Introduction

This project “Image Processing Based Embedded Quality Analyzer and Controller” deals with developing an economic and safe way to analyze the fruit or vegetable quality which is based on color, and to control the speed of the grinder according to the quality of the product.

Here we are analyzing the quality of the product with the help of image processing toolbox in MATLAB After checking the quality of the apples we control the speed of the grinder according to the quality of the apple. Apple of grade A being more ripe needs less speed of the grinder for converting it into juice. Similarly more speed of the grinder is required of grade C and grade D apples.

Thus, here we have a fully automated system with the help of which farmers and the manufacturers of juice and jam making units can easily check the quality of the fruit and vegetables and also control the speed of the process according to the quality of the fruit/vegetables.

For developing the technique to check the quality of the fruit based on color matching and controlling the speed of the motor we use a help the color fundamentals in image processing. For color matching, MATLAB is used and the hardware part includes a motor controlled by a microcontroller handled with the help of C language.

The color fundamentals are used to access the color content of the Apples. As the Image Acquisition is not a part of this project so it is assumed that the image acquired through a Digital Camera is stored in the computer memory.

Then the Image Processing Fundamentals are taken into consideration. These details out the representation of the Image in space coordinate form. The types of resolution along with the bit depth are explained henceforth in the hardware description part. The software description details out the code along with the brief description of the various commands of MATLAB version 7.
Images Under Test

apple. Jpg

apple1.jpg

apple2.jpg
**Color & Image Processing Fundamentals**

An Image may be defined as a two dimensional function, \( f(x, y) \) where \( x \) and \( y \) are spatial co-ordinates, and the amplitude of \( f \) at any pair of co-ordinates \((x, y)\) is called the intensity or gray level of that image at that point. When \( x, y, \) and the amplitude of \( f \) are all finite, discrete quantities, we call the image a digital image. The field of digital image processing involves processing digital images by means of digital computer. Image processing is composed of a finite number of elements, each of which has a particular location and value. These elements are referred to as picture elements, image elements, pels and pixels. Image processing involves processing or altering an existing image in terms of pixels in a desired manner.

**Resolution of digital images**

The resolution of digital images can be described in many different ways.

**Pixel resolution:** The term resolution is often used as a pixel count in digital imaging. An image of \( N \) pixels high by \( M \) pixels wide can have any resolution less than \( N \) lines per picture height, or \( N \) TV lines. But when the pixel counts are referred to as resolution, the convention is to describe the pixel resolution with the set of two positive integer numbers, where the first number is the number of pixel columns (width) and the second is the number of pixel rows (height), for example as 640 by 480. Another popular convention is to cite resolution as the total number of pixels in the image, typically given as number of mega pixels, which can be calculated by multiplying pixel columns by pixel rows and dividing by one million. Other conventions include describing pixels per length unit or pixels per area unit, such as pixels per inch or per square inch. None of these pixel resolutions are true resolutions, but they are widely referred to as such; they serve as upper bounds on image resolution.

**Spatial resolution:** The measure of how closely lines can be resolved in an image is called spatial resolution, and it depends on properties of the system creating the image, not just the pixel resolution in pixels per inch (ppi). For practical purposes the clarity of the image is decided by its spatial resolution, not the number of pixels in an image. The spatial resolution of computer monitors is generally 72 to 100 lines per inch, corresponding to pixel resolutions of 72 to 100 ppi.
Spectral resolution: Color images distinguish light of different spectrum. Multi-spectral images resolve even finer differences of spectrum or wavelength than is needed to reproduce color. That is, they can have higher spectral resolution.

Temporal resolution: Movie cameras and high-speed cameras can resolve events at different points in time. The time resolution used for movies is usually 15 to 30 frames per second (fps), while high-speed cameras may resolve 100 to 1000 fps, or even more.

Radiometric resolution: Radiometric resolution determines how finely a system can represent or distinguish differences of intensity, and is usually expressed as a number of levels or a number of bits, for example 8 bits or 256 levels is typical of computer image files. The higher the radiometric resolution, the better subtle differences of intensity or reflectivity can be represented, at least in theory.

The Color Theory

Computer monitors emit color as RGB (red, green, and blue) light. Although all colors of the visible spectrum can be produced by merging red, green and blue light, monitors are capable of displaying only a limited gamut (i.e., range) of the visible spectrum.

Whereas monitors emit light, inked paper absorbs or reflects specific wavelengths. Cyan, magenta and yellow pigments serve as filters, subtracting varying degrees of red, green and blue from white light to produce a selective gamut of spectral colors. Like monitors, printing inks also produce a color gamut that is only a subset of the visible spectrum, although the range is not the same for both. Consequently, the same art displayed on a computer monitor may not match to that printed in a publication.
Also, because printing processes such as offset lithography use CMYK (cyan, magenta, yellow, black) inks, digital art must be converted to CMYK color for print. Many printers now prefer digital art files be supplied in the RGB color space with ICC profiles attached. Images can then be converted to the CMYK color space by the printer using color management methods that honor profiles if present; this helps preserve the best possible detail and vibrancy.
Color Image Processing

The human visual system can distinguish hundreds of thousands of different color shades and intensities, but only around 100 shades of grey. Therefore, in an image, a great deal of extra information may be contained in the color, and this extra information can then be used to simplify image analysis, e.g. object identification and extraction based on color.

Three independent quantities are used to describe any particular color. The hue is determined by the dominant wavelength. Visible colors occur between about 400nm (violet) and 700nm (red) on the electromagnetic spectrum.

The saturation is determined by the excitation purity, and depends on the amount of white light mixed with the hue. A pure hue is fully saturated, i.e. no white light mixed in. Hue and saturation together determine the chromaticity for a given color. Finally, the intensity is determined by the actual amount of light, with more light corresponding to more intense colors.

Achromatic light has no color - its only attribute is quantity or intensity. Grey level is a measure of intensity. The intensity is determined by the energy, and is therefore a physical quantity. On the other hand, brightness or luminance is determined by the perception of the color, and is therefore psychological. Color depends primarily on the reflectance properties of an object. We see those rays that are reflected, while others are absorbed. However, we also must consider the color of the light source, and the nature of human visual system.

Tristimulus Theory of Colour Perception

The human retina has 3 kinds of cones. The response of each type of cone as a function of the wavelength of the incident light is shown the peaks for each curve are at 440nm (blue), 545nm (green) and 580nm (red). Note that the last two actually peak in the yellow part of the spectrum.
Spectral response curves for each cone type.

In other words, the perception of colour is an entirely arbitrary creation of our nervous system, and is in no way contained in the wavelengths themselves.

The tristimulus theory of color perception seems to imply that any color can be obtained from a mix of the three primaries, red, green and blue, but although nearly all visible colors can be matched in this way, some cannot.

The Chromaticity Diagram
In 1931, the Commission Internationale de l’Éclairage (CIE) defined three standard primaries, called X, Y and Z, that can be added to form all visible colors. The primary Y was chosen so that its color matching function exactly matches the luminous-efficiency function for the human eye, given by the sum of the three curves in the spectral response curve. A color is then specified by its trichromatic coefficients:

\[
\begin{align*}
X &= \frac{X}{X + Y + Z} \\
Y &= \frac{Y}{X + Y + Z} \\
Z &= \frac{Z}{X + Y + Z}
\end{align*}
\]

Obviously, \(x + y + z = 1\)

The pure colors of the spectrum lie on the curved part of the boundary, and a standard white light has color defined to be near (but not at) the point of equal energy \(x = y = z = \frac{1}{3}\). Complementary colors, i.e. colors that add to give white, lie on the endpoints of a line through this point. As illustrated in the next figure, all the colors along any line in the chromaticity diagram may be obtained by mixing the colors on the end points of the line. Furthermore, all colors within a triangle may be formed by mixing the colors at the vertices. This property illustrates graphically the fact that all visible colors cannot be obtained by a mix of R, G and B (or any other three visible) primaries alone, since the diagram is not triangular!

Mixing colors on the chromaticity diagram.
All colors on the line IJ can be obtained by mixing colors I and J, and all colors in the triangle IJK can be obtained by mixing colors I, J and K.

(a) Colour Image

(b) Intensity Image

(c) Luminance Image

Image (a) shows a colour test pattern, consisting of horizontal stripes of black, blue, green, cyan, red, magenta and yellow, a colour ramp with constant intensity, maximal saturation, and hue changing linearly from red through green to blue, and a greyscale ramp from black to white.

Image (b) shows the intensity for image (a).

Image (c) shows the luminance.

This third image accurately reflects the brightness variations perceived in the original image.
Color models provide a standard way to specify a particular color, by defining a 3D coordinate system, and a subspace that contains all constructible colors within a particular model. Any color that can be specified using a model will correspond to a single point within the subspace it defines. Each color model is oriented towards either specific hardware (RGB, CMY, YIQ), or image processing applications (HSI).

**Intensity image (gray scale image)**: It represents an image as a matrix where every element has a value corresponding to how bright/dark the pixel at the corresponding position should be colored. There are two ways to represent the number that represents the brightness of the pixel: The double class (or data type). This assigns a floating number ("a number with decimals") between 0 and 1 to each pixel. The value 0 corresponds to black and the value 1 corresponds to white. The other class is called uint8 which assigns an integer between 0 and 255 to represent the brightness of a pixel. The value 0 corresponds to black and 255 to white. The class uint8 only requires roughly 1/8 of the storage compared to the class double. On the other hand, many mathematical functions can only be applied to the double class. We will see later how to convert between double and uint8.

