Palestine Polytechnic University

College of Engineering & Technology
Electrical and Computer Department

Graduation Project

Moving Mouse Cursor According To Eye Tracking

Project Team
Ala'a AL- Badareen
Fady Waheed Qabajeh
Mohammad Younes Ateya

Project Supervisor
Eng. Liana Tamimi

Hebron – Palestine

Palestine Polytechnic University

College of Engineering & Technology
Electrical and Computer Department

Graduation Project

Moving Mouse Cursor According To Eye Tracking

Project Team
Ala'a AL- Badareen
Fady Waheed Qabajeh
Mohammad Younes Ateya

Project Supervisor
Eng. Liana Tamimi

Hebron – Palestine

Moving Mouse Cursor
According To Eye Tracking

Project Team
Ala'a AL- Badareen
Fady Waheed Qabajeh
Mohammad Younes Ateya

Project Supervisor
Eng. Liana Tamimi

This Under-Graduate Project Report submitted to Computer and Electrical Engineering "Department in College of Engineering and Technology

Palestine Polytechnic University

For accomplishment the requirements of the bachelor degree in Engineering field at Computer System Engineering

Palestine Polytechnic University
Hebron – Palestine

2007

جامعة بوليتكنك فلسطين
الخليل – فلسطين
كلية الهندسة والتكنولوجيا
دائرة الهندسة الكهربائية والحاسوب

اسم المشروع

Moving Mouse Cursor According To Eye Tracking

أسماء الطلبة

علاء الدين عبد الباري البدارين   فادي وحيد قباجه
محمد يونس عطيه

بناء على نظام كلية الهندسة والتكنولوجيا واشراف ومتابعه المشرف المباشر على المشروع وموافقة اعضاء اللجنة الممتحنة تم تقديم هذا المشروع الى دائرة الهندسة الكهربائية والحاسوب وذلك للوفاء بمتطلبات درجة البكالوريوس في الهندسة تخصص هندسة أنظمة الحاسوب.

توقيع المشرف

....

توقيع اللجنة الممتحنة

....

توقيع رئيس الدائرة

....

DEDICATION

To our parents who made it all possible to receive the stage .

To our brothers and sisters for their encouragement .

To all the people who encouraged us during the preparation of this project .

To all the people who like to know and look for the knowledge .

For your mentorship, friendship, and tireless devotion to insisting that we "get it right" and helping us to so .

Ala'a AL-Badareen
Fady Waheed Qabajeh
Mohammad Younes Ateya

ACKNOWLEDGMENT

We would like to thank Engineer Layana Tamimi who read our numerous revisions, helped us to make some sense of the confusion, and his valuable time.

We must not forget to thank all the instructors in the Electrical and Computer Engineering Department for their great impact in our education.

Special thanks to Eng. Mazen Zaloum for all his care and valuable advices since the first day we joined Palestine Polytechnic University.

Special thanks to Eng. Hashem Tamimi for his precious help in image processing, and Eng. Khaled Daghameen for his help in Java.

We acknowledge Palestine Polytechnic University for giving us the possibility to show some of what we have learned from it

Finally we can't forget to acknowledge our great parents who sacrificed themselves for educating us and facilitate our life, and for all their *coffee* and tolerance.

Ala'a AL-Badareen
Fady Waheed Qabajeh
Mohammad Younes Ateya

ABSTRACT

In this project the user can control the pc mouse cursor and its operation using his eyes by connect special camera with the user computer using special software then the mouse can moving according to the eye movements.

The project importance appear in the filed to help the hand cut person to using his pc in normal way without necessary to use his hands, and control his desk top like any other person , also this person can control some functions such as open the door , close the window, light the room through his pc .

The first part of the project is the hardware part and it presented by the DC Motor circuit it controlled by user eye through the parallel port, we use the motor to open window and close it as sample application.

The second part is the software part and its represent by agent paint program the user eye control this program and the user can drown and pint the figures through his eye movement, also the this part contain the desktop application through this part the user can use the desktop normally as click some item and drag it to move to some folder.

The project success is about 85% but because the camera are used are not very accurate and the domain for the object to move from the camera is very near this case the make some nose in specified location to the cursor sure also the cursor cant receive every pixel in the screen although we increase the scale of the output coordinates x & y that are taken by the camera .

خلاصة

من خلال هذا المشروع يستطيع المستخدم أن يستخدم بمؤشر الفارة وبوظائفه من خلال عين المستخدم ويتم ذلك استخدام كاميرا ذات حساسية شديدة للحركة أضافه إلى برنامج خاض, وتظهر أهمية المشروع بشكل كبير في مجال مساعده الاحتياجات ا الخاصة وخصوص مقطعو الأيدي من استخدام الكمبيوتر بشكل طبيعي كأي شخص طبيعي بالاضافه إلى ذلك يستطيع المستخدم من أن يتحكم بوظائف معينه كفتح شباك معين من خلال الكمبيوتر .

يتكون هذا المشروع من جزأين, الجزء الأول يتكون من الكيان المادي ويتمثل هذا الجزء في بناء دائرة التشغيل لل "DC Motor" حيث يتم التحكم بهذا الجهاز عن طريق منفذ الكمبيوتر "Parallel Port" من خلال إشارة تتحدد من خلال موقع المؤشر في شاشة المستخدم وهذا الموقع يتحدد تبعا لحركة عين المستخدم ويستخدم هذا الماتور كتطبيق بسيط يمكن للمستخدم القيام به مثل فتح أو إغلاق شباك معين في غرفة المستخدم للنظام .

أما الجزء الآخر من النظام فهو الكيان البرمجي من برنامج رسام حيث يتمكن المستخدم من أن يرسم الأشكال المختلفة ويلونها من خلال حركة عينه, كما يتكون هذا الجزء من تطيق سطح المكتب حيث يمكن للمستخدم للنظام من ممارسة الوظائف الأساسية الخاصة بسطح المكن مثل فتح أيقونه معينه أو نقل ملف من مكان إلى آخر على سطح المكتب الخاص به.

نسبة النجاح لهذا المشروع بلغت حوالي 85% ولكن بسبب عدم توفر الدقة العالية للكاميرا المستخدمة في المشروع هناك نسبه من الخطأ في تحديد الموقع الجديد للمؤشر بشكل صحيح , أضافه إلى ذلك فان مجال الحركة للجسم أمام هذه الكاميرا ضيق نسبيا لذلك فان المؤشر لا يتمكن من الوصول إلى كامل المساحة على سطح المكتب على الرغم من قيامنا بزيادة قيمة الإحداثيات الجديدة للمؤشر من خلال ضربها برقم معين لكي تصل إلى جميع أجزاء سطح المكتب إلا أن هذا أدى إلى تقليل نسبه الدقة في اختيار رمز معين على سطح المكتب .

Table of Contents

List of Figures

List of Tables

CHAPTER 1 : INTRODUCTION

Chapter One

Introduction

Introduction

1.1 General Description Of The Project

Use of the human eye as an input device for personal computer remains a largely untapped field. This design project shall attempt to explore this field further by developing a low-cost input device to be usable by an average PC user. The team shall attempt to combine special purpose EOG sensor or special purpose camera and optical image processing to emulate a basic PC mouse, an apparatus shall be designed and built to perform this tracking and emulate a standard PC mouse via serial, PS/2, or USB interface.

Also the problem of human-computer interaction can be viewed as two powerful information processors (human and computer) attempting to communicate with each other via a narrow-bandwidth, highly constrained interface, To address it we seek faster, more natural, and more convenient means for users and computers to exchange information.

The user's side is constrained by the nature of human communication organs and abilities; the computer's is constrained only by input/output devices and interaction techniques that we can invent. Current technology has been stronger in the computer-to-user direction than user-to-computer; hence today's user-computer dialogues are rather one-sided, with the bandwidth from the computer to the user far greater than that from user to computer.

The Eye Mouse offers people with extreme disabilities the opportunity to control a computer simply by moving his or her eyes or head. Computers are generally controlled by hand, using a keyboard, a mouse, or a trackball. Many people who are unable to use their hands are thus disenfranchised from using this equipment.

1.2 project objective

The main objectives of our project are:

- Moving the mouse curser by eye tracking.

- Help handicapped and to move the normal PC computer mouse curser easily

- Creating an over view to other people who don't have background about eye tracking and its advantages.

- To provide the user ability to connect the pc computer with other devices and how to build suitable interface using appropriate programming language.

- Using eye tracking to control some hardware application such as closing and opening window.

- Using eye tracking to control some software application such as agent painter program.

1.3 Literature review for eye tracking

The project name: An eye tracking computer user interface, done by Arie E. Kaufman, Amit Bandopadhay and Bernard D. Shaviv refried to Computer Science Department in State University of New York at Stony Brook

We developed an inexpensive hardware software system for eye tracking. It is based on electro - oculography (EOG) rather than expensive reflectance based methods. We built a prototype to demonstrate the viability of EOG for human - computer communication. The system is applicable for many virtual reality systems, video games, and for the handicapped, the hardware component of the project are: 4 EOG sensing channels, Signal filters to eliminate noise, Signal amplifiers, A/D converters

Fig 1.1 The location of sensor on the face

1.4 Project Schedule and Time Plan

In this section we will demonstrate the system schedule for the developers, take in care the time period given for delivery the system, and the dependences between tasks.

1.4.1 System Schedule

This system can be divided into the following tasks:

T1: Collecting Information and Determining the Main Objectives of the System

This includes collecting whole information and research in many resources a bout central database management between server and clients such as how are data exchanged between the server and clients, after this finish we will determine the objectives of the system.

T2: Determining the Main Requirements of the System:

Involves determining the functional and non-functional requirements of the system.

T3: Software and System Requirements Specification:

Involves finding out, analyzing and checking the main services and constraints on system.

T4: System Organization and Hardware and Software Requirements Specification:

Which includes how this system will organize between students, and also determining the most important HW and SW requirements of this system.

T5: Analysis of Expected Risks:

Involves the system risks, types of risks and what resolutions strategies that will be performed to reduce these risks.

T6: System Scheduling:

In this phase, will determine the work activities, system schedule and report monitoring.

T7: Creating a Suitable Model for All parts that Will Be Built in Our System:

This includes graphical representations for the centralized database system that is to be developed.

T8: Designing the hardware circuit:

This involves connect hardware device together.

T9: Designing the camera used:

This involves selected the camera type and it we want to wok .

T10: Built the needed classes:

This includes built the classes that the camera needed to work.

T11: Designing the User Interface:

This includes built the final interface that the user well deal with in this system .

T12: Implementing the hardware circuit:

This involves creating and conceiting the hardware device alarm bell and alarm light to the parallel port.

T13: Implementing the camera used in our system:

Which includes creating and connecting the camera circuit through the serial port .

T14: Implementing the classes needed for the project :

This includes implement input output classes that our system needed also implement the camera driver .

T15: Implementing the Interface program:

This involves creating the final interface that the user can deal with.

T16: Testing:

This includes testing overall system by using multiple testing technologies.

T17: Documentation:

This includes writing the full documentation for our system from the first moment the works began until the delivery time.

1.4.2 Task Duration and Schedule

All of the above tasks are tabulated in the following table, which specifying tasks with its duration and dependences if found.

Table 1.1 Task Schedule, Duration, and Dependence

Task	Duration(days)	Dependencies
T1	24	
T2	15	T1(M1)
T3	20	
T4	13	T3(M2)
T5	12	T4(M3)
T6	13	T4(M3)
T7	17	T3(M2)
T8	18	T7(M4)
T9	16	T8(M5)
T10	22	T8(M5)
T11	17	T3(M2)
T12	25	T9,T10(M6)
T13	20	T12(M7)
T14	60	T13(M8)
T15	30	T14(M9)
T16	18	T15(M10)
T17	266	

1.4.2.1 Allocation of Activities to Engineers

This section specifies allocation of tasks to engineers participated in the system.

Table 1.2 Allocations of Tasks to Engineers

Task	Engineer(s)
T1	Fady, Ala'a
T2	Muhammad
T3	Fady
T4	Fady
T5	Mohammad
T6	Ala'a
T7	Ala'a
T8	Fady
T9	Mohammad
T10	Ala'a
T11	Mohammad
T12	Fady
T13	Ala'a
T14	Fady
T15	Mohammad
T16	Fady
T17	Fady , Mohammad , Ala'a

1.5 Estimated Cost

In this section we will demonstrate the system cost estimation in term of system hardware, system software, and human resources.

1.5.1 Hardware Cost Estimation:

The following table contains the main requirements for our system and their costs:

Table 1.3 Hardware Cost Estimation

Component	Cost
One Personal Computer	600 $
Printer	80 $
CMU Camera	200 $
Operational amply fire	5 $
Alarm Bell	5 $
Light Led	2 $
Breadboard	5 $
Total Cost	897 $

1.5.2 Software Cost Estimation:

The following table contains the software that will be in our system and its cost:

Table 1.4 Software Cost Estimation

Component	Cost
Visual Studio.NET	799 $
Windows XP professional	299$
Microsoft Office	329 $
Java Virtual Machine	200 $
Microsoft Visio	165 $
JCreator Pro	200 $
Total Cost	1992 $

1.5.3 Human Resources Cost Estimation:

The following table contains the human resources that will be used in our system and its cost.

Table 1.5 Human Resources Cost Estimation

Number of Developers	Work Hours (monthly)	Cost per hour	Total Cost (monthly)
3	120	10 $	3600 $

1.5.4 Total Cost:

The total estimated costs that need to implement our system may be raised up to 6489 $.

1.6 Risk Management:

This section specifies the system risks that may face the developer, type of these risks, and resolution strategies for these risks.

1.6.1 System Risks:

In this project the team may face the following problems:

1. Some activities may not be made on time for some reasons, such as any of the team members may become ill.

2. The system may fail in the operation stage...

3. Changing the requirements, which require major redesign and need to be proposed.

4. The end-user may face some problems in using the system; even if he wants more or maybe he thinks that the system can do more than what is done.

5. The cost of the project may exceed the estimated cost, so the budget will not cover the cost.

6. Political situation may affect the time schedule.

7. The time required to develop the system is underestimate.

8. Failure in the system as a hole.

1.6.2 Types of Risks:

The possible risks are considered to be:

1. Technology: the database management system used in the system cannot process as many transaction per second as expected, the network may become over headed by a high throughput resulted from sending and receiving messages.

2. Human: any of the team members may stop his work for many reasons, such as illness.

3. Organizational: organization financial problem force reduction in the system budget.

4. Requirements: change the requirements, which require major redesign and re-implementing.

5. Estimation: the team requires developing the software is underestimating; the rate of defect repair is underestimating.

1.7 Report Contents:

This documentation consists of six chapters, each chapter contain many sections.

The first chapter is an introduction which is consists of six sections general idea about the system and its importance, literature review, estimated cost, time plane and system schedule, risk management, and reports contents.

The second chapter is a theoretical background which is consisting of three sections which have theoretical subject related to the main ideas of the system, information about special components, and system requirement.

The third chapter is architectural design which is consisting of four sections whish have system objectives, general block diagram, how system work, and system modelling.

The fourth chapter is detailed system design which is consisting of four sections, design option, detailed description of system component, user interface design, and general algorithms and flowcharts.

The fifth chapter is software needed for the project and algorithms and code listing for the project.

The sixth chapter is system implementation and testing which is consisting of two sections system implementation, and testing.

The seventh chapter is conclusion and future works which is consist of two sections conclusion and results, and suggestion for future developments.

Finally the required needed appendices will be appended.

Chapter Two

Theoretical Background

2.1 Theoretical Background Related to the Main Idea of Project.

2.2 Hardware Related to the Project.

2.3 Project Integrity.

2.4 Theoretical Background about Project Components.

Chapter Two

Theoretical Background

This chapter provides an illustrative theoretical background for our project applications in general and for its each component in particularly.

