%i%s', 'Environment\foto', i, j, '.ppm');
30. fprintf('Image saved: %s\n', filename);
31. imwrite(I(:,:,), filename);
32.
33. end
34.end
35.goto(0,0); %στέλνουμε την κάμερα στην αρχική της θέση
36.subplot(3,3,1),imshow('Environment\foto11.ppm'),title(['',num2str(A(1,1)),',',num2str(A(1,2)),''])
37.subplot(3,3,2),imshow('Environment\foto12.ppm'),title(['',num2str(A(2,1)),',',num2str(A(2,2)),''])
38.subplot(3,3,3),imshow('Environment\foto13.ppm'),title(['',num2str(A(3,1)),',',num2str(A(3,2)),''])
39.subplot(3,3,4),imshow('Environment\foto21.ppm'),title(['',num2str(A(4,1)),',',num2str(A(4,2)),''])
40.subplot(3,3,5),imshow('Environment\foto22.ppm'),title(['',num2str(A(5,1)),',',num2str(A(5,2)),''])
41.subplot(3,3,6),imshow('Environment\foto23.ppm'),title(['',num2str(A(6,1)),',',num2str(A(6,2)),''])
42.subplot(3,3,7),imshow('Environment\foto31.ppm'),title(['',num2str(A(7,1)),',',num2str(A(7,2)),''])
43.subplot(3,3,8),imshow('Environment\foto32.ppm'),title(['',num2str(A(8,1)),',',num2str(A(8,2)),''])
44.subplot(3,3,9),imshow('Environment\foto33.ppm'),title(['',num2str(A(9,1)),',',num2str(A(9,2)),''])
45.vfm('show',true); %μετά την εμφάνιση των φωτογραφιών σε ένα figure ανοίγουμε την κάμερα
```

Το scnroom είναι ένα M-file το οποίο κινεί την κάμερα σε εννέα επιλεγμένα από το χρήστη σημεία, λαμβάνει σε κάθε σημείο μια φωτογραφία και τελικά εμφανίζει σε ένα γράφημα τις φωτογραφίες, με τίτλο κάθε φωτογραφίας τις συντεταγμένες της θέσης λήψης.

Έτσι δίνεται η δυνατότητα να εμφανίζεται η εικόνα από εννέα σημαντικά για τον χρήστη σημεία, τα οποία επιθυμεί να εμποτεύει σε ένα χώρο και γνωρίζοντας τις συντεταγμένες τους να μετακινεί την κάμερα με τη συνάρτηση goto σε σημείο της επιλογής του.

### **nomove**

```
s = serial('COM1');  
set(s,'BaudRate',9600);  
fopen(s);  
fwrite(s,bin2dec('00000000'),'char');  
fwrite(s,0,'int8');  
fwrite(s,0,'int8');  
fclose(s);
```

Το M-file αυτό μεταφέρει στον μικροελεγκτή τρία byte με μηδενικά ώστε να μην κινηθεί η κάμερα. Ο τρόπος χρήσης του φαίνεται στον κώδικα του movcam, όπου χρησιμοποιείται.



### **setopos**

```
s = serial('COM1');  
set(s,'BaudRate',9600);  
fopen(s);  
fwrite(s,bin2dec('00000000'),'char');  
fclose(s)
```

Όταν καλείται αυτό το M-file, ορίζεται στον μικροελεγκτή η αρχική θέση έτσι ώστε να μηδενιστούν οι μετρητές που χρησιμοποιούνται για τον υπολογισμό της θέσης της κάμερας. Όπως φαίνεται παραπάνω και στον κώδικα, στέλνεται “0” στον μικροελεγκτή.