**Binary image**: This image format also stores an image as a matrix but can only color a pixel black or white (and nothing in between). It assigns a 0 for black and a 1 for white.

**Indexed image**: This is a practical way of representing color images. An indexed image stores an image as two matrices. The first matrix has the same size as the image and one number for each pixel. The second matrix is called the color map and its size may be different from the image. The numbers in the first matrix is an instruction of what number to use in the color map matrix.

**RGB image**: This is another format for color images. It represents an image with three matrices of sizes matching the image format. Each matrix corresponds to one of the colors red, green or blue and gives an instruction of how much of each of these colors a certain pixel should use.

**Multiframe image**: To study a sequence of images. This is very common in biological and medical imaging where you might study a sequence of slices of a cell. For these cases, the multiframe format is a convenient way of working with a sequence of images.
Bit Depth

The term "bit depth" is used to describe the number of bits used to store information about each pixel of an image. The higher the depth, the more colors that are available for storage. The bit depth of an image will determine how many levels of gray (or color) can be generated. Some examples are given below:

- **Monochrome image**
  - 1-Bit image
  - 1 bit = 2 levels of gray

- **Grayscale image**
  - 8-Bit image (1 x 8-Bit channels)
  - 8 bit = 256 levels of gray

- **Four Color (CMYK) image**
  - 32-Bit image (4 x 8-Bit channels)
  - 8 bit = 256 levels of gray per color
If an image is stored as a JPEG-image on the disc we first read it into Matlab. However, in order to start working with an image, for example perform a wavelet transform on the image, we must convert it into a different format. There are various formats.

**Major graphic file formats**

There are many graphic file formats, if we include the proprietary types. The PNG, JPEG, and GIF formats are most often used to display images. These graphic formats are listed and briefly described below, separated into the two main families of graphics: raster and vector.

**JPEG:** The JPEG (Joint Photographic Experts Group) image files are a lossy format. The DOS filename extension is JPG, although other operating systems may use JPEG. Nearly all digital cameras have the option to save images in JPEG format. The JPEG format supports 8-bit per color - red, green, and blue, for 24-bit total - and produces relatively small file sizes. Fortunately, the compression in most cases does not detract noticeably from the image. But JPEG files do suffer generational degradation when repeatedly edited and saved. Photographic images are best stored in a lossless non-JPEG format if they will be re-edited in future, or if the presence of small "artifacts" (blemishes), due to the nature of the JPEG compression algorithm, is unacceptable. JPEG is also used as the image compression algorithm in many Adobe PDF files.

**TIFF:** The TIFF (Tagged Image File Format) is a flexible image format that normally saves 16-bit per color - red, green and blue for a total of 48-bits - or 8-bit per color - red, green and blue for a total of 24-bits - and uses a filename extension of TIFF or TIF. TIFF's flexibility is both a feature and a curse, with no single reader capable of handling all the different varieties of TIFF files. TIFF can be lossy or lossless. Some types of TIFF offer relatively good lossless compression for bi-level (black and white, no grey) images. Some high-end digital cameras have the option to save images in the TIFF format, using the LZW compression algorithm for lossless storage. The TIFF image format is not widely supported by web browsers, and should not be used on the World Wide Web. TIFF is still widely accepted as a photograph file standard in the
printing industry. TIFF is capable of handling device-specific color spaces, such as the CMYK defined by a particular set of printing press inks.

**RAW:** The RAW image format is a file option available on some digital cameras. It usually uses a lossless compression and produces file sizes much smaller than the TIFF format. Unfortunately, the RAW format is not standard among all camera manufacturers and some graphic programs and image editors may not accept the RAW format.

**PNG:** The PNG (Portable Network Graphics) file format is regarded and was made as the free and open-source successor to the GIF file format. The PNG file format supports true color (16 million colors) whereas the GIF file format only allows 256 colors. PNG excels when the image has large areas of uniform color. The lossless PNG format is best suited for editing pictures, and the lossy formats like JPG are best for final distribution of photographic-type images because of smaller file size. Many older browsers do not yet support the PNG file format, however with the release of Internet Explorer 7 all popular modern browsers fully support PNG.

**GIF:** GIF (Graphic Interchange Format) is limited to an 8-bit palette, or 256 colors. This makes the GIF format suitable for storing graphics with relatively few colors such as simple diagrams, shapes and cartoon style images. The GIF format supports animation and is still widely used to provide image animation effects. It also uses a lossless compression that is more effective when large areas have a single color, and ineffective for detailed images or dithered images.

**BMP:** The BMP (bit mapped) format is used internally in the Microsoft Windows operating system to handle graphics images. These files are typically not compressed resulting in large files. The main advantage of BMP files is their wide acceptance, simplicity, and use in Windows programs. However, they may pose problems for users of other operating systems. Commonly, BMP files are used for Microsoft's Paint program. Since most BMP files are uncompressed, and BMP's RLE compression has serious limits, the large size of BMP files makes them unsuitable for file transfer. Desktop backgrounds and images from scanners are usually stored in BMP files.

**WDP:** The WDP format is the newly introduced image format by Microsoft for media print quality, lossless image compression.

**XPM:** The XPM format is the default X Window System picture format (very popular in the Linux world). Its structure is based on the string format of the C programming language.
MrSID: The MrSID (Multiresolution Seamless Image Database) format is a wavelet compression format used mostly by Geographic Information Systems to store massive satellite imagery for map software.

SVG: SVG (Scalable Vector Graphics) is an open standard created and developed by the World Wide Web Consortium to address the need (and attempts of several corporations) for a versatile, scriptable and all-purpose vector format for the web and otherwise. The SVG format does not have a compression scheme of its own, but due to the textual nature of XML, an SVG graphic can be compressed using a program such as gzip. Because of its scripting potential, SVG is a key component in web applications: interactive web pages that look and act like applications.

Given all these different representations of color, and hence color images, the question arises as to what is the best way to apply the image processing techniques we have covered so far to these images? One possibility is to apply the transformations to each color plane in an RGB image. If we want to increase the contrast in a dark image by histogram equalization, can we just equalize each color independently? This will result in quite different colors in our transformed image. In general it is better to apply the transformation to just the intensity component of an HSI image, or the luminance component of a YIQ image, thus leaving the chromaticity unaltered.

Steps in Image Processing

System where the process of applying any operation to image data for a given purpose is used. Examples of operations include

- Image Acquisition
- Image enhancement,
- Image restoration,
- Image preprocessing,
- Image Compression,
- Morphological Processing,
- Segmentation
- Representation & Description
- Object Recognition
**Working**

The Project entitled “Image Processing Based Embedded Quality Analyzer and Controller” consists of the hardware and the software modules. The two modules are

- The Image Processing module
- The Motor Control module

**The Image Processing module:** This module consists of the image acquisition part through any of the modes explained in the hardware description section (this is not a part of this project and hence optional). The acquired image is stored in the PC in the JPEG format. The JPEG image is then processed for the presence of various colors, as we now know that a color image comprises of varying amounts of the primary colors.

![A Color Image](image)

The Bitonal Components of the color Image

The quantity of the colors is determined and then depending on the content of the presence of the red color the apples are categorized in four categories, namely

- Grade A (with more red content and hence more ripe)
- Grade B (with less red content and hence less ripe)
- Grade C (with lesser red content and hence lesser ripe)
- Grade D. (with least red content and hence least ripe)
This whole is done using MATLAB version 7 (a high-performance language for technical computing.). The Image Processing Toolbox is used.

**The Motor Control module:** The ATMEL 89C51 is used to control the speed of the motor according to the category. A ‘C’ language program is used. The motor has different speeds for the different categories.

- Apples of grade A needs less speed of the grinder for converting into juice.
- Apples of grade B needs more speed of the grinder.
- Apples of grade C needs more speed of the grinder as compared to grade B.
- Apples of grade D needs more speed of the grinder as compared to grade C.

Thus the grinder speed is controlled through this motor.

The PC is interfaced with the hardware module via DB 9 and using MAX 232 communication interface.

An LCD is attached which shows the results of categorizing the apples. Hence a supervisor sitting in his cabin can monitor the whole process.
Hardware Description

Basic Components used for designing are:

- Digital Camera
- Microcontroller
- LCD
- MAX 232
- Power Supply Source
- DB-9 Connector
- Optocoupler
- Electromagnetic Relay

Digital Camera

A digital camera is an electronic device used to capture and store photographs electronically instead of using photographic film like conventional cameras. Modern compact digital cameras are typically multifunctional, with some devices capable of recording sound and/or video as well as photographs. Professional digital cameras are generally dedicated to photography.

The resolution of a digital camera is often limited by the camera sensor (usually a charge-coupled device or CCD chip) that turns light into discrete signals, replacing the job of film in traditional photography. The sensor is made up of millions of "buckets" that collect charge in response to light. Generally, these buckets respond to only a narrow range of light wavelengths, due to a color filter over each. Each one of these buckets is called a pixel, and interpolation algorithm is needed to turn the image with only one wavelength range per pixel into an RGB image where each pixel is three numbers to represent a complete color.