2.1 Theoretical Background Related to the Project

Project is fully constructed over a communication between the computer and human using camera .

The problem of human-computer interaction can be viewed as two powerful information processors (human and computer) attempting to communicate with each other via a narrow-bandwidth, highly constrained interface.

To address it, we seek faster, more natural, and more convenient means for users and computers to exchange information. The user's side is constrained by the nature of human communication organs and abilities; the computer's is constrained only by input/output devices and interaction techniques that we can invent.

Current technology has been stronger in the computer-to-user direction than user-to-computer, hence today's user-computer dialogues are rather one-sided, with the bandwidth from the computer to the user far greater than that from user to computer.

Using eye movements as a user-to-computer communication medium can help redress this imbalance. This chapter describes the relevant characteristics of the Human eye, eye tracking technology, how to design interaction techniques that incorporate eye movements into the user-computer dialogue in a convenient and natural way, and the relationship between eye movement interfaces and virtual environments.

Eye tracking technology is newest and not use in large domain to help other student and people to understand eye tracking technology , so we choose this project and so we don't found many subject that are talk in this side but the little we find it help us to understand the idea of eye tracking and how that technology are used and worked and well we make our project .

From study the project that talk about eye tracking we set the plain of our project and how we well use the suitable hardware dive and how to connect theses devices into other and how they are worked and how we can connect them the personal computer , first we well talk about eye movement and the around environment.

2.2 Hypothesis and Hardware Related to the Project

2.2.1 Hypothesis

As with other areas of research and design in human-computer interaction, it is helpful to build on the equipment and skills humans have acquired through evolution and experience and search for ways to apply them to communicating with a computer.

Direct manipulation interfaces have enjoyed great success largely because they draw on analogies to existing human skills (pointing, grabbing, moving objects in space), rather than trained behaviors.

Similarly, we try to make use of natural eye movements in designing interaction techniques for the eye. Because eye movements are so different from conventional computer inputs, our overall approach in designing interaction techniques is, wherever possible, to obtain information from a user's natural eye movements while viewing the screen, rather than requiring the user to make specific trained eye movements to actuate the system.

This requires careful attention to issues of human design, as will any successful work in virtual environments. The goal is for human-computer interaction to start with studies of the characteristics of human communication channels and skills and then develop devices, interaction techniques, and interfaces that communicate effectively to and from those channels.

We thus begin with a study of the characteristics of natural eye movements and then attempt to recognize appropriate patterns in the raw data obtainable from the eye tracker, turn them into tokens with higher-level meaning, and design interaction techniques for them around the known characteristics of eye movements.

This approach to eye movement interfaces meshes particularly well with the field of virtual environments. The essence of virtual environment and other advanced interface approaches is to exploit the user's pre-existing abilities and expectations.

Navigating through a conventional computer system requires a set of learned, unnatural commands, such as keywords to be typed in, or function keys to be pressed.

Navigating through a virtual environment exploits the user's existing "navigational commands," such as positioning his or her head and eyes, turning his or her body, or walking toward something of interest.

By exploiting skills that the user already possesses, advanced interfaces hold out the promise of reducing the cognitive burden of interacting with a computer by making such interactions more like interacting with the rest of the world.

The result is to increase the user-to-computer bandwidth of the interface and to make it more natural. An approach to eye movement interaction that relies upon natural eye movements as a source of user input extends this philosophy. Here, too, the goal is to exploit more of the user's pre-existing abilities to perform interactions with the computer.

Moreover, eye movements and virtual environments both exemplify a new, non-command style of interaction. Some of the qualities that distinguish such interfaces from more conventional types of interaction are shared by other newly emerging styles of human-computer interaction that can collectively be characterized as "non-command-based." In a non-command-based dialogue, the user does not issue specific commands; instead, the computer passively observes the user and provides appropriate responses to, for example, movement of his or her eyes, head, or hands.

Non-command-based interfaces will also have a significant impact on the design of future user interface software, because of their emphasis on continuous, parallel input streams and real-time timing constraints, in contrast to conventional single-thread dialogues based on discrete tokens.

2.2.2 Hardware Component

The hardware equipments needed for the operation of this system are:

- A personal computer.
- CMU cam version one type six.
- Breadboard.
- Optocupler (4N25)
- Transistor (NPN "BD243C") and (PNP "BD244C")

- Resistors (100,220 Ω each).

- DC Motor.

- A 12V,5V power supply and connection wires.

The Operating systems that must be installed on the PC is Microsoft Windows XP(R) Professional.

We use CMUcam in this project because we need sensitive camera for the color that are efficient for eye tracking and more fast in capture and processing the captured image , and because the eye boll movement is very fast the CMUcam is very fast in taking and processing the frames

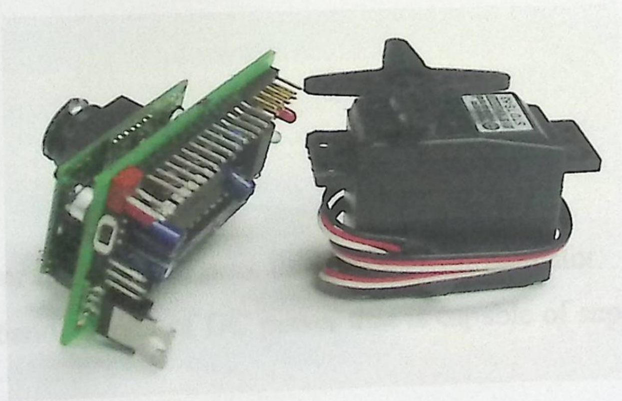


Fig 2.1 the CMUcam

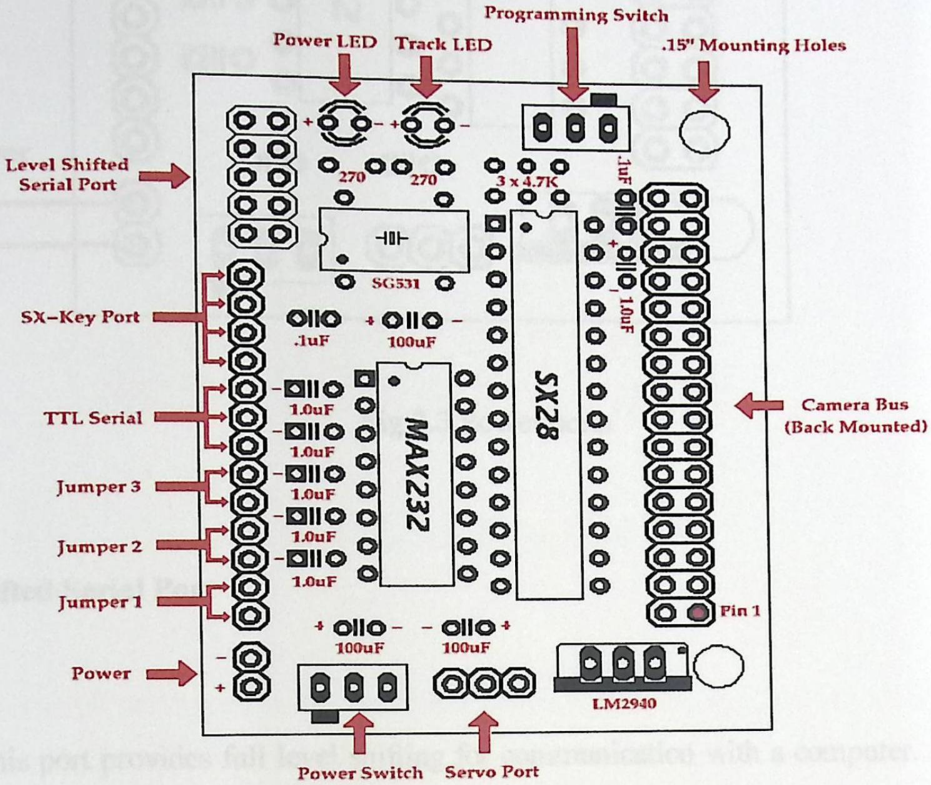


Fig 2.2 board layout

Ports:

Power

The input power to the board goes through a 5 volt regulator. It is ideal to supply the board with between 6and 7 volts of DC power that is capable of supplying at least 200 mill amperes of current.

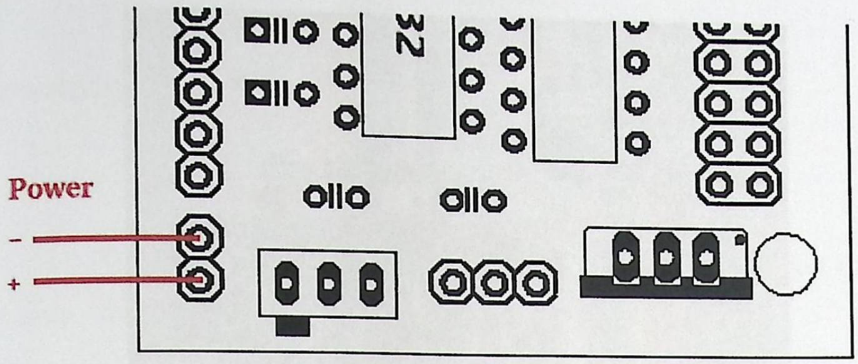


Fig 2.3 Power ports

Level Shifted Serial Port

This port provides full level shifting for communication with a computer. Though it only uses 3 of the 10 pins it is packaged in a 2x5 pin configuration to fit standard 9 pin ribbon cable clip-on serial sockets and 10 pin female clip on serial headers that can both attach to a 10 wire ribbon cable.

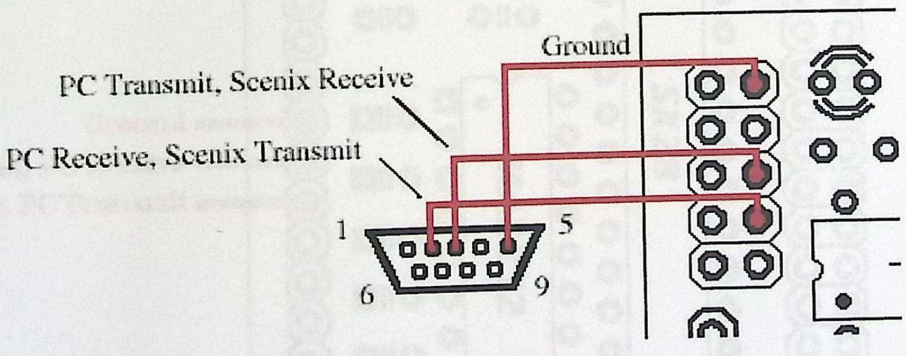


Fig 2.4 Level Shifted Serial Port

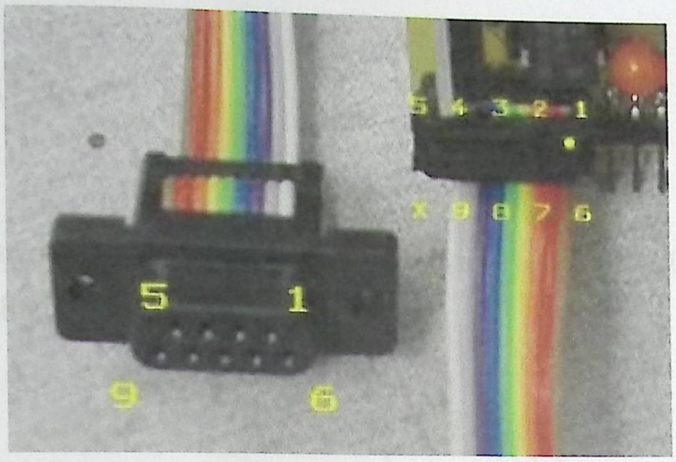


Fig 2.5 Level Shifted Serial Port

TTL Serial Port

This serial port taps into the serial I/O before it goes through the MAX232 chip. This port may be ideal for communication with a microcontroller that does not have any built in level shifting. Note that, it may be necessary to remove the max232 chip.

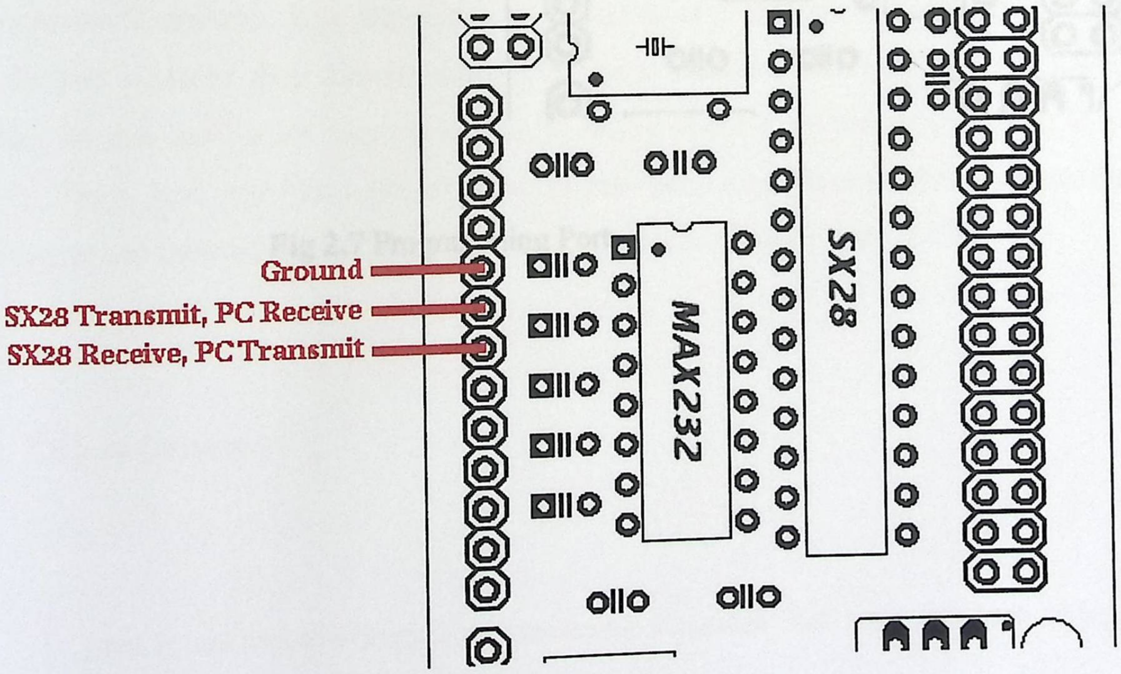


Fig 2.6 TTL Serial Port

Programming Port

The programming port allows the firmware to be downloaded to the SX28 using a SX-Key / Blitzer or equivalent programmer.