The one attribute most commonly compared on cameras is the pixel count. The pixel count alone is commonly presumed to indicate the resolution of a camera, but this is a misconception. There are several other factors that impact a sensor's resolution. Some of these factors include sensor size, lens quality, and the organization of the pixels (for example, a monochrome camera without a Bayer filter mosaic has a higher resolution than a typical color camera). Many digital compact cameras are criticized for having too many pixels, in that the
sensors can be so small that the resolution of the sensor is greater than the lens could possibly deliver. Excessive pixels can even lead to a decrease in image quality. As each pixel sensor gets smaller it is catching fewer photons, and so the signal-to-noise ratio will decrease. This decrease leads to noisy pictures, poor shadow region quality and generally poorer-quality pictures.

**Methods of image capture**

Since the first digital backs were introduced, there have been three main methods of capturing the image, each based on the hardware configuration of the sensor and color filters.

The first method is often called single-shot, in reference to the number of times the camera’s sensor is exposed to the light passing through the camera lens. The second method is referred to as multi-shot because the sensor is exposed to the image in a sequence of three or more openings of the lens aperture. The most common originally was to use a single image sensor with three filters (once again red, green and blue) passed in front of the sensor in sequence to obtain the additive color information. A third version combined the two methods without a Bayer filter on the chip. The third method is called scanning because the sensor moves across the focal plane much like the sensor of a desktop scanner. Their linear or tri-linear sensors utilize only a single line of photo sensors, or three lines for the three colors. In some cases, scanning is accomplished by rotating the whole camera; a digital rotating line camera offers images of very high total resolution.

The choice of method for a given capture is of course determined largely by the subject matter. It is usually inappropriate to attempt to capture a subject that moves with anything but a single-shot system. However, the higher color fidelity and larger file sizes and resolutions available with multi-shot and scanning backs make them attractive for commercial photographers working with stationary subjects.

Many digital cameras can connect directly to a computer to transfer data:

- Early cameras used the PC serial port. USB is now the most widely used method (Most cameras are viewable as USB Mass Storage), though some have a FireWire port. Some cameras use USB PTP mode for connection instead of USB MSC; some offer both modes.

- Other cameras use wireless connections, via Bluetooth or IEEE 802.11 Wi-Fi.

Digital cameras need memory to store data. A wide variety of storage media has been used.
Microcontrollers for Embedded Systems

In the literature discussing microprocessors, we often see a term embedded system. Microprocessors and microcontrollers are widely used in embedded system products. An embedded product uses a microprocessor (or microcontroller) to do one task and one task only. A printer is an example of embedded system since the processor inside it performs one task only: namely, get data and print it. Contrasting this with a IBM PC which can be used for a number of applications such as word processor, print server, network server, video game player, or internet terminal. Software for a variety of applications can be loaded and run. Of course the reason a PC can perform myriad tasks is that it has RAM memory and an operating system that loads the application software into RAM and lets the CPU run it. In an embedded system, there is only one application software that is burned into ROM. An PC contains or is connected to various embedded products such as the keyboard, printer, modem, disk controller, sound card, CD-ROM driver, mouse and so on. Each one of these peripherals has a microcontroller inside it that performs only one task. For example, inside every mouse there is a microcontroller to perform the task of finding the mouse position and sending it to the PC.

Although microcontrollers are the preferred choice for many embedded systems, there are times that a microcontroller is inadequate for the task. For this reason, in many years the manufacturers for general-purpose microprocessors have targeted their microprocessor for the high end of the embedded market.

There are four major 8-bit microcontrollers. They are Motorola’s 6811, Intel’s 8051, Zilog’s Z8, and PIC 16 X from microchip technology. Each of the above microcontrollers has a unique instruction set and register set therefore they are not compatible with each other. Programs written for one will not run on the others. There also exist 16-bit and 32- bit microcontrollers, manufactured by different chip manufacturers. With all these different microcontrollers, what criteria do designers consider in choosing one.

Three main points are as follows:

- Meeting the computing needs of the tasks at hand efficiently and cost effectively.
- Availability of software development tools such as compilers, assemblers and debuggers.
- Wide availability and reliable sources of microcontrollers.
ATMEL 89C51

It is a low-power, high-performance CMOS 8-bit microcomputer with 4K bytes of Flash programmable and erasable read only memory (PEROM). The device is manufactured using ATMEL’S high-density nonvolatile memory technology and is compatible with the industry-standard MCS-51 instruction set and pin out. The on-chip Flash allows the program memory to be reprogrammed in system or by a conventional nonvolatile memory programmer. By combining a versatile 8-bit CPU with Flash on a monolithic chip, the ATMEL AT89C51 is a powerful microcomputer, which provides a highly flexible and cost-effective solution to many embedded control applications.

The 89c51 architecture consists of these specific features:

The registers and memory locations can be made to operate using the software instructions that are incorporated as a part of the design. The program instructions have to do with the control of registers and digital data paths that are physically contained inside the 8051, as well as memory locations that are physically located outside the 8051. The model is complicated by the presence of special function registers (SFR) that must be present to make the microcomputer a microcontroller. The diagram of the 8051 packaged in a 40-pin DIP is shown in figure 2.11 with the full-abbreviated names for the signals of each pin. Many of the Pins are used for more than one function. The schematic for the 89C51 is shown in figure 2.11. ATMEL
89C51 microcontroller has 4k flash memory that is easily reprogrammable and reprogrammable by using a simple programmer.

8051 Oscillators and Clock Circuit:

It is the heart of the microcontroller that generates the clock pulses by which all the internal operations are synchronized. Pins XTAL1 and XTAL2 are provided for the connection of a resonant network to form an oscillator. Typically, a quartz crystal and capacitor are employed and thus crystal frequency will become the clock frequency of microcontroller. The 8051 maximum and minimum frequency range between 1 megahertz to 16 megahertz and 20MHz in case of AT89C51. Thus, the 8051 operate based on an external crystal. This is an electrical device which, when energy is applied, emits pulses at a fixed frequency. One can find crystals of virtually any frequency depending on the application requirements. When using on 8051, the most common crystal frequencies are 12 megahertz and 11.0592 megahertz-with
11.0592 being much more common. Why would anyone pick such an oddball frequency? There’s a real reason for it-it has to do with generating baud rates in the Serial Communication.

A cycle is, in reality, 12 pulses of the crystal. That is to say, if an instruction takes one machine cycle to execute, it will take 12 pulses of the crystal to execute. Since the crystal is pulsing 11,059,000 times per second and that one machine cycle is 12 pulses, we can calculate how many instruction cycles the 8051 can execute per second:

\[
\frac{11,059,000}{12} = 921,583
\]

This means that the 8051 can execute 921,583 single-cycle instructions per second. Since a large number of 8051 instructions are single-cycle instructions it is often considered that the 8051 can execute roughly 1 million instructions per second, although in reality it is less and depending on the instructions being used, an estimate of about 600,000 instructions per second is more realistic.

The 8051 equipped with two timers, both of which may be controlled, set, read, and configured individually.

- Keeping time and/or calculating the amount of time between events
- Counting the events themselves
- Generating baud rates for the serial port

**Input/Output Pins and Ports:**

One of the major features of a microcontroller is the versatility built into the input/output circuits that connect the 8051 to the outside world. The microprocessor designs add the additional chips to the interface with the external circuitry, but this ability is built into the microcontroller. Given this pin flexibility the 8051 may be applied simply as a single component with the I/O only or it may be expanded to include the additional memory, parallel ports and serial data communication by using the alternate pin assignments.

**P0 (Port 0, Bit-Addressable)**

This is input/output port 0. Each bit of this SFR corresponds to one of the pins on the microcontroller. For example, bit 0 of port 0 is pin P0. 0, bit 7 is pin P0. 7. Writing a value of 1 to a bit of this SFR will send a high level on the corresponding I/O pin whereas a value of 0 will
bring it to a low level. Port 0 pins may serve as inputs, outputs, or when used together as a bi-
directional low order address and data bus for the external memory.

**P1 (Port 1, Bit-Addressable)**

Port 1 pins have no dual functions. Therefore, it can just be either used as input or output.

**P2 (Port 2, Bit-Addressable)**

This is input/output port 2. Each bi of this SFR corresponds to one of the pins on the microcontroller. For example: bit 0 of port 2 is pin P2.0, bit 7 is pin P2.7. Writing a value of 1 to a bit of this SFR will send a high level on the corresponding I/O pin whereas a value of 0 will bring it to a low level. It has functions similar to the port 1. The alternate use of port 2 is to supply a high order address byte in conjunction with the port 0 low order byte to address external memory. Port 2 pins are momentarily changed by the address control signals when supplying the high order byte of 16-bit address.

**P3 (Port 3, Bit-Addressable)**

This is input/output port 3. Each bit of this SFR corresponds to one of the pins on the microcontroller. For example: bit 0 of port 3 is pin P3.0, bit 7 is pin P3.7. Writing a value of 1 to a bit of this SFR will send a high level on the corresponding I/O pin whereas a value of 0 will bring it to a low level. Thus the input/output functions can be programmed under the control of various other special function registers.