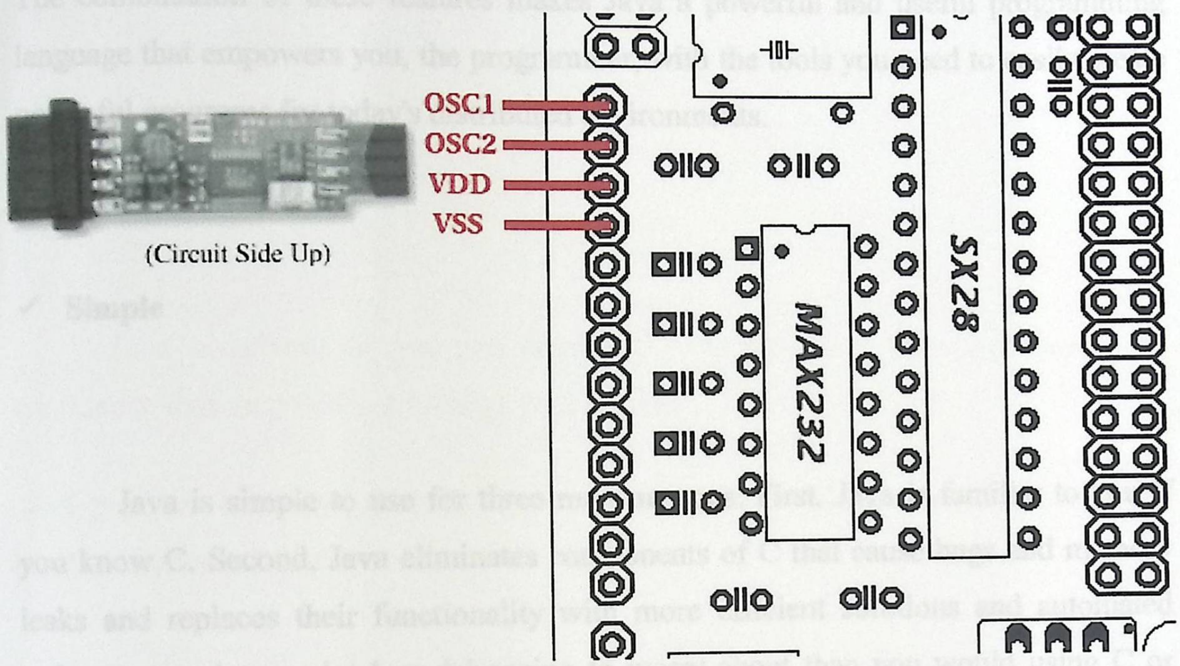


Fig 2.7 Programming Port

2.2.3 Software Feature

Sun describes Java as a "simple, object-oriented, interpreted, robust, secure, architecture-neutral, portable, high-performance, multithreaded, and dynamic language."

The combination of these features makes Java a powerful and useful programming language that empowers you, the programmer, with the tools you need to easily create powerful programs for today's distributed environments.

✓ **Simple**

Java is simple to use for three main reasons: First, Java is familiar to you if you know C. Second, Java eliminates components of C that cause bugs and memory leaks and replaces their functionality with more efficient solutions and automated tasks, so you have a lot less debugging to worry about than you would using C or C++. Third, Java provides a powerful set of pre-tested class libraries that give you the ability to use their advanced features with just a few additional lines of code.

✓ **Object-Oriented**

Java is an object-oriented programming language that uses software objects called classes and is based upon reusable, extensible code.

This means that you can use Java's classes, which are sets of variables and methods, as templates to create other classes with added functionality without rewriting the code from the parent classes or super classes.

If you plan your application's class hierarchy well, your application will be small and easy to develop. The hierarchy of classes is explained later in this chapter.

✓ **Interpreted**

you need only to compile for a single, virtual machine and your code can run on any hardware platform that has the Java interpreter ported to it.

✓ **Secure**

Java is secure, so you can download Java programs from anywhere with confidence that they will not damage your system.

Java provides extensive compile-time checking, followed by a second, multilayered level of runtime checking

✓ **High Performance**

Java is "high performance" because its byte code is efficient and has multithreading built in for applications that need to perform multiple concurrent activities.

Although threads still require the use of classes, Java balances the addition of thread synchronization between the language and class levels.

Java's byte code is efficient because it is compiled to an intermediate level that is near enough to native machine code that performance is not significantly sacrificed when the Java byte code is run by the interpreter.

2.3 Project Integrity

User Side :

The part in the project must be compatible with each other, started with the user eye it must be normal eye although the lashes over the eye make some nose in the selected frame, also some time the user eye ball have black color as the lashes cooler so the camera track both the lashes and the eye ball and this give wrong value for the new x & y coordinate value .

The eye ball movement is very small so it hard to track it so we cam put the camera in near location to the user eye and the user can move his eye ball in slow way in order to have correct result in the tracking process .

Camera Side :

In the camera side in order to be combatable to our system we must give it suitable DC voltage and current and deal with it in sensitive way and in the tracking process we can we can change it focus value to give frames with smallest value of noise and make its location fixes because any motion for it give wrong value in the both domain software and hardware .

PC Side :

The PC witch the camera work on must have a serial port in order to exchange data between the camera and software application among it, also the PC must be contain a parallel port to enable the user to control the hardware circuit through it, we need just two pen in the parallel port so there no need to need to active all the parallel port pens, the isolated circuit must be built in order to protect the PC parallel port .

In the software part the PC speed must be acceptable to order to combatable with high camera processor speed, also it must have some version of JVM to run the application on it .

In the desktop we arrange the important item near to the side of the screen that start with (0,0) pixel because the tracking result x & y don't contain the all the user screen, and we decrease the time to open item or to select item in desk top because the tracking curser are very twitter so we cant fixed the curser along time in specified location, in response the agent painter we design it to be an example for the future program the worked under tracking system .

DC Motor Side :

We don't choose an complex circuit for the motor also we don't choose accurate motor type because we need just to open window and close it, so there is no reason to put our self in problems we don't it such that the angle of the motor and arrangement of the signal from the parallel port in order to activate the motor .

2.4 Theoretical Background about Project Components

In this section, we provide a full explanation of each component and each part of this project.

2.4.1.1 CMUcam Vision System

Traditionally fast computers are needed to capture and process camera images. It is also necessary to write the software to perform this processing. Because of this it is difficult to use vision as a sensor in simple systems.

The CMUcam vision system uses a fast low cost microcontroller to handle all of the high speed processing of the camera data and contains software to perform simple vision tasks.

Because the user can choose to output only low bandwidth high level information from the vision system, like the red object is at position X-Y, it is possible for a simple processor like a PIC microcontroller to read this data and direct a small robot in tasks like chasing a colored ball. The CMUcam vision system makes it possible to ignore the complexity of camera interfacing and use vision just like any other sensor (i.e. sonar) often used in robotic systems.

2.4.1.2 CMUcam PCB Board Costs

The CMUcam PCB board costs about $60. The cost of the components to stuff the board at unit cost is about $34. The total cost of a single vision system is about $100.

2.4.1.3 CMUcam Frame Rate

The fastest frame rate is 16.7 frames per second. This means CMUcam can tell you the position of an object about 17 times per second. Using the serial software protocol you can slow this down if desired.

2.4.1.4 Baud Rates That the CMUcam Support

As of the latest version of the firmware (v1.12) the CMUcam can communicate at baud rates of 9600, 19200, 38400 or 115200. The baud rate is selected via jumper settings on the board.

2.4.1.5 Upgrading CMUcam Firmware

If you are currently using firmware v1.11 there is no reason to upgrade unless you need support for the lower baud rates. Version 1.11 supports 38400 and 115200 baud. Version 1.12 supports those baud rates and adds support for 9600 and 19200 baud. To upgrade your firmware you will need to reprogram the flash memory in your processor chip. The easiest way to upgrade is to purchase a new processor with the new code flashed into it.

To do this contact the company you purchased your CMUcam from. If you have a programmer (which is expensive) you can download the current version from our Downloads page and re-flash the processor yourself.

2.4.1.6 CMUcam Can Work Outside

CMUcam works outside but not in direct sunshine. The CMOS camera we are using does not have a high-quality IR filter, and so sunlight saturates the red pixels and the image becomes, essentially, monochrome in direct sunlight. (It is possible to add an external filter to improve outdoor operation.) We have a short write up in Publications that shows you how illumination conditions affect CMUcam.

2.4.1.7 Writing Custom Code For CMUcam

Certainly. To do this, ask us for the firmware source code, order yourself a C programming environment for the Ubicom (Scenix) chip and go for it. When we send you the firmware source code we will also send you a summary of how the code is structured and information about the programming environment we use.

2.4.1.8 The Power That CMUcam Consume

The complete system consumes about 200 mill amperes.

2.4.1.9 When The Camera's Field of View

This depends on the lens that you fit to the CMOS camera. If you order the CMOS camera as it is sold standard, you will end up with about a 25 degree field of view, which is relatively narrow. You can custom-order wider angle lenses when you order your CMOS camera, however.

2.4.1.10 Processor of CMUcam Use

Our image processing code resides on a microcontroller chip from Ubicom running at 75 MHz, the SX28AC. (Note that the company that manufactures this chip recently changed their name from Scenix to Ubicom.) This chip reads all the pixels from the CMOS camera via a parallel interface and does all the processing in real time. It then communicates the results to your microcontroller or computer via a serial port interface.

2.4.1.11 Developing More Powerful Versions of This Vision Sensor

Yes. We are currently working on a system which uses the Ubicom SX52 processor. This processor runs 33% faster than the processor we are currently using and has almost twice as much RAM and twice as much flash program ROM.

With this processor we should be able to achieve a higher frame rate and include more functionality in the system. We chose not to use this processor in the initial system because it is only available in a harder to work with surface mount package.

Chapter Three

Project Conceptual Design

3.1 Project Objectives.

3.2 Design Options.

3.3 Design Realization Approach (Implementation).

3.4 Project Design Block Diagrams.

3.5 Project Interacting With the Surrounding Environment.

Chapter Three

Project Conceptual Design

In this chapter, we are going to describe the detailed objectives of the system, the general block diagram and explain how the system works.

3.1 Project Objectives

This project supports many ideas and objectives that can be summarized as follows:

- Moving the mouse curser by eye tracking.
- Help handicapped and to move the normal PC computer mouse curser easily
- Creating an over view to other people who don't have background about eye tracking and its advantages.
- To provide the user ability to connect the pc computer with other devices and how to build suitable interface using appropriate programming language.

- Using eye tracking to control some hardware application such as closing and opening window .

- Using eye tracking to control some software application such as agent painter program .

3.2 Design Options

we faced my option in our project design, first the to select the tracking process with camera or sensor and the second in design the selection the motor type and its drive circuit , here we well type our selection in each part and the reasons to select our choose .

3.2.1 Camera Selecting :

In this section, the justifications the tow options :

- EOG sensors.
- Special purpose camera .

The first way That using EOG sensors to the face of the user connected to an electronic circuit directly to the Computer through Parallel or serial port, EOG sensors measurement of changes in electrical potential difference across a subjects face due to eye movement, measured via electrodes attached to compass points on the face (see figure below)

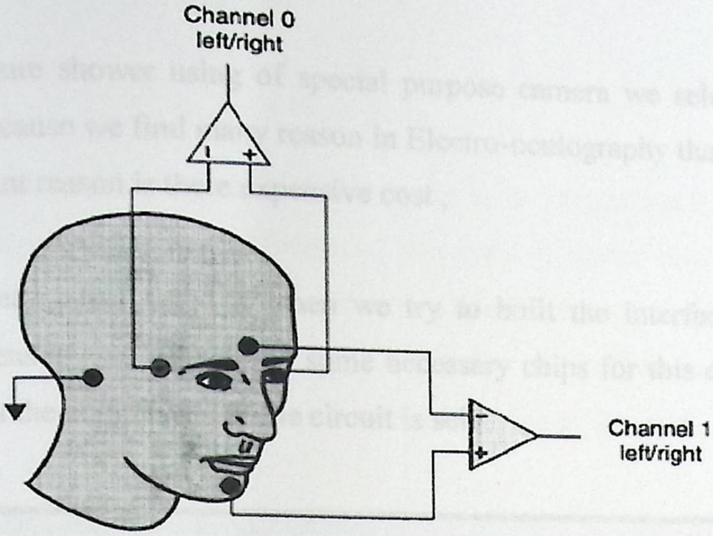


Fig 3.1 using of sensor to measure electoral potential

The second way is that it will comprise put special purpose camera that are sensitive to some cooler such that black and Wight , the function of this camera is to tracing the movement of the eyeball and every small interval of time the camera send an image to eyeball state to electrical circuit to emulate with basic pc mouse deeply the camera track the cooler of the eyeball and then out put of the circuit well we enter to parallel port and then work as basic mouse .

Fig 3.2 using of special purpose camera

The upper figure shower using of special purpose camera we select the second way to work in our project because we find many reason in Electro-oculography that make us avoid to use them the most important reason is there expensive cost ,

The second reason we faced it when we try to built the interface circuit between the computer and these sensor , we cant found some necessary chips for this circuit such as filtering and the there' reason is the complexity of the circuit is self .

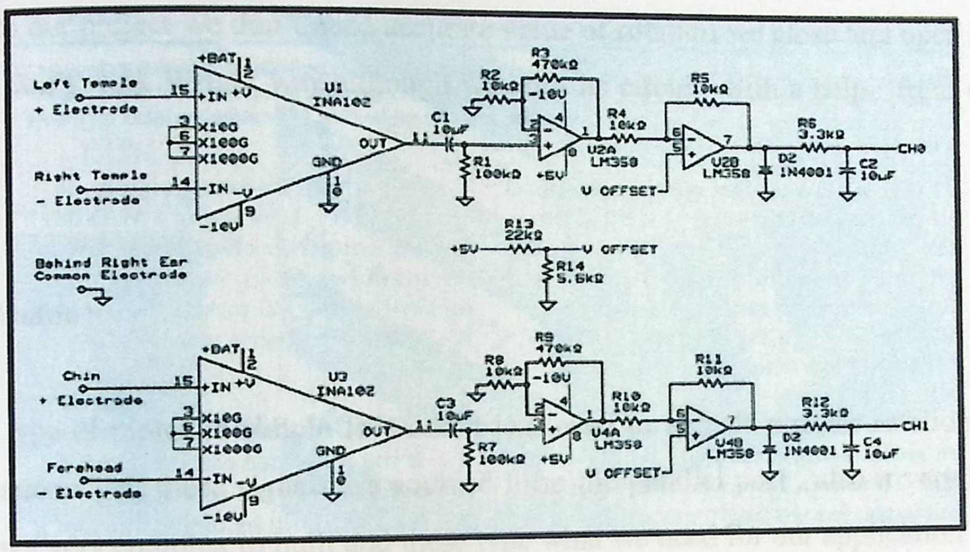


Fig 3.3 The complex circuit for EOG sensor

3.2.2 Motor Type :

In this section we faced two type of motor the first type is stepper motor and the second is the dc motor, we define each one advantages and disadvantages and mention our selection in the end of our working .

3.2.2 Stepper Motor :

The main advantage characteristic of this motor is its very accurate and its rotation defend by the angle of rotation but the disadvantage of this type of motor is its circuit driver is very complex to built and its work need some conditions such as it wanted 4 signal input "from parallel port in our project " and it rotation depending on the value delay between these signals and how many signals are activate in the same time, also its not very found like dc motor .

In response to our project we don't need accurate value of rotation we close and open accrete , and the circuit it don't work Wright way although we built its circuit with a help from electrical lab supervisor,

3.2.3 DC Motor :

This type of motor is simple to work it just need to signals one for rotation left and the other for rotation right, these signals are sourced form the parallel port , also it very fundable and its circuit is not very complex to built and these type what we need for our application .

3.3 Design Realization Approach (Implementation)

One of the most complicated parts to decide on is how to apply our idea of project on the real world, many approaches are used, implementation, modeling, or simulation.

3.4 Project Design Block Diagrams

The following Figure shows the block diagrams of our system

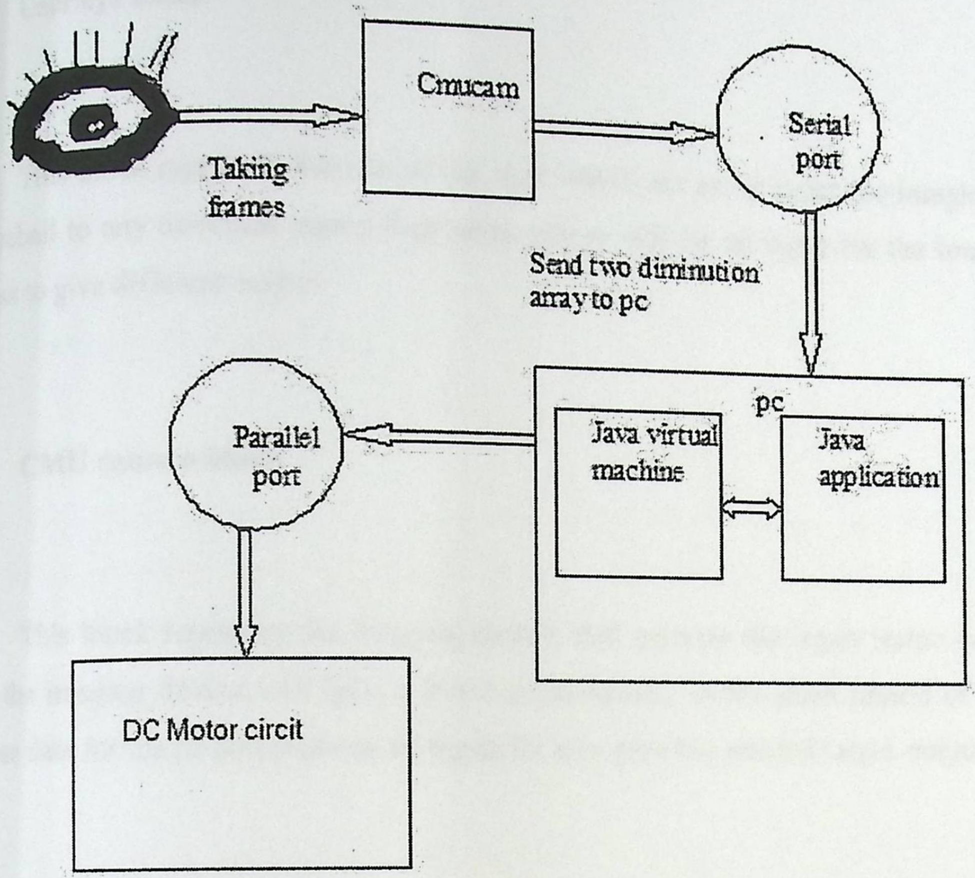


Fig 3.4 Block Diagram For The System

Now we will talk about the block diagram in more detail and the function for each block in the block diagram:

- **User eye block**

This block represent the eye of the user which act as an input for imaging device ,moving the eyeball to any direction means that anew frame will be an input for the imaging device to be process to give different output.

- **CMU camera block**

This block represent the imaging device that process the input frame come from the user eye , the imaging device will take a frame continuously every short period of time and give the process data for the pc computer as an input for it to give the needed target output.

- **Parallel port block**

This block used for transfer the process data to the pc computer, and its act as interface between the computer and imaging device and the curser of the mouse will move cording to signal in the each part, the signals in the ports will be pressed by the computer application that will we build it at and the result will determined the location of mouse curser.

- **PC computer block**

This block represent a normal pc computer that hold the development application we will build it in java language to do processing operation for data that come from the parallel port.

- **JAVA applications block :**

This block represent the desk top application where the user can select item and drag item, this can be performed by using threads in java language which mean that when we stopped on selected item some time the mouse function processed on this item, this bock also contain the agent painter application as we mention the painter let us to draw figure and colored it using the through the eye tracking application .

- **DC Motor application :**

This block represent the both the DC motor device and driver circuit, when the dc motor receive (1,0) from the two pens on the parallel port the DC motor is rotate left and when it receive a (0,1) signals form the parallel port in arrangement it rotate left and when it receive nether (0.0) signals form the parallel port it don't work .

3.5 Project Interacting with the Surrounding Environment

The result the end-user shall have with the eye mouse depends on certain assumptions regarding the user:

- The user has normal, healthy eyes that can be detectable as any other eye by the imaging device, can focus on a point for any amount of time, can blink, and can move smoothly.

- If the user is paralyzed, they will have a person around to attach the electrodes to the user's face.

- The user has a computer with a USB, PS/2, or serial port.

- The imaging device refreshes often enough to catch a blink the majority of the time from the user.

Success of the design may also be limited by the following factors:

- The imaging device does not have an infinite refresh rate and may not process some inputs.

- The mouse needs to be re-calibrated every time the user moves his or her head.

- A severely handicapped user will need assistance to use the device.

Chapter four

Detailed Technical Project Design

4.1 Discussing Design Options and Justifying those Chosen for the System.

4.2 Detailed Description of System Components.

4.3 Overall System Design.

4.4 User Interface Design.

Chapter four

Detailed Technical Project Design

Over View

In this chapter we will talk in details about system design including design options and specify why we use it in implementing our system.

4.1 Discussing Design Options and Justifying those Chosen For the System:

In this section, we will demonstrate design option for the system, and discuss why we choose it in implementing the system, in terms of programming language and CMU camera .

4.1.1 CMU Camera

CMU camera divided into version CMUcam1 and CMUcam2 both of them work in the same way but the differ in structure, This type of cam can be used for many systems and projects can be used for IRIS projects robot and eye tracking.

This type of cameras is designed for tracking system and its original work is in robot system specially playing football robot, the camera track the ball on area and give its location on the area to the control unit in the robot and this unit move the robot motor depend in that location, we edit this idea to eye tracking and so we choose this type of camera .

4.1.2 Programming Language :

We chose the Java programming language in our project because we familiar on it because we study it through our university study also its simple and worked in deferent platform, and the main reason for choose it is there some driver for the camera are written in java language and the way to test the camera if work or not is a java application .

4.1.3 Hardware Selection :

The idea of hardware come form the idea of the project , we design our project to special purpose people who don't have ability to use the muse with this hand, so we chose the hardware application to complete the soft ware function and help this person to not just control the computer using his eye , also he can do other things like open and close a window if he don't have ability to walk .

4.1.4 Serial Port:

Serial port is a serial communication physical interface through which information transfers in or out one bit at a time. Throughout most of the history of personal computers, data transfer through serial ports connected the computer to devices such as terminals or modems, keyboards, and other peripheral devices also connected in this way.

4.1.5 Parallel Port :

We chose the parallel port to be the output station not the serial port for the project because we continuously take data from the serial port and if we need to have an out put such that activate the DC motor and in the same time take frames form the camera this well making aloes for both output and input data, so we choose the parallel port.

4.2 Detailed Description of System Components:

This section contains detailed description about the component that used to develop our system.

4.2.1 CMU Camera:

The CMUcam was the basis of eye track We were given the CMUcam and told to design a system around it to test some of the camera's features and limitations. We chose to design eye tracking which we felt would leverage the strengths of the CMUcam1.

For our purposes we needed the camera to be able to track a moving eye a high enough frame rate so that we could get reasonably accurate enough information to determine the motion of the eye. The CMUcam tracks at a frame rate that appeared to be high enough to accurately track a moving eye within the parameters of our system.

4.2.1.1 Tracking A Color For Eye and How Does the CMUcam1 Do It

Color tracking is the ability to take an image for eye background , isolate a particular color and extract information about the location of a region of that image that contains just that color. As an example, assume that you are given a photograph that contains a red ball sitting on a dirt road. If Someone were to ask you to draw box around anything that was the color red in the image, you would quite, easily draw a rectangle around the ball.

This is the basic idea behind color tracking. You did not need to know that the object was a ball. You only needed to have a concept of the color red in order to isolate the object in the picture.

In this section we will briefly address how the CMUcam1actually uses the information in a camera image to perform color tracking.

In order to specify color, you need to define a minimum and maximum allowable value for each of those three color channels.
Every unique color is represented by a red, green, and blue value that indicates how much of each channel is mixed into that final color. The tricky part about specifying a color is that you need to define a range of allowable values for all three color channels.

Since light is not perfectly uniform and the color of an object is no perfectly uniform, you need to accommodate for these variations. However, you don't want to relax these bounds too much, or many unwanted colors will be accepted.

Since, in the case of the CMUcam1, each color channel is converted into a number between 16 and 240, you can bound each channel with two numbers,
an upper and lower limit.

If you have two limits for each of the three channels, this means that six values can be used to constrain the entire color space that you wish to track. If you imagine the colors being represented by a cube where each side is a different color channel (red, green and blue) then the six values used to select your color would draw a three dimensional box inside

Once you have a bound for the color you wish to track, the CMUcam1 takes these bounds and processes the image.

There are many ways to track colors in an image that can be quite complex. The CMUcam1 uses a simple one pass algorithm that processes each new image frame from the camera independently.

It starts at the top left of the image and sequentially examines every pixel row by row.

If the pixel it is inspecting falls inside the range of colors that the userspecified, it marks that pixel as being tracked. It also examines the position of the current tracked pixel to see if it is the top most, bottom most, left most or right most position of all the tracked pixel found thus far in the image.

If it finds that the pixel is outside of the current bounding box of the tracked region, it grows the bounding box to contain this new pixel. Because the location of even a single tracked pixel can change the bounding box, the bounding box can sometimes fluctuate quite a bit from frame to frame.

Noise filtering (see next paragraph) can be used to reduce some of that fluctuation.

The only other major piece of information that is stored is a sum of the horizontal and vertical coordinates of the tracked pixels. At the end the image you can take the horizontal sum and the vertical sum of the tracked pixels and divide each by the total number of tracked pixels, you get a value that shows where the middle of the tracked object is located. Because each tracked pixel only contributes a small part to the final horizontal and vertical sums the middle (often called the centric) of the tracked pixels is typically a much more stable measurement than the bounding box.

Once all of the pixels in the image have been checked, the total number of tracked pixels can also be used in conjunction with the area of the bounding box to calculate the confidence of the tracked object.

Noise filtering allows us to make the color tracking ranges larger so we can accommodate larger variations in the image pixel values without causing other random variations in the image to be tracked. The idea behind noise filtering is that we only want to consider a pixel to be of the tracked color if it is part of a group of pixels that are within the color tracking bounds.

Again in the CMUcam1we implement this in a way that only requires a single pass over the image. While processing the pixels in an image the CMUcam1 maintains a counter which keeps of track of how many sequential pixels in the current row, before the current pixel were within the tracked
color bounds.

If that value is above the noise filter value then the current pixel is marked as a tracked pixel.

Fig 4.1 color tracking

4.2.1.2 How does the CMUcam1 do Frame Differencing :

Frame differencing is a method of identifying changes in a series of images. Given multiple images at different times from the same or similar view points, it is possible to compare them in order to isolate objects that may have moved.

Using the CMUcam1's frame differencing functionality is a good way to detect and track such motion in a scene. Instead .of storing an entire image, the CMUcam1 stores an abstraction of the image. Using a similar process to color tracking,

The CMUcam1 will generate or compare the image on a line by line basis as it receives the data.

Example:

The CMUcam1 internally represents a reference image as an array of 8 by 8 bytes. Each element of this array stores the average of a corresponding region on the main camera image. The default setting uses the green or intensity channel, but this can be changed for situations where one channel clearly shows more variation than the others. When a new image is read in, it is also converted into an array of 8x8 bytes. To look for a change, each block in the 8x8 grid is subtracted from the corresponding reference image block.

If there is more than a specified threshold, a change is flagged. The rest of the data, such as middle mass, is calculated in an almost identical manner to the way it is in color tracking.

4.2.2 Serial Port :

Serial port used to received image from camera and send to computer.

The serial port is an Asynchronous port which transmits one bit of data at a time, usually connecting to the UART Chip.

Serial Ports are commonly found on the majority of PC Compatible computers. Usually referred to as a DB9 or DB25 connection, both of which adhere to the RS-232c interface standard and defined in ISO 2110 and ISO 4902. D represents the shape of the connector if placed vertically as shown in the below illustrations.

The number 9 / 25 indicating the number of pins found on the connector. DB9 Serial connections are now commonly found on modern PCs where DB25 is commonly found on older computers.

4.2.3 Computer device:

When the image received by computer after transfer by serial port the user run the java program is built for determine the position of object by more one algorithm like histogram algorithm and determine the x position and y position the mouse cur cure moving in monitor of computer.

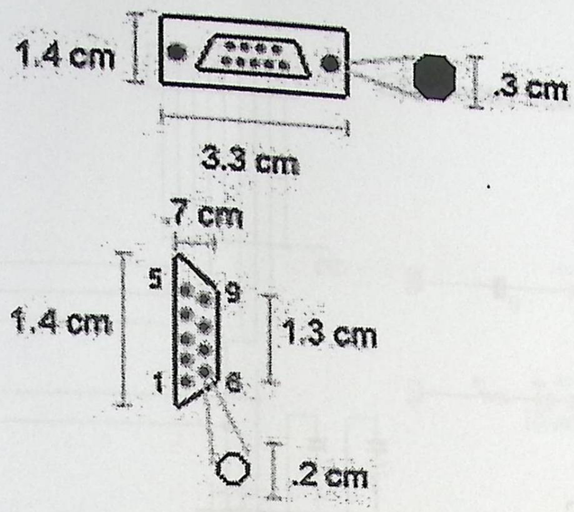


Fig 4.2 serial port

4.3 Overall System Design :

CMUcam Schematic

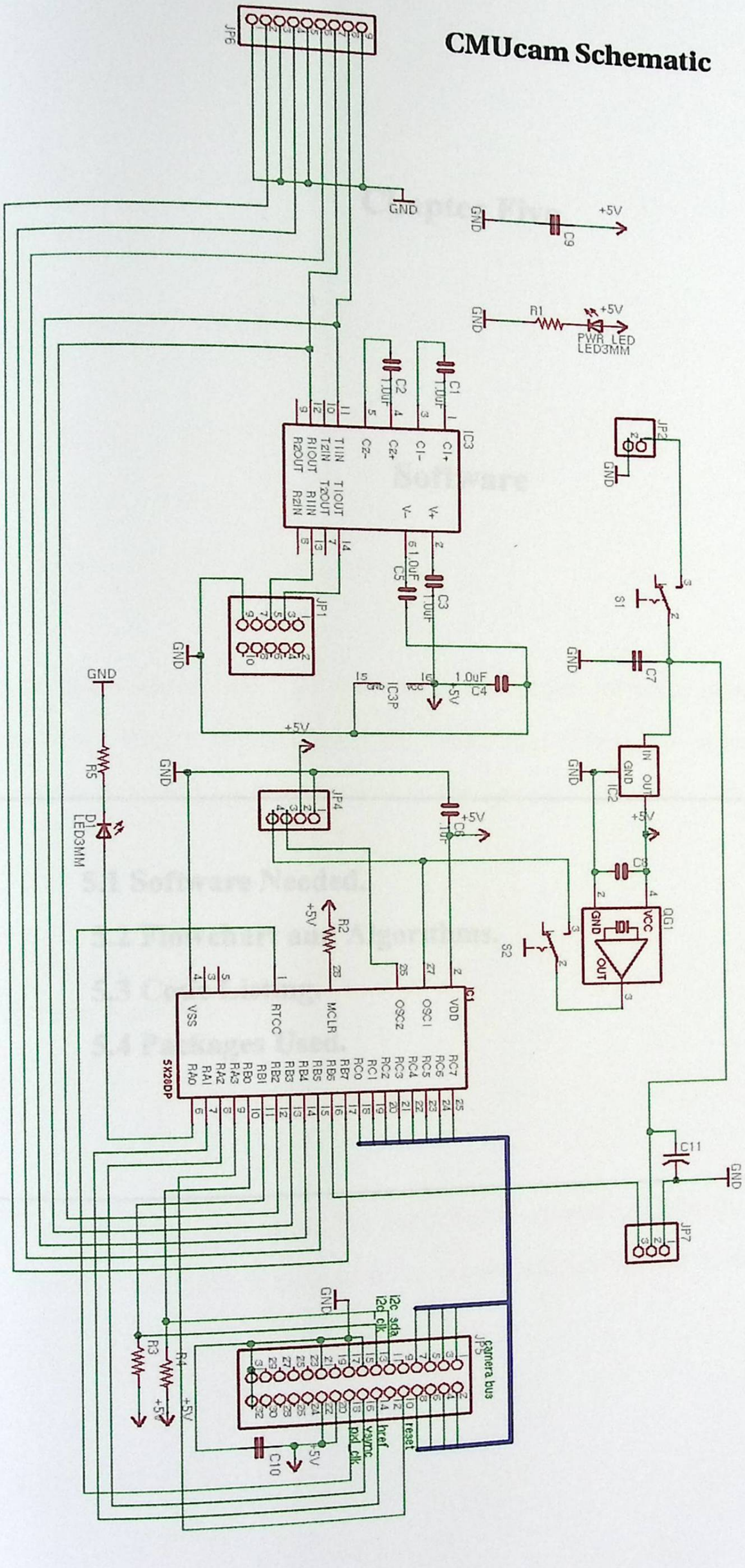


Figure (4.3) CMUcam Schematic

Chapter Five

Software

5.1 Software Needed.

5.2 Flowchart and Algorithms.

5.3 Code Listing.

5.4 Packages Used.

Chapter Five

Software

In this chapter we will talk well talk about software needed for the project and any other topics related such as, code, algorithms, flowcharts and how these component are work to gather to give us the specific function of the project .