**Memory Space Allocation**

1. **Internal ROM**

   The 89C51 has a 4K bytes of on-chip ROM. This 4K bytes ROM memory has memory addresses of 0000 to 0FFFh. Program addresses higher than 0FFFFh, which exceed the internal ROM capacity will cause the microcontroller to automatically fetch code bytes from external memory. Code bytes can also be fetched exclusively from an external memory, addresses 0000h to FFFFh, by connecting the external access pin to ground. The program counter doesn’t care where the code is: the circuit designer decides whether the code is found totally in internal ROM, totally in external ROM or in a combination of internal and external ROM.
2. **Internal RAM**

   The 1289 bytes of RAM inside the 8051 are assigned addresses 00 to 7Fh. These 128 bytes can be divided into three different groups as follows:

   1. A total of 32 bytes from locations 00 to 1Fh are set aside for register banks and the stack.
   2. A total of 16 bytes from locations 20h to 2Fh are set aside for bit addressable read/write memory and instructions.
   3. A total of 80 bytes from locations 30h to 7Fh are used for read and write storage, or what is normally called a scratch pad. These 80 locations of RAM are widely used for the purpose of storing data and parameters by 8051 programmers.
The Electromagnetic Relay

The electromagnetic relay consists of a multi-turn coil, wound on an iron core, to form an electromagnet. When the coil is energized, by passing current through it, the core becomes temporarily magnetized. The magnetized core attracts the iron armature. The armature is pivoted which causes it to operate one or more sets of contacts.

When the coil is de-energized the armature and contacts are released. The coil can be energized from a low power source such as a transistor while the contacts can switch high powers such as the mains supply. The relay can also be situated remotely from the control source. Relays can generate a very high voltage across the coil when switched off. This can damage other components in the circuit. To prevent this a diode is connected across the diode.
Optocouplers

There are many situations where signals and data need to be transferred from one subsystem to another within a piece of electronics equipment, or from one piece of equipment to another, without making a direct electrical connection. Often this is because the source and destination are at very different voltage levels i.e. microcontroller, which is operating from 5V DC but being used to control a relay, which is switching 220V AC. In such situations the link between the two must be an isolated one, to protect the microcontroller from over voltage damage. Relays can of course provide this kind of isolation, but even small relays tend to be fairly bulky compared with ICs and many of today’s other miniature circuit components.

Optocouplers use a beam of light to transmit the signals or data across an electrical barrier, and achieve excellent isolation. Optocouplers typically come in a small 6-pin package, but are essentially a combination of two distinct devices: an optical transmitter, typically a gallium arsenide LED (light-emitting diode) and an optical receiver such as a phototransistor. The two are separated by a transparent barrier which blocks any electrical current flow between the two, but does allow the passage of light. The basic idea is shown in Fig.2.16, along with the usual circuit symbol for an optocoupler. Usually the electrical connections to the LED section are brought out to the pins on one side of the package and those for the phototransistor to the other side, to physically separate them as much as possible. This usually allows optocouplers to withstand voltages of anywhere between 500V and 7500V between input and output. Optocouplers are essentially digital or switching devices, so they are best for transferring either on-off control signals or digital data.

![Optocoupler Diagram](image)

Construction of typical optocoupler and the usual circuit symbol
Display

For the display of the values LMB162A Liquid Crystal Display is used. The operating principles & methods of LCD are as given below:

Liquid Crystal Display

Liquid crystal displays (LCD) are widely used in recent years as compares to LEDs. This is due to the declining prices of LCD, the ability to display numbers, characters and graphics, incorporation of a refreshing controller into the LCD, their by relieving the CPU of the task of refreshing the LCD and also the ease of programming for characters and graphics. HD 44780 based LCDs are most commonly used.

LCD pin description:

The LCD discussed in this section has the most common connector used for the based LCD is 14 pins in a row and modes of operation and how to program and interface with microcontroller is describes in this section.

V\text{CC}, V\text{SS}, V\text{EE}:

The voltage $V_{\text{CC}}$ and $V_{\text{SS}}$ provided by +5V and ground respectively while $V_{\text{EE}}$ is used for controlling LCD contrast. Variable voltage between Ground and Vcc is used to specify the contrast (or "darkness") of the characters on the LCD screen.
RS (register select):

There are two important registers inside the LCD. The RS pin is used for their selection as follows. If RS=0, the instruction command code register is selected, then allowing to user to send a command such as clear display, cursor at home etc.. If RS=1, the data register is selected, allowing the user to send data to be displayed on the LCD.

R/W (read/write):

The R/W (read/write) input allowing the user to write information from it. R/W=1, when it read and R/W=0, when it writing.

EN (enable):

The enable pin is used by the LCD to latch information presented to its data pins. When data is supplied to data pins, a high power, a high-to-low pulse must be applied to this pin in order to allow the LCD to latch in the data presented at the data pins.

D0-D7 (data lines):

The 8-bit data pins, D0-D7, are used to send information to the LCD or read the contents of the LCD’s internal registers. To display the letters and numbers, we send ASCII codes for the letters A-Z, a-z, and numbers 0-9 to these pins while making RS =1. There are also command codes that can be sent to clear the display or force the cursor to the home position or blink the cursor.

We also use RS =0 to check the busy flag bit to see if the LCD is ready to receive the information. The busy flag is D7 and can be read when R/W =1 and RS =0, as follows: if R/W =1 and RS =0, when D7 =1(busy flag =1), the LCD is busy taking care of internal operations and will not accept any information. When D7 =0, the LCD is ready to receive new information.
Interfacing of Microcontroller with LCD display

In most applications, the "R/W" line is grounded. This simplifies the application because when data is read back, the microcontroller I/O pins have to be alternated between input and output modes.

In this case, "R/W" to ground and just wait the maximum amount of time for each instruction (4.1 msecs for clearing the display or moving the cursor/display to the "home position", 160 µsecs for all other commands) and also the application software is simpler, it also frees up a microcontroller pin for other uses. Different LCD execute instructions at different rates and to avoid problems later on (such as if the LCD is changed to a slower unit). Before sending commands or data to the LCD module, the Module must be initialized. Once the initialization is complete, the LCD can be written to with data or instructions as required. Each character to display is written like the control bytes, except that the "RS" line is set. During initialization, by setting the "S/C" bit during the "Move Cursor/Shift Display" command, after each character is sent to the LCD, the cursor built into the LCD will increment to the next position (either right or left). Normally, the "S/C" bit is set (equal to "1")
Data Communication Concepts

Within a microcomputer data is transferred in parallel, because that is the fastest way to do it. For transferring data over long distances, however, parallel data transmission requires too many wires. Therefore, data to be sent long distances is usually converted from parallel form to serial form so that it can be sent on a single wire or pair of wires. Serial data received from a distant source is converted to parallel form so that it can be easily transferred on the microcomputer buses.

Serial Interface

Basic concepts concerning the serial communication can be classified into categories below:
- Interfacing requirements
- Transmission format
- Error check in data communication
- Standards in serial I/O

Interfacing Requirements

The serial interface requirement is very much similar to parallel interface requirement. Computer identifies the peripheral through port address and enable if using the read and write signals. The primary difference between the parallel I/O and serial I/O is the number of lines used for data transfer. Parallel I/O requires the entire bus while the serial I/O requires only one or pair of data lines for communication.

- Transmission Format
  - Transmission format for communication is concerned with the issues such as synchronization, direction of data flow, speed, errors and medium of transmission. Serial data can be sent synchronously or asynchronously.

- Synchronous Data Transmission
  - For synchronous data transmission data is sent in blocks at a constant rate. The start and end of the block are identified with specific bytes or bit patterns.

- Asynchronous Data Transmission
  - For asynchronous transmission each data character has a bit which identifies its start and 1 or 2 bits, which identifies its end. Since each character is individually...
identified, characters can be sent at any time (asynchronously), in the same way that a person types on a keyboard.

The asynchronous format is character oriented. Each character carries the information of the start and stop bits. When no data is being transferred, a receiver stage high at the logic 1 called mark; logic 0 is called space. The transmission of data begins with one start bit (low) followed by a character and one or two stop bits (high). This is known as framing. The asynchronous format is generally used in low speed transmission (less than 20k bits/sec) in serial I/O one bit is sent out at a time. Therefore how long the bit stays on or off is determined by the speed at which bits are transmitted. The receiver should be set up to receive the bits at the same rate of transmission; otherwise the receiver may not be able to differentiate between the two consecutive 0s and 1s.

The rate at which the bits are transmitted (bits/sec) is called baud. Each equipment has its own baud requirements. The figure shown below shows how the ASCII character A (41) will be transmitted with the 1200 baud with the framing information of one start and one stop bit. The bit time (delay between any two successive bits) is 0.83ms; this is determined by the baud as follows.

Mark

Start Bit

Transmission Data

Stop Bit

Asynchronous Data Format

- Error Check In Data Communication

During transmission, various types of errors can occur. These errors need to be checked, therefore, additional information for error checking is sent during transmission the receiver can check the received data against the error check information, and if the error is detected, the receiver can request there retransmission of that data segment. Three methods generally used for this purpose are parity check, checksum and redundancy check.
Standard in Serial I/O

The serial I/O technique is commonly used to interface terminals, printers etc. A standard is normally defined by a professional organization (such as IEEE). A standard may include such items as assignment of pin positions for signals, voltage levels, speed of data transfer, length of cable and mechanical specifications. When data are transmitted as voltage, the commonly used standard is known as RS232C. It is defined as reference to data terminal equipment (DTE) and data communication equipment (DCE). The rate of transmission is RS232C is restricted to a maximum of 20k baud and a distance of 50 feet.

Serial Port Description

The electrical specifications of the serial port are contained in the EIA (Electronics industry Association) RS232 standard.

It states many parameters such as

1. A “Space” {logic 0} will between +3 and +25 Volts.
2. A “Mark” {logic 1} will between -3 and -25 Volts.
3. The region between +3 and -3 volts is undefined.
4. An open circuit voltage should never exceed 25 volts. {In Reference to GND}
5. A short circuit current should not exceed 500 ma. The driver should be able to handle this without damage.