5.1 Software Needed:

Here we well talk about the main software required for the project in order to work correctly and make it easy to us to create and testing the code.

5.1.1 Windows XP:

Rally the software the basic thing in our project , because we need an operating system for the pc that carry the application software .

5.1.2 Microsoft Visio:

This program help us to draw the needed flowchart and block diagram for the project , as we know these thing is important in order to understand the function of the project and how the code of the project are work .

5.1.3 Java Virtual Machine:

We must have new version of java virtual machine, this software is important for our project because our project application interface program is written in java language so, we need this software in order to run our application .

5.1.4 JCreator Pro:

This program is one of many program used to write the java code, we chose it because we learn the java language in our study in the university using this program so we are familiar with it .

5.1.5 JMF Demo:

This program help us in the testing state for the camera, and its java applet application that support multimedia device such as camera, microphone, specially you can used it when your device is connected through serial com

5.2 Flowchart and Algorithms:

In section we well talk about the basic flowchart for the system and other flowcharts also we well talk about the algorithm used in the tracking in the camera we are used in our system .

5.2.1 Basic Flowchart For the System :

This flowchart show clearly how the systems are work and basic action the system perform after occurring and specific event.

After starting the applet application we take an initial frame, this frame help us to select the cooler we want to track, the camera store the initial x and y witch mean the canter of the cooler (object) because the cooler we want to track it should be cooler homogenous.

The camera start taking frames rapidly and start processes each frame individually ,and send the new x and y to applet application after each frame , the applet application take the new value of the x and y and compare them to initial values of x and in order .

If the values are equal no operation are perform , else the applet application are entered into second compare operation, this operation importance is to test if the New value of the x and y are in the range of sensitive area that make the parallel port give eclectic signal .

If the values are in the range the mouse cursor are change its location to the new x and y value in the applet application screen and in parallel the alarm circuit are sourced by electrical signal from the parallel port so the alarm system well start .

And if the new values of x and y are not equal the applet application just move the mouse Cursor to new x and y location and these operation are performed readily unless we end the program .

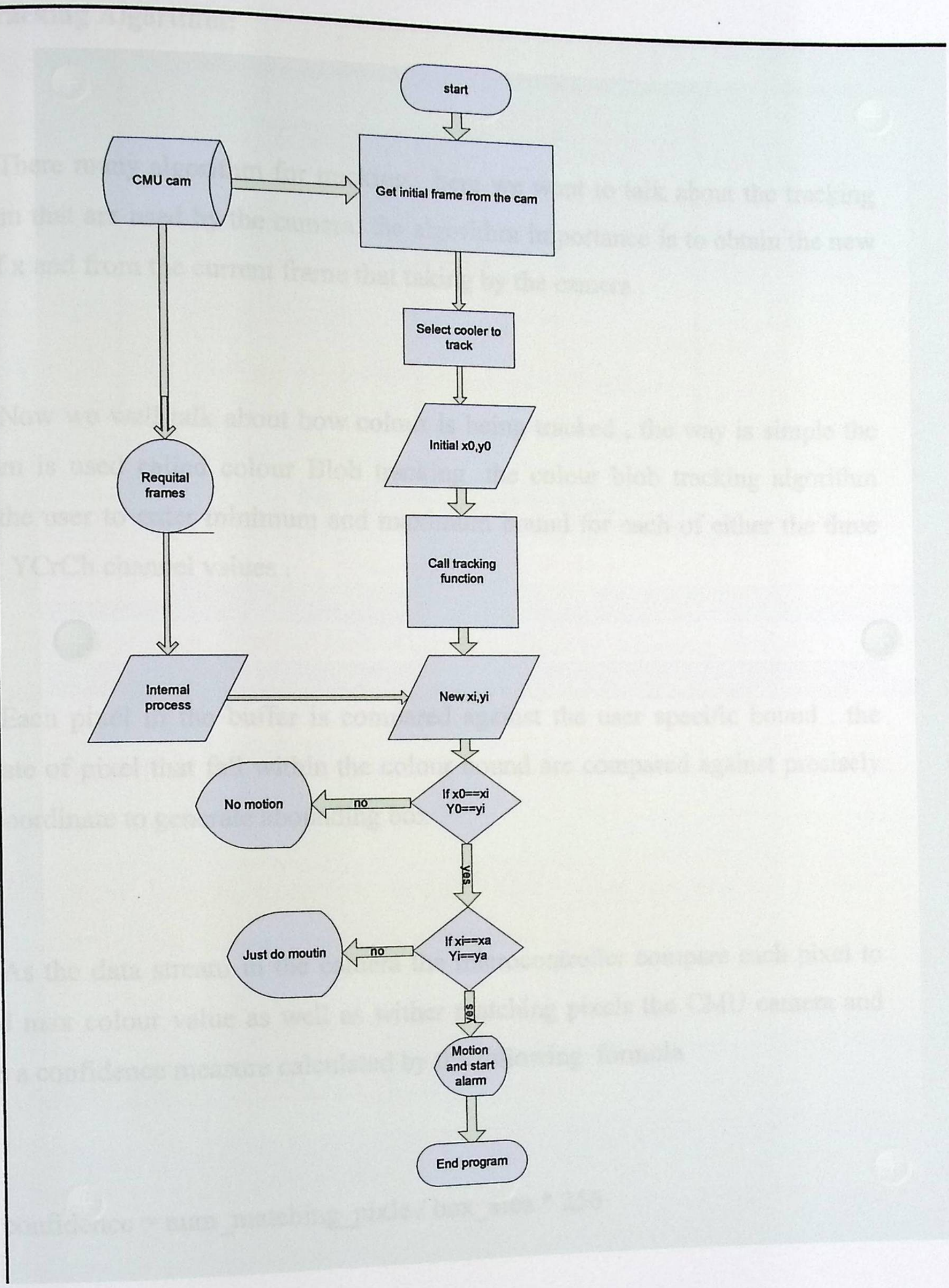


Fig 5.1 basic flow chart for the system

5.2.2 Tracking Algorithm:

There many algorithm for tracking , here we want to talk about the tracking algorithm that are used by the camera, the algorithm importance is to obtain the new value of x and from the current frame that taking by the camera .

Now we well talk about how colour is being tracked , the way is simple the algorithm is used called colour Blob tracking .the colour blob tracking algorithm allows the user to enter minimum and maximum bound for each of either the three RGB or YCrCb channel values .

Each pixel in the buffer is compared against the user specific bound . the coordinate of pixel that fall within the colour bound are compared against precisely stored coordinate to generate abounding box .

As the data stream in the camera the microcontroller compare each pixel to min and max colour value as well as wither matching pixels the CMU camera and produce a confidence measure calculated by the following formula

confidence = num_matching_pixle / box_area * 256

The centre of an identified blob is also calculated by summing all the x and y coordinates of the matched pixels and diving that by the number of detected pixels .

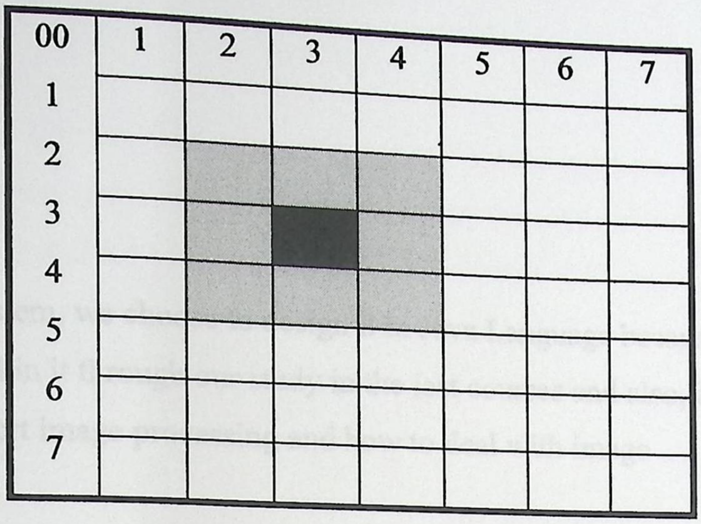


Fig 5.2 cooler blob tracking

The center of an identified blob is also calculated by summing all x and y coordinates e of the match pixel and dividing that value by the number of detected pixel .

The above illustrating representation an 8X8 bounding box with a 3*3 "blob" , here the centriod calculation would look as the follows:

Centriod_x = (2+3+4)*3=27/9=3

Centriod_y = (2+3+4)*3=27/9=3

So the centriod is the illustration is located at (3,3) and is represented by dark pixel .

5.3 Code Listing:

In our system, we choose to design it in Java Language because we have good background in it through our study in the last courses and also, it have some library that support image processing and how to deal with image .

5.3.1 The Agent Painter:

This application used the thread in java language , so when we stopped the cursor on some item to interval of time the java application check if the time of the thread is end and then if the location is don't change the function of the mouse on that item is processed, such that click or released .

First we built the painter and then we connected the x and y coordinate that the painter is used with eye tracking x and y coordinate output so its take its new value of coordinate from the tracking process, the new thing in the painter is the used of threads .

5.3.2 The desktop application:

This application is similar to the painter in using thread system but the deferent between them is that when we used the desktop application we used some windows function such that click start it need some command to be processed in the same windows .

5.3.3 DC Motor Code:

In this section we write the code for the needed operation form the motor , rotate left and rotate write and don't do any thing this code is done by activate tow parallel port pines and the value of these pines is control the function of the motor, so if we entered (0,1) vale the motor is rotate forward and if we enter the value is (1,0) the motor is rotate backward and if we enter (0,0) the motor don't do any thing .

Chapter Six

Implementation and System Testing

6.1 Implementation.

6.2 System Testing.

Chapter Six

Implementation and System Testing

6.1 Implementation

In this chapter we will talk about the most important tasks in our system to work as needed, which is implementation and system testing.

6.1.1 Setting up the Required Software and Hardware:

Here we will demonstrate the required software and hardware that must be setting to implement our system.

6.1.2 Setting Up The Hardware and Operating Systems:

The hardware equipments needed for the operation of this system are:

- A personal computer.
- CMU cam version one type six.
- Breadboard.
- Relay.
- Transistor (NPN).
- Resistors (100,220 Ω each).
- DC Motor
- Optocupler (4N25)
- Transistor (NPN "BD243C") and (PNP "BD244C")
- A 12V power supply and connection wires.

The operating systems that must be installed on the PC is Microsoft Windows XP(R) Professional.

6.1.3 Setting Up The system software:

The software packages that this system is built on are:

- Microsoft Windows XP Professional.
- Java Virtual Machine (j2sdk1.4.2_14).
- Java creator.
- Input/output packages for serial and parallel ports.

6.1.4 Actual System Implementation

A simple prototype of the project is shown in figure (6.1)

It shows the following parts:

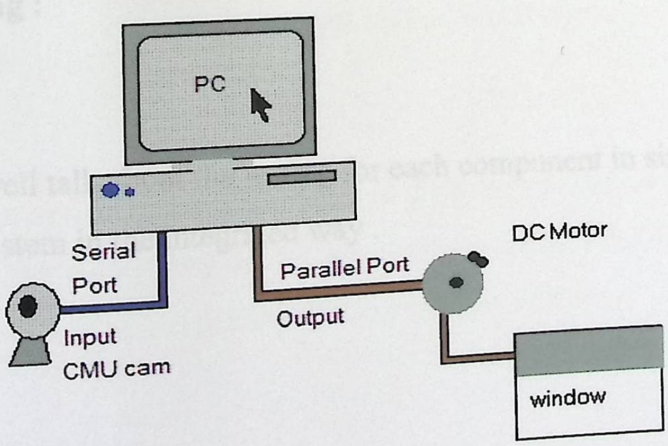


Fig 6.1 simple prototype of the project

- PC that we used in the project which we built the Java Program in it and the operation system used in our project is Windows XP.

- Serial Port: which is the input port used, it receives the signal from the CMU camera and sends it to the PC, the data is fetched by the software (Java application) .

- CMU Cam: which gets the picture from the moving object to be tracked every 0.23 second .

- Parallel Port : are used as out put for this project , the pins of port are reactive when the cursor receive specific area in applet screen .

- DC Motor : and used to control the window application so that close and open according to the signals come form the parallel port .

6.2 System Testing :

Here we well talk about the testing for each component in single way and then testing the system in the integrated way .

6.2.1 CMU camera Testing :

First of all we should connect the camera correctly after understand it datasheet and its port and its power port, the camera have there port and we well use the RS-232 serial port and its is the normal pc port .

The following figure show the serial cable for connection between the camera and the computer port .

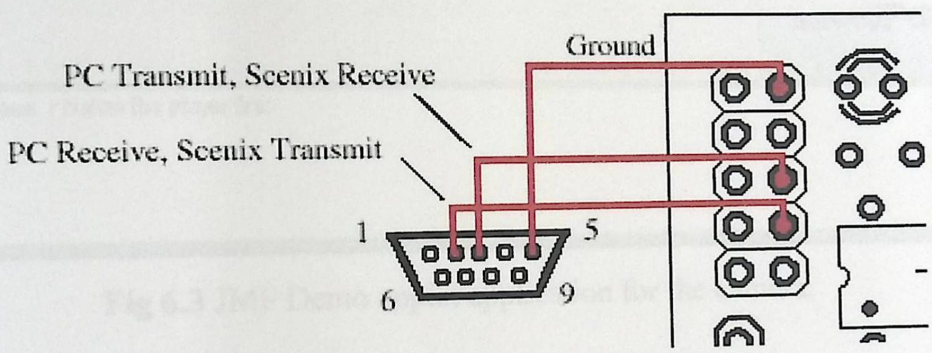


Fig 6.2 level shifted serial port

We must be carefully when we connect the power to the camera because its very sensitive and the wrong way my be destroy the camera, after do these steps we can move to next state which is getting frame form the camera .

You well receive an testing applet application for the camera when you buy it this code is called (JMF Demo) java frame demo this application give you the ability to have single frame for the camera .

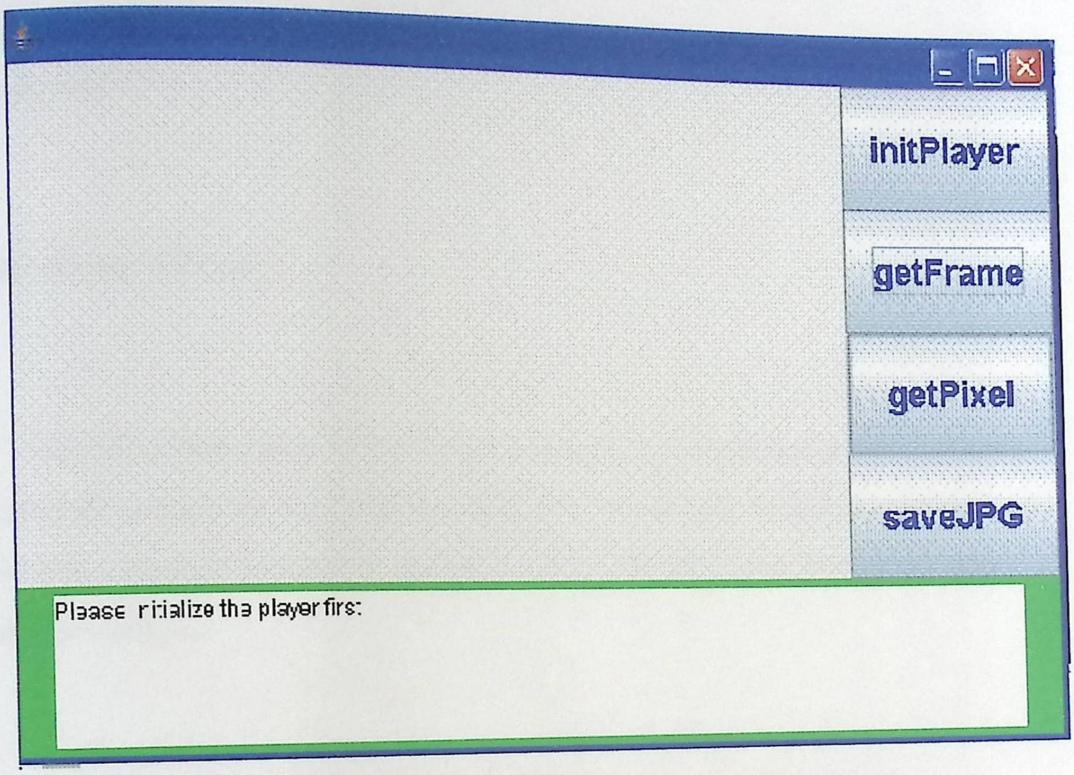


Fig 6.3 JMF Demo applet application for the camera.

You first chose initiate player , to test if the camera are connect of not to the pc ,if you correctly connect the camera you we have the massage in the teat area box as ("Player initialized ") .

Then you can chose the following bottom ("getFrame") , this bottom save the image in the buffer of the camera, and in the same time you can see the selected image .

You can chose ("getpixel") bottom to take the x and y coordinate for ant pixel you chose it in the image, at least you ("saveframe") bottom to save the frame to you pc the serial com .

In our project we pass this stat in excellent result , so test that our camera is work correctly .

6.2.2 DC Motor Circuit:

The following figure is the way to connect this circuit and the power input for this circuit is come from the pc parallel port , we can use anther power supply to resource this circuit if we need to use high voltage device , more than (4-5) volt parallel port voltage .

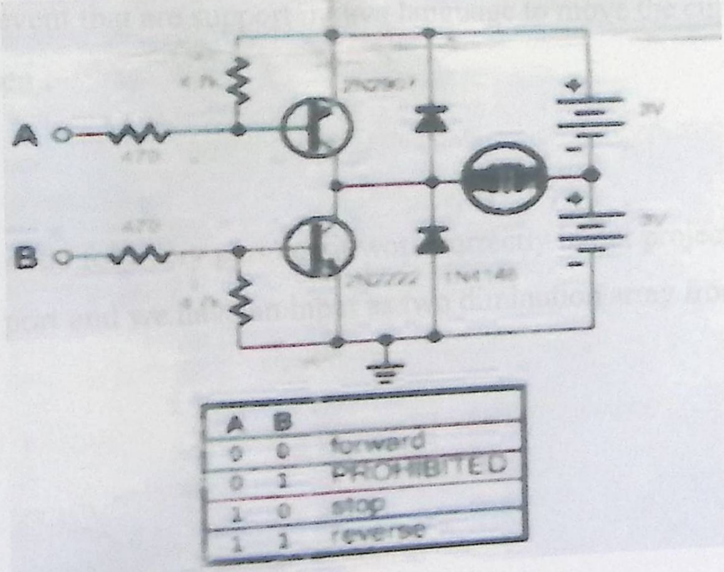


Fig 6.4 DC motor system circuit

6.2.3 Code Testing :

The code testing is dive in tow part code to active parallel port witch responsible for the output part of the project , the code function is the following , when the mouse curser enter specific area in the applet screen the parallel port we be active

We can do that by suppose that this area is an image in the applet application and save this image into two diminution array when we entered this area we have active parallel port .

The second part of the code is to act as an input for the pc , the jenral work of the camera it to give us the new coordinate for the object, when we take these va;ues we can use mouse event that are support in java language to move the cursor along the java applet screen .

We test the code for every part and it work correctly in our project, we have data in the parallel port and we have an input as two diminution array from the camera .

6.2.4 Integrated System Testing :

Here we well try to test the complete system to gather , we well connect the camera as we mention in the last and connect the alarm circuit to the parallel port we well do the following steps in order :

- Start the applet application for the project
- Start on the power of camera , after check the power connection
- Connect the alarm system to the parallel port
- Locate the camera in suitable place in front of the eye of the user
- Take frame for the poll of the eye of the user in its first stare
- Select this ball as an object for tracking
- Call the tracking function for this object

In the final we can say that , we have an accusable result for our work in this state , although we have noise that come form the non stop eye motion .

Chapter Seven

Conclusions and Future work

Chapter Seven

Future Development and Conclusions

In this chapter we will talk about some of the modifications that can be done over our system in the future, and the conclusion that we derived it.

7.1 Future work :

Our system can be modified on the future in many ways to achieve more technical systems, some of this modifications are.

- Improving the system to be deal with more motion of the mouse cursors but do sum of general mouse function such as right click and left click and select and other general mouse action .

- Improving the system cursor to be received all the tracked screen pixel.
- We improve the system to be worked with public program such as office.
- We can improve the system to reduce the amount of noise on new x and y new value coordinate .
- We can improving the system hardware to deal with more device rather than the DC motor and do more complex applications rather than open and close a window.

7.2 Conclusions :

By ending the implementation of this system, we become familiar with the component that used on it, we learn many things to implement the system, some of these are:

- We programming our system in java language, so we become specialist in java programming and we also now that the language is newer and important in our days .
- The main idea of our project talk about image processing , so we become familiar with image processing such as image comparing ,how to fined the histogram ,deal with the image as an integer array .

- In our project we interface between the camera input and the general mouse function using serial port connection, so we now become familiar we the hardware connection using this port .

- We become familiar with principle of tracing system, so we can in the future design complex system that using tracking principle such as robots tracking system .

References

[1]: Lei Zhang, Yanfeng Sun, Mingjing Li, Hongjiang Zhang .Automated red-eye detection and correction in digital photographs

[2]: Mazen M Kharbutli, Aristos I Aresti, Ali A El-Haj-Mahmoud. Parallelization of Digital Image Processing Algorithms

[3]: Local Markets

[4]: http://www.cs.cmu.edu

[5]: Digital Image Processing, Second Edition (283, 6.1)

[6]: Digital Image Processing,, Second Edition (286, 6.1)

[7]: http://www.sketchpad.net/

[8]: Digital Image Processing, Second Edition (1, 1.1)

[9]: Digital Image Processing, First Edition, (7, 1.3)

[10]: Digital Image processing with FilterMeister, Werner D. Streidt

[11]: http://vbaccelerator.com/codelib/gfx/imgproc.htm

[12]: Lewis & Loftus's textbook Java Software Solutions

[13]: The Image Processing Handbook, (7, 1.1)

[14]: http://computer.howstuffworks.com/

[15]: www.microsoft.com

[16]: MSDN Library for Visual Studio 2005

Appendices

Appendix A
Appendix B

Appendix A

(Datasheets)

BD243B/BD243C
BD244B/BD244C

COMPLEMENTARY SILICON POWER TRANSISTORS

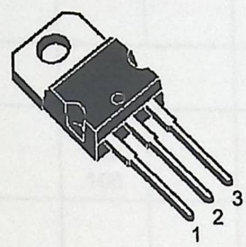
- STMicroelectronics PREFERRED SALESTYPES

DESCRIPTION

The BD243B and BD243C are silicon Epitaxial-Base NPN transistors mounted in Jedec TO-220 plastic package.

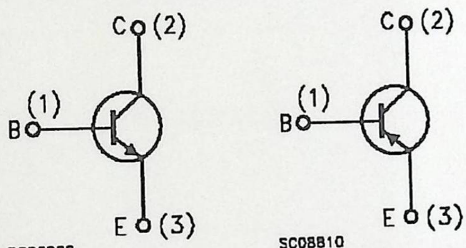
They are inteded for use in medium power linear and switching applications.

The complementary PNP types are BD244B and BD244C respectively.



TO-220

INTERNAL SCHEMATIC DIAGRAM



ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Value		Unit
	NPN	BD243B	BD243C	
	PNP	BD244B	BD244C	
V_{CBO}	Collector-Base Voltage ($I_E = 0$)	80	100	V
V_{CEO}	Collector-Emitter Voltage ($I_B = 0$)	80	100	V
V_{EBO}	Emitter-Base Voltage ($I_C = 0$)	5		V
I_C	Collector Current	6		A
I_{CM}	Collector Peak Current	10		A
I_B	Base Current	2		A
P_{tot}	Total Dissipation at $T_c \leq 25\ ^oC$	65		W
T_{stg}	Storage Temperature	-65 to 150		^oC
T_j	Max. Operating Junction Temperature	150		^oC

For PNP types voltage and current values are negative.

THERMAL DATA

$R_{thj-case}$	Thermal Resistance Junction-case	Max	1.92	°C/W
$R_{thj-amb}$	Thermal Resistance Junction-ambient	Max	62.5	°C/W

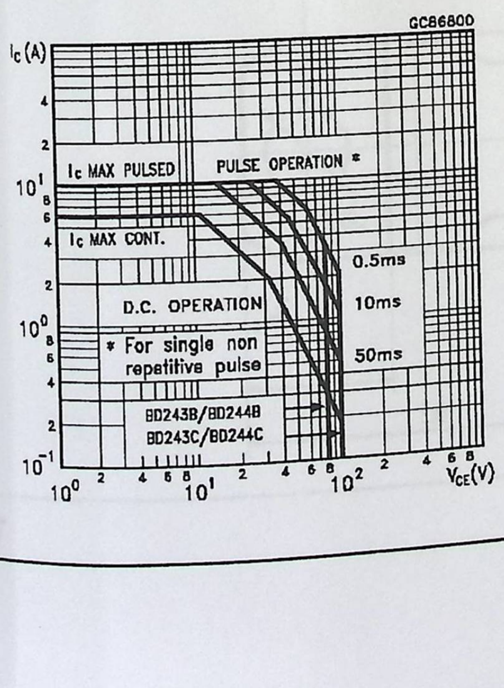
ELECTRICAL CHARACTERISTICS (T_{case} = 25 °C unless otherwise specified)

Symbol	Parameter	Test Conditions	Min.	Typ.	Max.	Unit
I_{CES}	Collector Cut-off Current (V_{BE} = 0)	V_{CE} = rated V_{CEO}			0.4	mA
I_{CEO}	Collector Cut-off Current (I_B = 0)	V_{CE} = 60 V			0.7	mA
I_{EBO}	Emitter Cut-off Current (I_C = 0)	V_{EB} = 5 V			1	mA
$V_{CEO(sus)}$*	Collector-Emitter Sustaining Voltage (I_B = 0)	I_C = 30 mA for **BD243B/BD244B** for **BD243C/BD244C**	 80 100			 V V
$V_{CE(sat)}$*	Collector-Emitter Saturation Voltage	I_C = 6 A I_B = 1 A			1.5	V
V_{BE}*	Base-Emitter Voltage	I_C = 6 A V_{CE} = 4 V			2	V
h_{FE}*	DC Current Gain	I_C = 0.3 A V_{CE} = 4 V I_C = 3 A V_{CE} = 4 V	30 15			
h_{fe}	Small Signal Current Gain	I_C = 0.5 A V_{CE} = 10 V f = 1MHz I_C = 0.5 A V_{CE} = 10 V f = 1KHz	3 20			

* Pulsed: Pulse duration = 300 µs, duty cycle ≤ 2 %

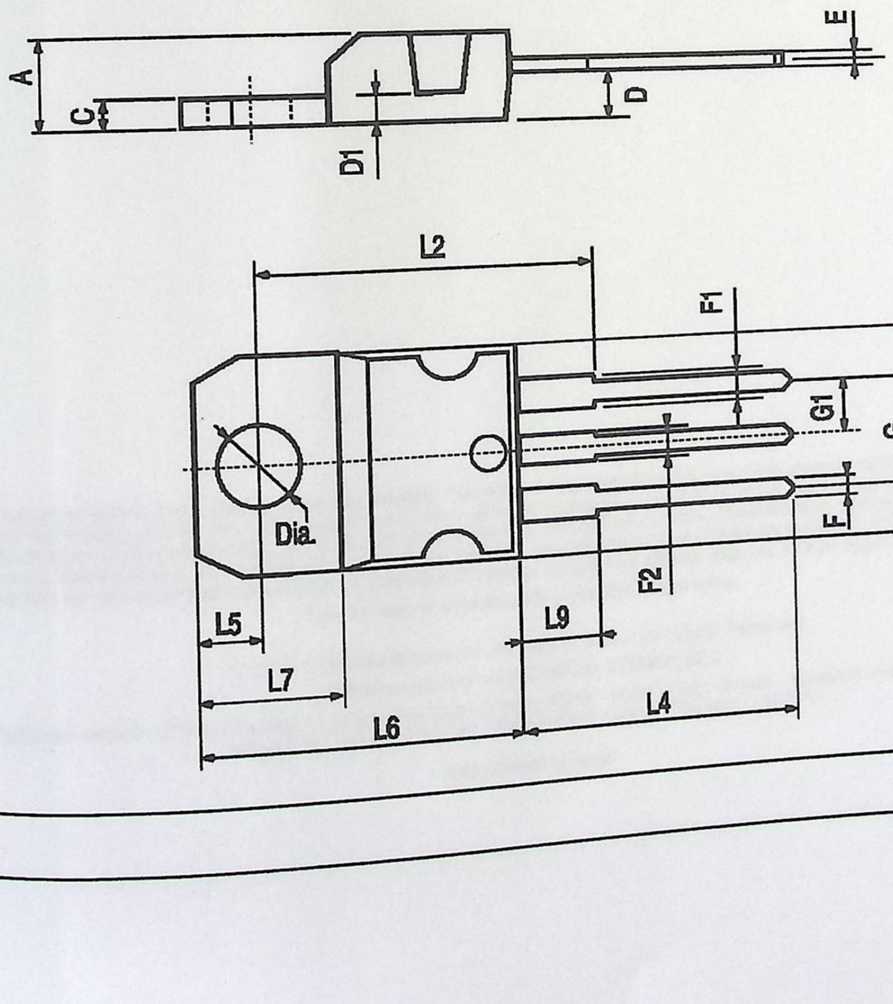
For PNP types voltage and current values are negative.