These connectors come in two forms: A male and a female connector. There are the D-Type 9 pin connector and D-Type 25 pin connector both of which are male on the back of the PC. The female connector has holes that allow the pins on the male end to be inserted into the connector.
**DB-9**

This is a female DB-9 connector:

![DB-9 Female Connector](image1)

The female DB-9 connector is typically used as the "plug" that goes into a typical PC. It is likely not to be used for serial communication, but rather for things like early VGA or CGA monitors (not SVGA) or for some special control/ joystick equipment.

And this is a male DB-9 connector:

![DB-9 Male Connector](image2)

This is the connector that is used for serial communications on a "generic" PC. Often you will see two of them side by side (for COM1 and COM2). Special equipment that you might communicate with would have either connector, or even one of the DB-25 connectors listed below.
Pin connections for 9 pin connectors

<table>
<thead>
<tr>
<th>Pin Number</th>
<th>Direction of signal:</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Carrier Detect (CD) (from DCE) Incoming signal from a modem</td>
</tr>
<tr>
<td>2</td>
<td>Received Data (RD) Incoming Data from a DCE</td>
</tr>
<tr>
<td>3</td>
<td>Transmitted Data (TD) Outgoing Data to a DCE</td>
</tr>
<tr>
<td>4</td>
<td>Data Terminal Ready (DTR) Outgoing handshaking signal</td>
</tr>
<tr>
<td>5</td>
<td>Signal Ground Common reference voltage</td>
</tr>
<tr>
<td>6</td>
<td>Data Set Ready (DSR) Incoming handshaking signal</td>
</tr>
<tr>
<td>7</td>
<td>Request To Send (RTS) Outgoing flow control signal</td>
</tr>
<tr>
<td>8</td>
<td>Clear To Send (CTS) Incoming flow control signal</td>
</tr>
<tr>
<td>9</td>
<td>Ring Indicator (RI) (from DCE) Incoming signal from a modem</td>
</tr>
</tbody>
</table>
The TD (transmit data) wire is the one through which data from a DTE device is transmitted to a DCE device. This name can be deceiving, because this wire is used by a DCE device to receive its data. The TD line is kept in a mark condition by the DTE device when it is idle. The RD (receive data) wire is the one on which data is received by a DTE device, and the DCE device keeps this line in a mark condition when idle.

RTS stands for Request To Send. This line and the CTS line are used when "hardware flow control" is enabled in both the DTE and DCE devices. The DTE device puts this line in a

---

### Serial pinouts [D9 Connectors]

<table>
<thead>
<tr>
<th>Pin</th>
<th>Pin Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>DCD (Data Carrier Detect)</td>
</tr>
<tr>
<td>2</td>
<td>RX (Receive Data)</td>
</tr>
<tr>
<td>3</td>
<td>TX (Transmit Data)</td>
</tr>
<tr>
<td>4</td>
<td>DTR (Data Terminal Ready)</td>
</tr>
<tr>
<td>5</td>
<td>GND (Signal Ground)</td>
</tr>
<tr>
<td>6</td>
<td>DSR (Data Set Ready)</td>
</tr>
<tr>
<td>7</td>
<td>RTS (Request To Send)</td>
</tr>
<tr>
<td>8</td>
<td>CTS (Clear To Send)</td>
</tr>
<tr>
<td>9</td>
<td>RI (Ring Indicator)</td>
</tr>
</tbody>
</table>
mark condition to tell the remote device that it is ready and able to receive data. If the DTE device is not able to receive data (typically because its receive buffer is almost full), it will put this line in the space condition as a signal to the DCE to stop sending data. When the DTE device is ready to receive more data (i.e. after data has been removed from its receive buffer), it will place this line back in the mark condition. The complement of the RTS wire is CTS, which stands for Clear To Send. The DCE device puts this line in a mark condition to tell the DTE device that it is ready to receive the data. Likewise, if the DCE device is unable to receive data, it will place this line in the space condition. Together, these two lines make up what is called RTS/CTS or "hardware" flow control. The Software Wedge supports this type of flow control, as well as X on/X off or "software" flow control. Software flow control uses special control characters transmitted from one device to another to tell the other device to stop or start sending data. With software flow control the RTS and CTS lines are not used.

DTR stands for Data Terminal Ready. Its intended function is very similar to the RTS line. DSR (Data Set Ready) is the companion to DTR in the same way that CTS is to RTS. Some serial devices use DTR and DSR as signals to simply confirm that a device is connected and is turned on. The Software Wedge sets DTR to the mark state when the serial port is opened and leaves it in that state until the port is closed. The DTR and DSR lines were originally designed to provide an alternate method of hardware handshaking. It would be pointless to use both RTS/CTS and DTR/DSR for flow control signals at the same time. Because of this, DTR and DSR are rarely used for flow control.

CD stands for Carrier Detect. Carrier Detect is used by a modem to signal that it has made a connection with another modem, or has detected a carrier tone.

The last remaining line is RI or Ring Indicator. A modem toggles the state of this line when an incoming call rings your phone.

The Carrier Detect (CD) and the Ring Indicator (RI) lines are only available in connections to a modem. Because most modems transmit status information to a PC when either a carrier signal is detected (i.e. when a connection is made to another modem) or when the line is ringing these two lines are rarely used.
Port Addresses and IRQ’s in the PC are as given below

<table>
<thead>
<tr>
<th>Name</th>
<th>Addresses</th>
<th>IRQ</th>
</tr>
</thead>
<tbody>
<tr>
<td>COM 1</td>
<td>3F8</td>
<td>4</td>
</tr>
<tr>
<td>COM 2</td>
<td>2F8</td>
<td>3</td>
</tr>
<tr>
<td>COM 3</td>
<td>3E8</td>
<td>4</td>
</tr>
<tr>
<td>COM 4</td>
<td>2E8</td>
<td>3</td>
</tr>
</tbody>
</table>

RS-232 communication is asynchronous. That is a clock signal is not sent with the data. Each word is synchronized using its start bit, and an internal clock on each side, keeps tabs on the timing.
MAX 232 (communication interface)

This chip is used when interfacing micro controller with PC to check the Baud rate and changes the voltage level because micro controller is TTL compatible whereas PC is CMOS compatible. The MAX 232 IC contains the necessary drivers{two} and receivers {two}, to adapt the RS- 232 signal voltage levels to TTL logic. It became popular, because it just needs one voltage{+5V} and generates the necessary RS-232 voltage levels{approx -10V AND +10V} internally. This greatly simplified the design of circuitry. And this made the IC so popular.MAX232 is just a driver/receiver. It does not generate the necessary RS-232 sequence of marks and spaces with the right timing, it does not decode RS-232 signal, it does not provide a serial /parallel conversion. All it does is to convert signal voltage levels.

General Description

The MAX220–MAX249 family of line drivers/receivers is intended for all EIA/TIA-232E and V.28/V.24 communications interfaces, particularly applications where ±12V is not available. These parts are especially useful in battery-powered systems, since their low-power shutdown mode reduces power dissipation to less than 5µW.

Applications

Portable Computers
Low-Power Modems
Interface Translation
Battery-Powered RS-232 Systems
Multidrop RS-232 Networks

Features

Superior to bipolar
Low-power receive mode in shutdown
Meet all EIA/TIA-232E and v.28 specifications.
3-state driver and receiver output.

The MAX220–MAX249 contain four sections: dual charge-pump DC-DC voltage converters, RS-232 drivers, RS-232 receivers, and receiver and transmitter enable control inputs.

**Dual Charge-Pump Voltage Converter**

The MAX220–MAX249 have two internal charge-pumps that convert +5V to ±10V (unloaded) for RS-232 driver operation. The first converter uses capacitor C1 to double the +5V input to +10V on C3 at the V+ output. The second converter uses capacitor C2 to invert +10V to -10V on C4 at the V- output. A small amount of power may be drawn from the +10V (V+) and -10V (V-) outputs to power external circuitry except on the MAX225 and MAX245–MAX247, where these pins are not available. V+ and V- are not regulated, so the output voltage drops with increasing load current. Do not load V+ and V- to a point that violates the minimum ±5V EIA/TIA-232E driver output voltage when sourcing current from V+ and V- to external circuitry. When using the shutdown feature in the MAX222, MAX225, MAX230, MAX235, MAX236, MAX240, MAX241, and MAX245–MAX249, avoid using V+ and V to power external circuitry. When these parts are shut down, V- falls to 0V, and V+ falls to +5V. For applications where a +10V external supply is applied to the V+ pin (instead of using the internal charge pump to generate +10V), the C1 capacitor must not be installed and the SHDN pin must be tied to VCC. This is because V+ is internally connected to VCC in shutdown mode.

**RS-232 Drivers**

The typical driver output voltage swing is ±8V when loaded with a nominal 5kΩ RS-232 receiver and VCC = +5V. Output swing is guaranteed to meet the EIA/TIA-232E and V.28
specification, which calls for ±5V minimum driver output levels under worst-case conditions. These include a minimum 3kΩ load, VCC = +4.5V, and maximum operating temperature. Unloaded driver output voltage ranges from (V+ -1.3V) to (V- +0.5V). Input thresholds are both TTL and CMOS compatible. The inputs of unused drivers can be left unconnected since 400kΩ input pull-up resistors to VCC are built in (except for the MAX220). The pull-up resistors force the outputs of unused drivers low because all drivers invert. The internal input pull-up resistors typically source 12µA, except in shutdown mode where the pull-ups are disabled. Driver outputs turn off and enter a high-impedance state—where leakage current is typically microamperes (maximum 25µA)—when in shutdown mode, or when in three-state mode, device power is removed. Outputs can be driven to ±15V. The power supply current typically drops to 8µA in shutdown mode.

The MAX220 does not have pull-up resistors to force the outputs of the unused drivers low. Connect unused inputs to GND or VCC. The MAX239 has a receiver three-state control line, and the MAX223, MAX225, MAX235, MAX236, MAX240, and MAX241 have both a receiver three-state control line and a low-power shutdown control. The receiver TTL/CMOS outputs are in a high-impedance, three-state mode whenever the three-state enable line is high (for the MAX225/MAX235/MAX236/MAX239– MAX241), and are also high-impedance whenever the shutdown control line is high. When in low-power shutdown mode, the driver outputs are turned off and their leakage current is less than 1µA with the driver output pulled to ground. The driver output leakage remains less than 1µA, even if the transmitter output is back driven between 0V and (VCC + 6V). Below -0.5V, the transmitter is diode clamped to ground with 1kΩ series impedance. The transmitter is also Zener clamped to approximately VCC + 6V, with a series impedance of 1kΩ. The driver output slew rate is limited to less than 30V/µs as required by the EIA/TIA-232E and V.28 specifications. Typical slew rates are 24V/µs unloaded and 10V/µs loaded with 3Ω and 2500pF.

RS-232 Receivers

EIA/TIA-232E and V.28 specifications define a voltage level greater than 3V as logic 0, so all receivers invert. Input thresholds are set at 0.8V and 2.4V, so receivers respond to TTL level inputs as well as EIA/TIA-232E and V.28 levels. The receiver inputs withstand an input over voltage up to ±25V and provide input terminating resistors with nominal 5kΩ values. The receivers implement Type 1 interpretation of the fault conditions of V.28 and EIA/TIA-232E.
The receiver input hysteresis is typically 0.5V with a guaranteed minimum of 0.2V. This produces clear output transitions with slow-moving input signals, even with moderate amounts of noise and ringing. The receiver propagation delay is typically 600ns and is independent of input swing direction.
Power Supply

The power supply circuit comprises of four basic parts:

The transformer steps down the 220 V a/c. into 12 V a/c. The transformer work on the principle of magnetic induction, where two coils: primary and secondary are wound around an iron core. The two coils are physically insulated from each other in such a way that passing an a/c. current through the primary coil creates a changing voltage in the primary coil and a changing magnetic field in the core. This in turn induces a varying a/c. voltage in the secondary coil.

The a/c. voltage is then fed to the bridge rectifier. The rectifier circuit is used in most electronic power supplies is the single-phase bridge rectifier with capacitor filtering, usually followed by a linear voltage regulator. A rectifier circuit is necessary to convert a signal having zero average value into a non-zero average value. A rectifier transforms alternating current into direct current by limiting or regulating the direction of flow of current. The output resulting from a rectifier is a pulsating D.C. voltage. This voltage is not appropriate for the components that are going to work through it.
The ripple of the D.C. voltage is smoothened using a filter capacitor of 1000 µF, 25V. The filter capacitor stores electrical charge. If it is large enough the capacitor will store charge as the voltage rises and give up the charge as the voltage falls. This has the effect of smoothing out the waveform and provides steadier voltage output. A filter capacitor is connected at the rectifier output and the d.c. voltage is obtained across the capacitor. When this capacitor is used in this project, it should be twice the supply voltage. When the filter is used, the RC charge time of the filter capacitor must be short and the RC discharge time must be long to eliminate ripple action. In other words the capacitor must charge up fast, preferably with no discharge.

When the rectifier output voltage is increasing, the capacitor charges to the peak voltage $V_m$. Just past the positive peak, the rectifier output voltage starts to fall but at this point the capacitor has $+ V_m$ voltage across it. Since the source voltage becomes slightly less than $V_m$, the capacitor will try to send current back through the diode of rectifier. This reverse biases the diode. The diode disconnects or separates the source the source form load. The capacitor starts to discharge through load. This prevents the load voltage from falling to zero. The capacitor continues to discharge until source voltage becomes more than capacitor voltage. The diode again starts conducting and the capacitor is again charged to peak value $V_m$. When capacitor is charging the rectifier supplies the charging through capacitor branch as well as load current, the capacitor sends currents through the load. The rate at which capacitor discharge depends upon time constant RC. The longer the time constant, the steadier is the output voltage. An increase in load current i.e. decrease in resistance makes time constant of discharge path smaller. The ripple increase and dc output voltage $V_{dc}$ decreases. Maximum capacity cannot exceed a certain limit because the larger the capacitance the greater is the current required to charge the capacitor.

The voltage regulator regulates the supply if the supply if the line voltage increases or decreases. The series 78xx regulators provide fixed regulated voltages from 5 to 24 volts. An unregulated input voltage is applied at the IC Input pin i.e. pin 1 which is filtered by capacitor. The out terminal of the IC i.e. pin 3 provides a regular output. The third terminal is connected to ground. While the input voltage may vary over some permissible voltage range, and the output voltage remains constant within specified voltage variation limit. The 78xx IC’s are positive voltage regulators whereas 79xx IC’s are negative voltage regulators.
These voltage regulators are integrated circuits designed as fixed voltage regulators for a wide variety of applications. These regulators employ current limiting, thermal shutdown and safe area compensation. With adequate heat sinking they can deliver output currents in excess of 1 A. These regulators have internal thermal overload protection. It uses output transistor safe area compensation and the output voltage offered is in 2% and 4% tolerance.
DC Series Motor

The motor used in this proposed system is 9V, 1A operated DC series motor. The series motor provides high starting torque and is able to move very large shaft loads when it is first energized. Fig. 3.13 shows the wiring diagram of a series motor. The field winding in this motor is wired in series with the armature winding. This is the attribute that gives the series motor its name. Since the series field winding is connected in series with the armature, it will carry the same amount of current that passes through the armature. For this reason the field is made from heavy-gauge wire that is large enough to carry the load. Since the wire gauge is so large, the winding will have only a few turns of wire. Notice that the series field is identified as S1 and S2. The amount of current that passes through the winding determines the amount of torque the motor shaft can produce. Since the series field is made of large conductors, it can carry large amounts of current and produces large torque.

![Electrical Diagram of DC Series Motor.](image)

Series motor Operation

Operation of the series motor is easy to understand. In Fig. 3.8, the field winding is connected in series with the armature winding. This means that power will be applied to one end of the series field winding and to one end of the armature winding (connected at the brush). When voltage is applied, current begins to flow from negative power supply terminals through the series winding and armature winding. The armature is not rotating when voltage is first applied, and the large conductors used in the armature and field windings will provide the only
resistance in this circuit. Since these conductors are so large, they will have a small amount of resistance. This causes the motor to draw a large amount of current from the power supply. When the large current begins to flow through the field and armature windings, it causes a strong magnetic field to be built. Since the current is so large, it will cause the coils to reach saturation, which will produce the strongest magnetic field possible.

**Producing Back EMF**

The strength of magnetic fields provides the armature shafts with the greatest amount of torque possible. The large torque causes the armature to begin to spin with the maximum amount of power. When the armature begins to rotate, it begins to produce voltage. From the concept of magnetism, anytime a magnetic field passes a coil of wire, a current will be produced. The stronger the magnetic field is or the faster the coil passes the flux lines, the more current will be generated. When the armature begins to rotate, it will produce a voltage that is of opposite polarity to that of the power supply. This voltage is called back voltage, back EMF (electromotive force), or counter EMF. The overall effect of this voltage is that it will be subtracted from the supply voltage so that the motor windings will see a smaller voltage potential. The reduced current will mean that the motor will continue to lose torque as the motor speed increases. Since the load is moving when the armature begins to pick up speed, the application will require less torque to keep the load moving. This works to the motor's advantage by automatically reducing the motor current as soon as the load begins to move. It also allows the motor to operate with less heat buildup.

This condition can cause problems if the series motor ever loses its load. The load could be lost when a shaft breaks or if a drive pin is sheared. When this occurs, the load current is allowed to fall to a minimum, which reduces the amount of back EMF that the armature is producing. Since the armature is not producing a sufficient amount of back EMF and the load is no longer causing a drag on the shaft, the armature will begin to rotate faster and faster. It will continue to increase rotational speed until it is operating at a very high speed. When the armature is operating at high speed, the heavy armature windings will be pulled out of their slots by centrifugal force. When the windings are pulled loose, they will catch on a field winding pole piece and the motor will be severely damaged. This condition is called runaway and you can see why a DC series motor must have some type of runaway protection. A centrifugal switch can be
connected to the motor to de-energize the motor starter coil if the rpm exceeds the set amount. It is difficult to control the speed of series motor by external means because its rpm is determined by the size of its load.
Resistors

Example: Circuit symbol:

Function
Resistors restrict the flow of electric current, for example a resistor is placed in series with a light-emitting diode (LED) to limit the current passing through the LED.

Connecting and soldering
Resistors may be connected either way round. They are not damaged by heat when soldering.

Resistor values - the resistor colour code
Resistance is measured in ohms, the symbol for ohm is an omega Ω. 1 Ω is quite small so resistor values are often given in kΩ and MΩ.

1 kΩ = 1000 Ω  1 MΩ = 1000000 Ω.