Safe Operating Area



TO-220 MECHANICAL DATA

DIM.	mm			inch		
	MIN.	TYP.	MAX.	MIN.	TYP.	MAX.
A	4.40		4.60	0.173		0.181
C	1.23		1.32	0.048		0.051
D	2.40		2.72	0.094		0.107
D1		1.27			0.050	
E	0.49		0.70	0.019		0.027
F	0.61		0.88	0.024		0.034
F1	1.14		1.70	0.044		0.067
F2	1.14		1.70	0.044		0.067
G	4.95		5.15	0.194		0.203
G1	2.4		2.7	0.094		0.106
H2	10.0		10.40	0.393		0.409
L2		16.4			0.645	
L4	13.0		14.0	0.511		0.551
L5	2.65		2.95	0.104		0.116
L6	15.25		15.75	0.600		0.620
L7	6.2		6.6	0.244		0.260
L9	3.5		3.93	0.137		0.154
DIA.	3.75		3.85	0.147		0.151



P011C

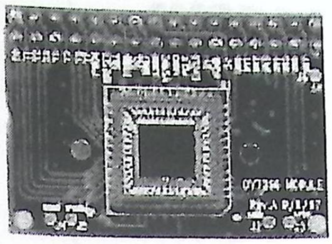
STMicroelectronics GROUP OF COMPANIES

Australia - Brazil - China - Finland - France - Germany - Hong Kong - India - Italy - Japan - Malaysia - Malta - Morocco - Singapore - Spain - Sweden - Switzerland - United Kingdom - U.S.A.

http://www.st.com

C3088
1/4" Color Camera Module With Digital Output

General Description



The C3088 is a 1/4" color camera module with digital output. It uses OmniVision's CMOS image sensor OV6620. Combining CMOS technology together with an easy to use digital interface makes C3088 a low cost solution for higher quality video image application.

The digital video port supplies a continuous 8/16 bit-wide image data stream. All camera functions, such as exposure, gamma, gain, white balance, color matrix, windowing, are programmable through I²C interface.

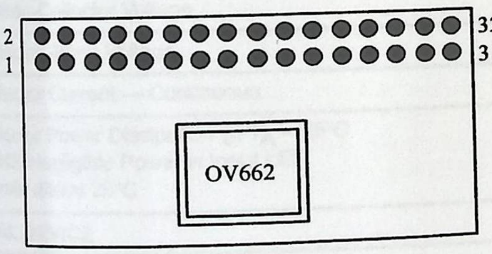
In combine with OV511+, USB controller chip, it will be easily form a USB camera for PC application.

Features:

- 101,376 pixel, CIF/QCIF format
- Small size : 40 x 28 mm
- Lens: f=4.9mm (Optional)
- 8/16 bit video data : CCIR601, CCIR656, ZV port
- Read out - progressive
- Data format - YCrCb 4:2:2, GRB 4:2:2, RGB
- I²C interface
- Wide dynamic range, anti blooming, zero smearing
- Electronic exposure / Gain / White balance control
- Image enhancement - brightness, contrast, gamma, saturation, sharpness, window, etc
- Internal / external synchronization scheme
- Frame exposure / line exposure option
- Single 5V operation
- Low power consumption (<100mW)
- Monochrome composite video signal output(50 Hz)

Specification

Imager	OV6620, CMOS image sensor
Array Size	356x 292 pixels
Pixel size	9.0 x 8.2 μm
Scanning	Progressive
Effective image area	3.1mm x 2.5mm
Electronic Exposure	500:1
Gamma Correction	0.45/0.55/1.0
S/N Ratio	48dB
Min Illumination	3lux @F1.2
FPN	<0.03% Vp-p
Dark current	<0.2 nA/cm²
Dynamic Range	72dB
Operation Voltage	5 VDC
Operation Current	80mW Active 30 μW Standby
Lens (Optional)	F4.9mm, F2.8



PCB Layout (Top side)

Pin Description

1~8	Y0~Y7	Digital output Y Bus.
9	PWDN	Power down mode
10	RST	Reset
11	SDA	I²C Serial data
12	FODD	Odd Field flag
13	SCL	I²C Serial clock input
14	HREF	Horizontal window reference output
15	AGND	Analog Ground
16	VSYN	Vertical Sync output
17	AGND	Analog Ground
18	PCLK	Pixel clock output
19	EXCLK	External Clock input (remove crystal)
20	VCC	Power Supply 5VDC
21	AGND	Analog Ground
22	VCC	Power Supply 5VDC
23~30	UV0-UV7	Digital output UV bus.
31	GND	Common ground
32	VTO	Video Analog Output (75Ω monochrome)

Application Example

- Video Conferencing
- PC Multimedia
- Video Phone
- Video Mail
- Still Image
- Machine Vision
- Process control

Note: Evaluation Board is available for C3088

Order this document
by 4N25/D

GlobalOptoisolator™

VDE | UL | CSA | SETI | SEMKO | DEMKO | NEMKO | BABT

6-Pin DIP Optoisolators Transistor Output

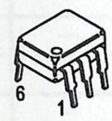
4N25*
4N25A*
4N26*
[CTR = 20% Min]
4N27
4N28
[CTR = 10% Min]

*Motorola Preferred Devices

The 4N25/A, 4N26, 4N27 and 4N28 devices consist of a gallium arsenide infrared emitting diode optically coupled to a monolithic silicon phototransistor detector.

- Most Economical Optoisolator Choice for Medium Speed, Switching Applications
- Meets or Exceeds All JEDEC Registered Specifications
- ***To order devices that are tested and marked per VDE 0884 requirements, the suffix "V" must be included at end of part number. VDE 0884 is a test option.***

STYLE 1 PLASTIC

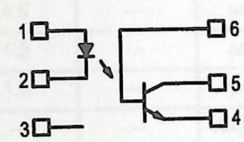


STANDARD THRU HOLE
CASE 730A–04

Applications

- General Purpose Switching Circuits
- Interfacing and coupling systems of different potentials and impedances
- I/O Interfacing
- Solid State Relays

SCHEMATIC



PIN 1. LED ANODE
2. LED CATHODE
3. N.C.
4. EMITTER
5. COLLECTOR
6. BASE

MAXIMUM RATINGS (T_A = 25°C unless otherwise noted)

Rating	Symbol	Value	Unit
INPUT LED			
Reverse Voltage	V_R	3	Volts
Forward Current — Continuous	I_F	60	mA
LED Power Dissipation @ T_A = 25°C with Negligible Power in Output Detector	P_D	120	mW
Derate above 25°C		1.41	mW/°C
OUTPUT TRANSISTOR			
Collector–Emitter Voltage	V_{CEO}	30	Volts
Emitter–Collector Voltage	V_{ECO}	7	Volts
Collector–Base Voltage	V_{CBO}	70	Volts
Collector Current — Continuous	I_C	150	mA
Detector Power Dissipation @ T_A = 25°C with Negligible Power in Input LED	P_D	150	mW
Derate above 25°C		1.76	mW/°C
TOTAL DEVICE			
Isolation Surge Voltage(1) (Peak ac Voltage, 60 Hz, 1 sec Duration)	V_{ISO}	7500	Vac(pk)
Total Device Power Dissipation @ T_A = 25°C Derate above 25°C	P_D	250 2.94	mW mW/°C
Ambient Operating Temperature Range(2)	T_A	–55 to +100	°C
Storage Temperature Range(2)	T_{stg}	–55 to +150	°C
Soldering Temperature (10 sec, 1/16" from case)	T_L	260	°C

1. Isolation surge voltage is an internal device dielectric breakdown rating.
   For this test, Pins 1 and 2 are common, and Pins 4, 5 and 6 are common.
2. Refer to Quality and Reliability Section in Opto Data Book for information on test conditions.

Preferred devices are Motorola recommended choices for future use and best overall value.
GlobalOptoisolator is a trademark of Motorola, Inc.

MOTOROLA

REV 5

ELECTRICAL CHARACTERISTICS (T_A = 25°C unless otherwise noted)(1)

Characteristic		Symbol	Min	Typ(1)	Max	Unit
INPUT LED						
Forward Voltage (I_F = 10 mA)	T_A = 25°C	V_F	—	1.15	1.5	Volts
	T_A = –55°C		—	1.3	—	
	T_A = 100°C		—	1.05	—	
Reverse Leakage Current (V_R = 3 V)		I_R	—	—	100	μA
Capacitance (V = 0 V, f = 1 MHz)		C_J	—	18	—	pF
OUTPUT TRANSISTOR						
Collector–Emitter Dark Current (V_{CE} = 10 V, T_A = 25°C	4N25,25A,26,27	I_{CEO}	—	1	50	nA
	4N28		—	1	100	
(V_{CE} = 10 V, T_A = 100°C)	All Devices	I_{CEO}		1	—	μA
Collector–Base Dark Current (V_{CB} = 10 V)		I_{CBO}	—	0.2	—	nA
Collector–Emitter Breakdown Voltage (I_C = 1 mA)		$V_{(BR)CEO}$	30	45	—	Volts
Collector–Base Breakdown Voltage (I_C = 100 μA)		$V_{(BR)CBO}$	70	100	—	Volts
Emitter–Collector Breakdown Voltage (I_E = 100 μA)		$V_{(BR)ECO}$	7	7.8	—	Volts
DC Current Gain (I_C = 2 mA, V_{CE} = 5 V)		h_{FE}	—	500	—	—
Collector–Emitter Capacitance (f = 1 MHz, V_{CE} = 0)		C_{CE}	—	7	—	pF
Collector–Base Capacitance (f = 1 MHz, V_{CB} = 0)		C_{CB}	—	19	—	pF
Emitter–Base Capacitance (f = 1 MHz, V_{EB} = 0)		C_{EB}	—	9	—	pF
COUPLED						
Output Collector Current (I_F = 10 mA, V_{CE} = 10 V)		I_C (CTR)(2)				mA (%)
	4N25,25A,26		2 (20)	7 (70)	—	
	4N27,28		1 (10)	5 (50)	—	
Collector–Emitter Saturation Voltage (I_C = 2 mA, I_F = 50 mA)		$V_{CE(sat)}$	—	0.15	0.5	Volts
Turn–On Time (I_F = 10 mA, V_{CC} = 10 V, R_L = 100 Ω)(3)		t_{on}	—	2.8	—	μs
Turn–Off Time (I_F = 10 mA, V_{CC} = 10 V, R_L = 100 Ω)(3)		t_{off}	—	4.5	—	μs
Rise Time (I_F = 10 mA, V_{CC} = 10 V, R_L = 100 Ω)(3)		t_r	—	1.2	—	μs
Fall Time (I_F = 10 mA, V_{CC} = 10 V, R_L = 100 Ω)(3)		t_f	—	1.3	—	μs
Isolation Voltage (f = 60 Hz, t = 1 sec)(4)		V_{ISO}	7500	—	—	Vac(pk)
Isolation Resistance (V = 500 V)(4)		R_{ISO}	10^{11}	—	—	Ω
Isolation Capacitance (V = 0 V, f = 1 MHz)(4)		C_{ISO}	—	0.2	—	pF

1. Always design to the specified minimum/maximum electrical limits (where applicable).
2. Current Transfer Ratio (CTR) = I_C/I_F x 100%.
3. For test circuit setup and waveforms, refer to Figure 11.
4. For this test, Pins 1 and 2 are common, and Pins 4, 5 and 6 are common.

TYPICAL CHARACTERISTICS

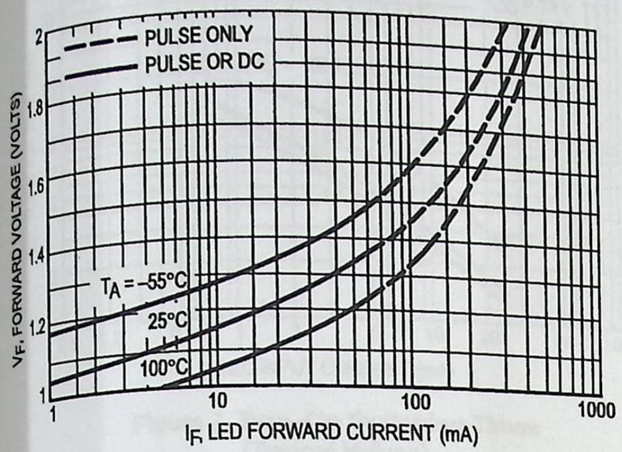


Figure 1. LED Forward Voltage versus Forward Current

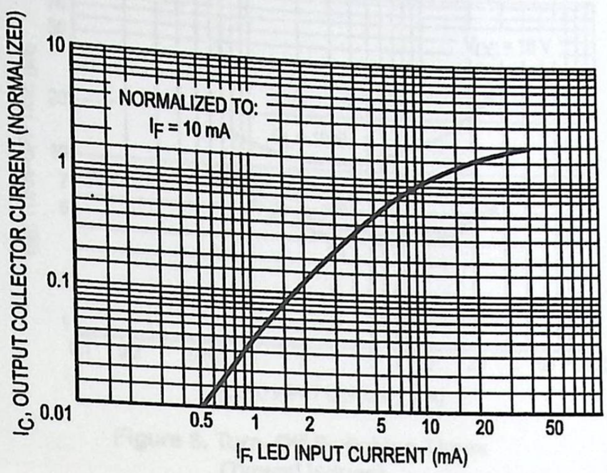


Figure 2. Output Current versus Input Current

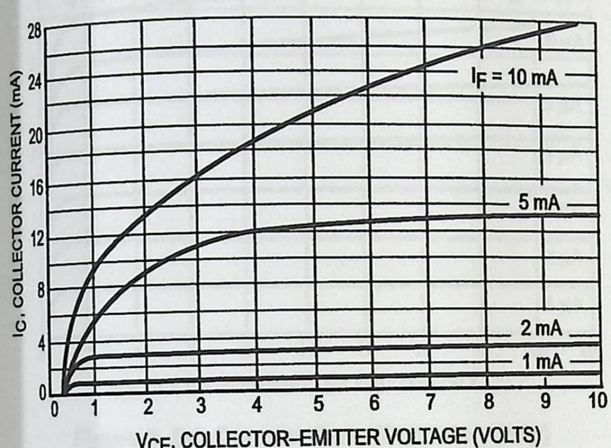


Figure 3. Collector Current versus Collector–Emitter Voltage

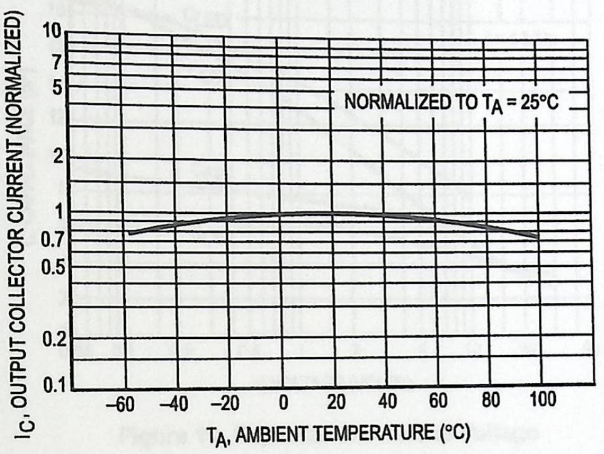


Figure 4. Output Current versus Ambient Temperature

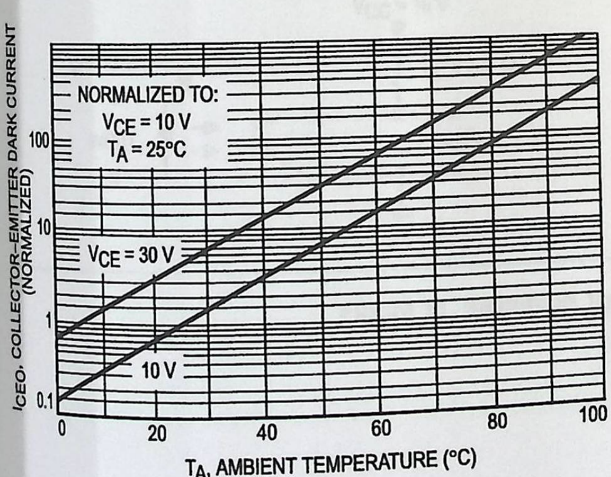


Figure 5. Dark Current versus Ambient Temperature

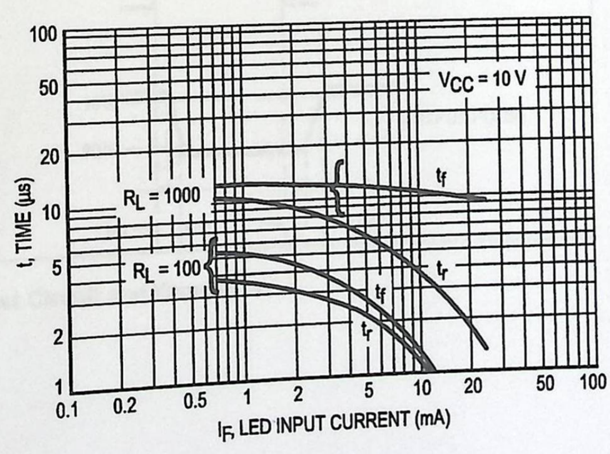


Figure 6. Rise and Fall Times (Typical Values)