Resistor values are normally shown using coloured bands. Each colour represents a number as shown in the table
Most resistors have 4 bands:

- The **first band** gives the **first digit**.
- The **second band** gives the **second digit**.
- The **third band** indicates the **number of zeros**.
- The fourth band is used to show the tolerance (precision) of the resistor, this may be ignored for almost all circuits but further details are given below.

This resistor has red (2), violet (7), yellow (4 zeros) and gold bands. So its value is $270000 \ \Omega = 270 \text{ k}\Omega$.

On circuit diagrams the $\Omega$ is usually omitted and the value is written 270K.
Capacitors

The capacitor's function is to store electricity, or electrical energy.

The capacitor also functions as a filter, passing alternating current (AC), and blocking direct current (DC).

This symbol — || — is used to indicate a capacitor in a circuit diagram.

The capacitor is constructed with two electrode plates facing each other, but separated by an insulator.

When DC voltage is applied to the capacitor, an electric charge is stored on each electrode. While the capacitor is charging up, current flows. The current will stop flowing when the capacitor has fully charged.

When a circuit tester, such as an analog meter set to measure resistance, is connected to a 10 microfarad (µF) electrolytic capacitor, a current will flow, but only for a moment. You can confirm that the meter's needle moves off of zero, but returns to zero right away.

When you connect the meter's probes to the capacitor in reverse, you will note that current once again flows for a moment. Once again, when the capacitor has fully charged, the current stops flowing. So the capacitor can be used as a filter that blocks DC current. (A "DC cut" filter.) However, in the case of alternating current, the current will be allowed to pass. Alternating current is similar to repeatedly switching the test meter's probes back and forth on the capacitor. Current flows every time the probes are switched.

The value of a capacitor (the capacitance), is designated in units called the Farad (F).

The capacitance of a capacitor is generally very small, so units such as the microfarad (10⁻⁶F), nanofarad (10⁻⁹F), and picofarad (10⁻¹²F) are used.

Recently, a new capacitor with very high capacitance has been developed. The Electric Double Layer capacitor has capacitance designated in Farad units. These are known as "Super Capacitors."

Sometimes, a three-digit code is used to indicate the value of a capacitor. There are two ways in which the capacitance can be written. One uses letters and numbers, the other uses only numbers.
In either case, there are only three characters used. [10n] and [103] denote the same value of capacitance. The method used differs depending on the capacitor supplier. In the case that the value is displayed with the three-digit code, the 1st and 2nd digits from the left show the 1st figure and the 2nd figure, and the 3rd digit is a multiplier which determines how many zeros are to be added to the capacitance. Picofarad ( pF ) units are written this way.

For example, when the code is [103], it indicates $10 \times 10^3$, or $10,000\text{pF} = 10\text{nF} = 0.01\text{µF}$.

If the code happened to be [224], it would be $22 \times 10^4 = 220,000\text{pF} = 220\text{nF} = 0.22\text{µF}$.

Values under 100pF are displayed with 2 digits only. eg, 47 would be 47pF.

The capacitor has an insulator( the dielectric ) between 2 sheets of electrodes. Different kinds of capacitors use different materials for the dielectric.
**Diodes**

Example: Circuit symbol:

**Function**

Diodes allow electricity to flow in only one direction. The arrow of the circuit symbol shows the direction in which the current can flow. Diodes are the electrical version of a valve and early diodes were actually called valves.

**Forward Voltage Drop**

Electricity uses up a little energy pushing its way through the diode, rather like a person pushing through a door with a spring. This means that there is a small voltage across a conducting diode, it is called the **forward voltage drop** and is about 0.7V for all normal diodes which are made from silicon. The forward voltage drop of a diode is almost constant whatever the current passing through the diode so they have a very steep characteristic (current-voltage graph).

**Reverse Voltage**

When a reverse voltage is applied a perfect diode does not conduct, but all real diodes leak a very tiny current of a few µA or less. This can be ignored in most circuits because it will be very much smaller than the current flowing in the forward direction. However, all diodes have a **maximum reverse voltage** (usually 50V or more) and if this is exceeded the diode will fail and pass a large current in the reverse direction, this is called **breakdown**.

Ordinary diodes can be split into two types: Signal diodes which pass small currents of 100mA or less and Rectifier diodes which can pass large currents. In addition there are LEDs and Zener diodes.
LEDs

Light Emitting Diodes

Example: Circuit symbol:

Function

LEDs emit light when an electric current passes through them.

Connecting and soldering

LEDs must be connected the correct way round, the diagram may be labelled a or + for anode and k or - for cathode (yes, it really is k, not c, for cathode!). The cathode is the short lead and there may be a slight flat on the body of round LEDs. If you can see inside the LED the cathode is the larger electrode (but this is not an official identification method).

LEDs can be damaged by heat when soldering, but the risk is small unless you are very slow. No special precautions are needed for soldering most LEDs.
Software Description

```matlab
i = imread('apple2.jpg');
[mask, c, r] = roipoly(i);
red = immultiply(mask, i(:,:,1));
green = immultiply(mask, i(:,:,2));
blue = immultiply(mask, i(:,:,3));
g = cat(3, red, green, blue);
figure, imshow(g);
j = rgb2hsi(g);
figure, imshow(j)
hue = j(:,:,1);
figure, imshow(hue)
hue1 = histeq(hue);
figure, imshow(hue1)
[p, npix] = histroi(hue, c, r);
figure, bar(p, 1)/numel(hue)
figure, plot(p)
z = max(p)
[z, l] = max(p)
 fid = fopen('exp.txt', 'W');
fprintf(fid, '%d
', l);
fclose(fid)
s = serial('COM1');
fopen(s)
[ss, errmsg] = sprintf('value = %2d ', l)
%fprintf(s, '%s', 'X-index of peak ')
%fprintf(s, '%s', ss)
fclose(s)
```
set(gca,'xtick',0:20:150)
set(gca,'ytick',0:50:1500)
sat=j(:,:,2);
inte=j(:,:,3);

w=fspecial('average',25);
I_filtered=imfilter(inte,w,'replicate');
k=cat(3,hue,sat,I_filtered);
v=p(:,:,1);
t=hsi2rgb(k);
figure,imshow(t)
t=min(t,1);
figure,imshow(t);
[M,N,K]=size(t);
I=reshape(t, M*N,3);
idx=find(mask);
I=double(I(idx, 1:3));
[c,m]=covmatrix(I);
d=diag(c);
sd=sqrt(d)'
output=sd(1)
E=colorseg('euclidean',t,output,m);
figure,imshow(E)

%tm = timer('singleshot','5','drop','on')
%wait(tm);
ii=0
while(ii==12000)
    jj=0
    while(jj==10000)
        jj = jj + 1;
    ii = ii + 1 ;
end
end

fopen(s)
if (l<=2)
    grading='grade A'
    \[ ss, errmsg = sprintf('And SD = %0.3f ',output) \]
    \% fprintf(s,'%s',ss)
    fprintf(s,'%s','A')
elseif (2<l && l<=5)
    grading='grade B'
    \[ ss, errmsg = sprintf('And SD = %0.3f ',output) \]
    \% fprintf(s,'%s',ss)
    fprintf(s,'%s','B')
elseif (5<l && l<=10)
    grading='grade C'
    \[ ss, errmsg = sprintf('And SD = %0.3f ',output) \]
    \% fprintf(s,'%s',ss)
    fprintf(s,'%s','C')
elseif (10<l && l<=15)
    grading='grade D'
    \[ ss, errmsg = sprintf('And SD = %0.3f ',output) \]
    \% fprintf(s,'%s',ss)
    fprintf(s,'%s','D')
else
    grading='wrong input'
    \[ ss, errmsg = sprintf('And SD = %0.3f ',output) \]
    \% fprintf(s,'%s',ss)
fprintf(s,'%s','w')

end

switch grading
    case 'grade A'
        disp('the apple is of A grade quality')
    case 'grade B'
        disp('the apple is of B grade quality')
    case 'grade C'
        disp('the apple is of C grade quality')
    case 'grade D'
        disp('the apple is of D grade quality')
    otherwise
        error('INVALID CHOICE')
end

fclose(s)
What is MATLAB?

MATLAB is a high-performance language for technical computing. It integrates computation, visualization, and programming in an easy-to-use environment where problems and solutions are expressed in familiar mathematical notation. Typical uses include

- Math and computation
- Algorithm development
- Data acquisition Modeling, simulation, and prototyping
- Data analysis, exploration, and visualization
- Scientific and engineering graphics
- Application development, including graphical user interface building

MATLAB is an interactive system whose basic data element is an array that does not require dimensioning. This allows you to solve many technical computing problems, especially those with matrix and vector formulations, in a fraction of the time it would take to write a program in a scalar non-interactive language such as C or Fortran. The name MATLAB stands for *matrix laboratory*. MATLAB is the tool of choice for high-productivity research, development, and analysis. MATLAB features a family of add-on application-specific solutions called toolboxes. Very important to most users of MATLAB, toolboxes allow you to learn and apply specialized technology. Toolboxes are comprehensive collections of MATLAB functions (M-files) that extend the MATLAB environment to solve particular classes of problems. Areas in which toolboxes are available include signal processing, control systems, neural networks, fuzzy logic, wavelets, simulation, and many others.

Images

Two-dimensional arrays can be displayed as images, where the array elements determine brightness or color of the images.

For example, the statements load durer

whos

<table>
<thead>
<tr>
<th>Name</th>
<th>Size</th>
<th>Bytes</th>
<th>Class</th>
</tr>
</thead>
</table>
X        648x509    2638656  double array
caption   2x28       112    char array
map       128x3     3072   double array

load the file durer.mat, adding three variables to the workspace. The matrix X is a 648-by-509 matrix and map is a 128-by-3 matrix that is the colormap for this image.