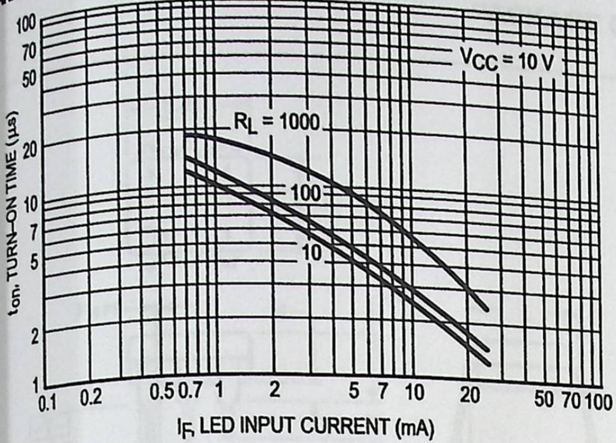


Figure 7. Turn–On Switching Times (Typical Values)

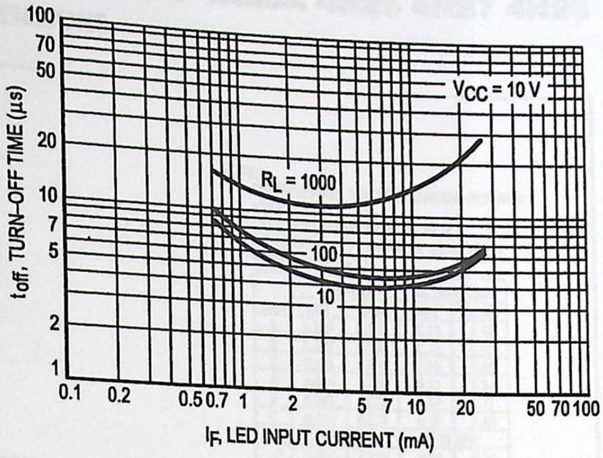


Figure 8. Turn–Off Switching Times (Typical Values)

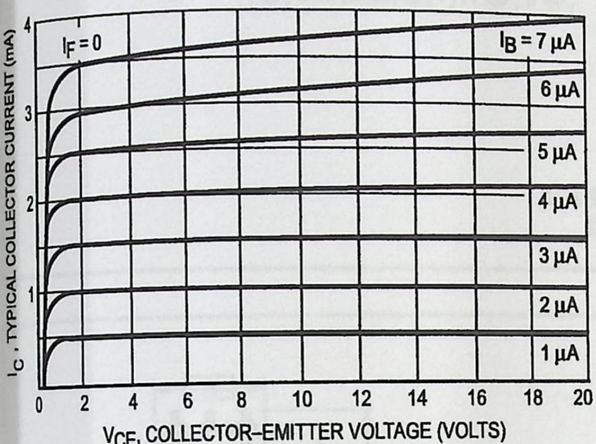


Figure 9. DC Current Gain (Detector Only)

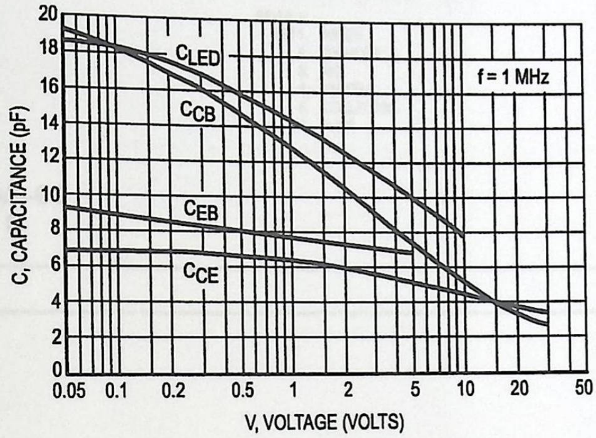


Figure 10. Capacitances versus Voltage

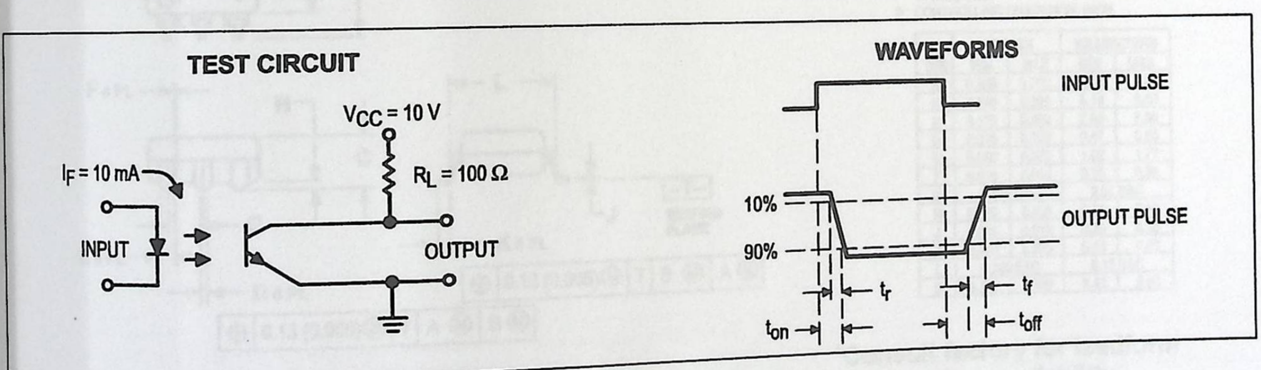
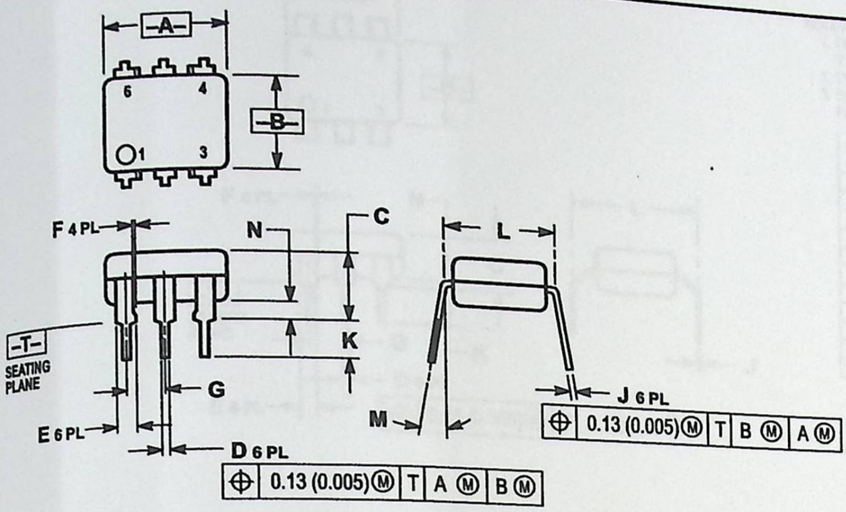


Figure 11. Switching Time Test Circuit and Waveforms

PACKAGE DIMENSIONS

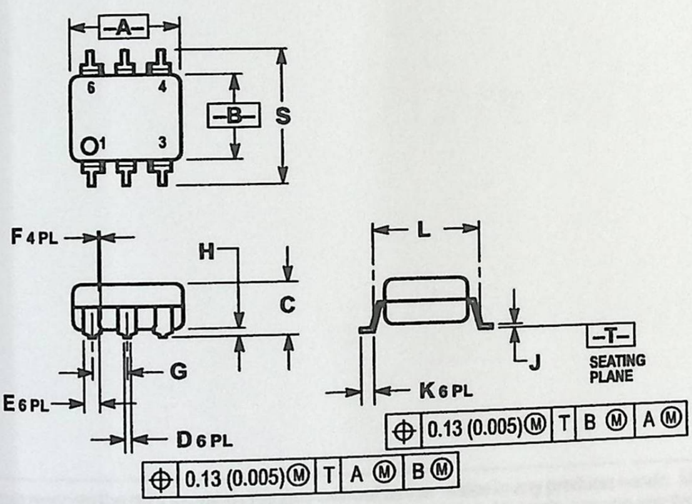


NOTES:
1. DIMENSIONING AND TOLERANCING PER ANSI Y14.5M, 1982.
2. CONTROLLING DIMENSION: INCH.
3. DIMENSION L TO CENTER OF LEAD WHEN FORMED PARALLEL.

DIM	INCHES MIN	INCHES MAX	MILLIMETERS MIN	MILLIMETERS MAX
A	0.320	0.350	8.13	8.89
B	0.240	0.260	6.10	6.60
C	0.115	0.200	2.93	5.08
D	0.016	0.020	0.41	0.50
E	0.040	0.070	1.02	1.77
F	0.010	0.014	0.25	0.36
G	0.100 BSC		2.54 BSC	
J	0.008	0.012	0.21	0.30
K	0.100	0.150	2.54	3.81
L	0.300 BSC		7.62 BSC	
M	0°	15°	0°	15°
N	0.015	0.100	0.38	2.54

STYLE 1:
PIN 1. ANODE
2. CATHODE
3. NC
4. EMITTER
5. COLLECTOR
6. BASE

CASE 730A–04
ISSUE G

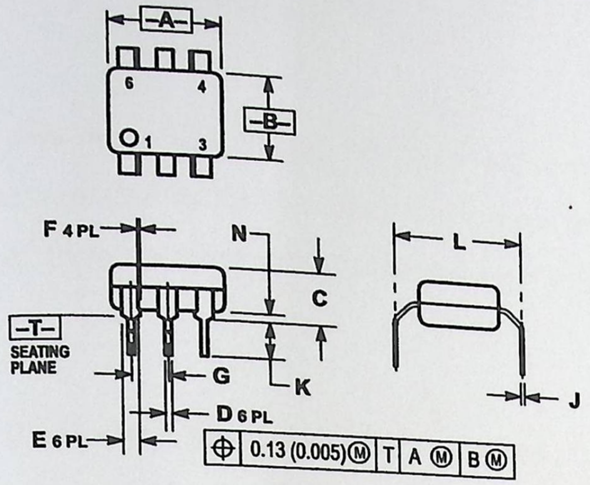


NOTES:
1. DIMENSIONING AND TOLERANCING PER ANSI Y14.5M, 1982.
2. CONTROLLING DIMENSION: INCH.

DIM	INCHES MIN	INCHES MAX	MILLIMETERS MIN	MILLIMETERS MAX
A	0.320	0.350	8.13	8.89
B	0.240	0.260	6.10	6.60
C	0.115	0.200	2.93	5.08
D	0.016	0.020	0.41	0.50
E	0.040	0.070	1.02	1.77
F	0.010	0.014	0.25	0.36
G	0.100 BSC		2.54 BSC	
H	0.020	0.025	0.51	0.63
J	0.008	0.012	0.20	0.30
K	0.006	0.035	0.16	0.88
L	0.320 BSC		8.13 BSC	
S	0.332	0.390	8.43	9.90

***Consult factory for leadform option availability**

CASE 730C–04
ISSUE D



NOTES:
1. DIMENSIONING AND TOLERANCING PER ANSI Y14.5M, 1982.
2. CONTROLLING DIMENSION: INCH.
3. DIMENSION L TO CENTER OF LEAD WHEN FORMED PARALLEL.

DIM	INCHES		MILLIMETERS	
	MIN	MAX	MIN	MAX
A	0.320	0.350	8.13	8.89
B	0.240	0.260	6.10	6.60
C	0.115	0.200	2.93	5.08
D	0.016	0.020	0.41	0.50
E	0.040	0.070	1.02	1.77
F	0.010	0.014	0.25	0.36
G	0.100 BSC		2.54 BSC	
J	0.008	0.012	0.21	0.30
K	0.100	0.150	2.54	3.81
L	0.400	0.425	10.16	10.80
N	0.015	0.040	0.38	1.02

***Consult factory for leadform option availability**

CASE 730D–05
ISSUE D

How to reach us:
USA / EUROPE: Motorola Literature Distribution;
P.O. Box 20912; Phoenix, Arizona 85036. 1–800–441–2447

MFAX: RMFAX0@email.sps.mot.com – TOUCHTONE (602) 244–6609
INTERNET: http://Design–NET.com

JAPAN: Nippon Motorola Ltd.; Tatsumi–SPD–JLDC, Toshikatsu Otsuki, 6F Seibu–Butsuryu–Center, 3–14–2 Tatsumi Koto–Ku, Tokyo 135, Japan. 03–3521–8315

HONG KONG: Motorola Semiconductors H.K. Ltd.; 8B Tai Ping Industrial Park, 51 Ting Kok Road, Tai Po, N.T., Hong Kong. 852–26629298

Appendix B

(Java algorithms Code)

Tracking Algorithm :

There many algorithm for tracking , here we want to talk about the tracking algorithm that are used by the camera, the algorithm importance is to obtain the new value of x and from the current frame that taking by the camera .

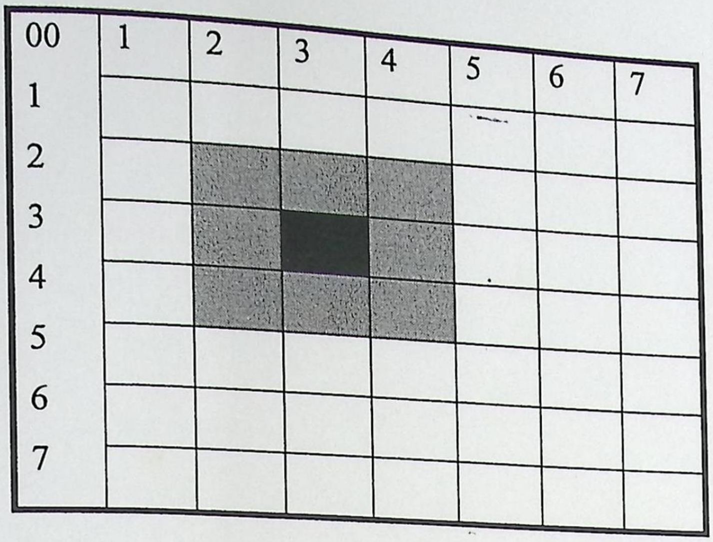
Now we well talk about how color is being tracked , the way is simple the algorithm is used called color Blob tracking .the color blob tracking algorithm allows the user to enter minimum and maximum bound for each of either the three RGB or YCrCb channel values .

Each pixel in the buffer is compared against the user specific bound . the coordinate of pixel that fall within the color bound are compared against precisely stored coordinate to generate abounding box .

As the data stream in the camera the microcontroller compare each pixel to min and max colour value as well as wither matching pixels the CMU camera and produce a confidence measure calculated by the following formula

confidence = num_matching_pixle / box_area * 256

The centre of an identified blob is also calculated by summing all the x and y coordinates of the matched pixels and diving that by the number of detected pixels .



The center of an identified blob is also calculated by summing all x and y coordinates e of the match pixel and dividing that value by the number of detected pixel .

The above illustrating representation an 8X8 bounding box with a 3*3 "blob" , here the centriod calculation would look as the follows:

Centriod_x = (2+3+4)*3=27/9=3

Centriod_y = (2+3+4)*3=27/9=3

So the centered is the illustration is located at (3,3) and is represented by dark pixel .