MAT-files, such as durer.mat, are binary files that can be created on one platform and later read by MATLAB on a different platform.
The elements of X are integers between 1 and 128, which serve as indices into the colormap, map. Then

\begin{verbatim}
    image(X)
colormap(map)
axis image
\end{verbatim}

reproduces Dürer's etching shown at the beginning of this book. A high-resolution scan of the magic square in the upper right corner is available in another file. Type

\begin{verbatim}
    load detail
\end{verbatim}

and then use the up arrow key on your keyboard to reexecute the image, colormap, and axis commands. The statement

\begin{verbatim}
    colormap(hot)
\end{verbatim}

adds some twentieth century colorization to the sixteenth century etching. The function hot generates a colormap containing shades of reds, oranges, and yellows. Typically a given image matrix has a specific colormap associated with it. See the colormap reference page for a list of other predefined colormaps.

**Reading and Writing Images**

You can read standard image files (TIFF, JPEG, BMP, etc.) into MATLAB using the `imread` function. The type of data returned by `imread` depends on the type of image you are reading.

You can write MATLAB data to a variety of standard image formats using the `imwrite` function.
Flow Control

MATLAB has several flow control constructs:

- if
- switch and case
- for
- while
- continue
- break
- try - catch
- return

if

The if statement evaluates a logical expression and executes a group of statements when the expression is true. The optional else if and else keywords provide for the execution of alternate groups of statements. An end keyword, which matches the if, terminates the last group of statements. The groups of statements are delineated by the four keywords -- no braces or brackets are involved.

While

The while loop repeats a group of statements an indefinite number of times under control of a logical condition. A matching end delineates the statements.

The Colon Operator

The colon, ;, is one of the most important MATLAB operators. It occurs in several different forms. The expression

1:10

is a row vector containing the integers from 1 to 10,

1 2 3 4 5 6 7 8 9 10

Subscript expressions involving colons refer to portions of a matrix. The colon by itself refers to all the elements in a row or column of a matrix and the keyword end refers to the last row or column.

A(:,end) refers to all the elements in the last column of A.
Expressions

Like most other programming languages, MATLAB provides mathematical expressions, but unlike most programming languages, these expressions involve entire matrices. The building blocks of expressions are

- Variables
- Numbers
- Operators
- Functions

Variables
MATLAB does not require any type declarations or dimension statements. When MATLAB encounters a new variable name, it automatically creates the variable and allocates the appropriate amount of storage. If the variable already exists, MATLAB changes its contents and, if necessary, allocates new storage.

Numbers
MATLAB uses conventional decimal notation, with an optional decimal point and leading plus or minus sign, for numbers. Scientific notation uses the letter e to specify a power-of-ten scale factor. Imaginary numbers use either i or j as a suffix.

Operators
Expressions use familiar arithmetic operators and precedence rules.

Functions
MATLAB provides a large number of standard elementary mathematical functions, including abs, sqrt, exp, and sin. Taking the square root or logarithm of a negative number is not an error; the appropriate complex result is produced automatically. MATLAB also provides many more advanced mathematical functions, including Bessel and gamma functions. Most of these functions accept complex arguments.

immultiply
Multiply two images, or multiply an image by a constant

Syntax

\[ Z = \text{immultiply}(X,Y) \]
Description

\[ Z = \text{immultiply}(X,Y) \] multiplies each element in array \( X \) by the corresponding element in array \( Y \) and returns the product in the corresponding element of the output array \( Z \).

If \( X \) and \( Y \) are real numeric arrays with the same size and class, then \( Z \) has the same size and class as \( X \). If \( X \) is a numeric array and \( Y \) is a scalar double, then \( Z \) has the same size and class as \( X \). If \( X \) is logical and \( Y \) is numeric, then \( Z \) has the same size and class as \( Y \).

If \( X \) is numeric and \( Y \) is logical, then \( Z \) has the same size and class as \( X \). \text{immultiply} computes each element of \( Z \) individually in double-precision floating point.

If \( X \) is an integer array, then elements of \( Z \) exceeding the range of the integer type are truncated, and fractional values are rounded. If \( X \) and \( Y \) are double arrays, you can use the expression \( X.*Y \) instead of this function.

\textbf{cat}

Concatenate arrays

Syntax

\[ C = \text{cat}(\text{dim},A,B) \]
\[ C = \text{cat}(\text{dim},A1,A2,A3,A4...) \]

Description

\[ C = \text{cat}(\text{dim},A,B) \] concatenates the arrays \( A \) and \( B \) along \( \text{dim} \).

\[ C = \text{cat}(\text{dim},A1,A2,A3,A4,...) \] concatenates all the input arrays (\( A1, A2, A3, A4, \) and so on) along \( \text{dim} \). \text{cat}(2,A,B) is the same as \([A,B]\), and \text{cat}(1,A,B) is the same as \([A;B]\).

Remarks: When used with comma-separated list syntax, \text{cat}(\text{dim},C{[\cdot]})) or \text{cat}(\text{dim},C.\text{field}) is a convenient way to concatenate a cell or structure array containing numeric matrices into a single matrix.

\textbf{histeq}

Enhance contrast using histogram equalization

Syntax

\[ J = \text{histeq}(I,hgram) \]
\[ J = \text{histeq}(I,n) \]
\[ [J,T] = \text{histeq}(I,...) \]

\[
\text{newmap} = \text{histeq}(X,\text{map},\text{hgram})
\]

\[
\text{newmap} = \text{histeq}(X,\text{map})
\]

\[
[\text{newmap},T] = \text{histeq}(X,...)
\]

**Description**

`histeq` enhances the contrast of images by transforming the values in an intensity image, or the values in the colormap of an indexed image, so that the histogram of the output image approximately matches a specified histogram.

\( J = \text{histeq}(I,hgram) \) transforms the intensity image \( I \) so that the histogram of the output intensity image \( J \) with \( \text{length}(hgram) \) bins approximately matches \( hgram \). The vector \( hgram \) should contain integer counts for equally spaced bins with intensity values in the appropriate range: \([0, 1]\) for images of class double, \([0, 255]\) for images of class uint8, and \([0, 65535]\) for images of class uint16. `histeq` automatically scales \( hgram \) so that \( \text{sum}(hgram) = \text{prod}(%text{size}(I)) \). The histogram of \( J \) will better match \( hgram \) when \( \text{length}(hgram) \) is much smaller than the number of discrete levels in \( I \).

\( J = \text{histeq}(I,n) \) transforms the intensity image \( I \), returning in \( J \) an intensity image with \( n \) discrete gray levels. A roughly equal number of pixels is mapped to each of the \( n \) levels in \( J \), so that the histogram of \( J \) is approximately flat. \( \) (The histogram of \( J \) is flatter when \( n \) is much smaller than the number of discrete levels in \( I \).) The default value for \( n \) is 64.

\[ [J,T] = \text{histeq}(I,...) \] returns the grayscale transformation that maps gray levels in the intensity image \( I \) to gray levels in \( J \).

\[
\text{newmap} = \text{histeq}(X,\text{map},\text{hgram})
\]

transforms the colormap associated with the indexed image \( X \) so that the histogram of the gray component of the indexed image \((X,\text{newmap})\) approximately matches \( hgram \). The `histeq` function returns the transformed colormap in \( \text{newmap} \). \( \text{length}(hgram) \) must be the same as \( \text{size}(\text{map},1) \).

\[
\text{newmap} = \text{histeq}(X,\text{map})
\]

transforms the values in the colormap so that the histogram of the gray component of the indexed image \( X \) is approximately flat. It returns the transformed colormap in \( \text{newmap} \).

\[ [\text{newmap},T] = \text{histeq}(X,...) \] returns the grayscale transformation \( T \) that maps the gray component of \( \text{map} \) to the gray component of \( \text{newmap} \).
Class Support

For syntaxes that include an intensity image I as input, I can be of class uint8, uint16, or double, and the output image J has the same class as I. For syntaxes that include an indexed image X as input, X can be of class uint8 or double; the output colormap is always of class double. Also, the optional output T (the gray-level transform) is always of class double.

plot

Linear 2-D plot

Syntax

plot(Y)
plot(X1,Y1,...)
plot(X1,Y1,LineSpec,...)
plot(...,'PropertyName',PropertyValue,...)
plot(axes_handle,...)

h = plot(...)
hlines = plot('v6',...)

Description

plot(Y) plots the columns of Y versus their index if Y is a real number. If Y is complex, plot(Y) is equivalent to plot(real(Y),imag(Y)). In all other uses of plot, the imaginary component is ignored.

plot(X1,Y1,...) plots all lines defined by Xn versus Yn pairs. If only Xn or Yn is a matrix, the vector is plotted versus the rows or columns of the matrix, depending on whether the vector's row or column dimension matches the matrix. plot(X1,Y1,LineSpec,...) plots all lines defined by the Xn,Yn,LineSpec triples, where LineSpec is a line specification that determines line type, marker symbol, and color of the plotted lines. You can mix Xn,Yn,LineSpec triples with Xn,Yn pairs:

plot(X1,Y1,X2,Y2,LineSpec,X3,Y3

fprint

To print from a port.

fclose

Close one or more open files
status = fclose(fid)
status = fclose('all')

Description

status = fclose(fid) closes the specified file if it is open, returning 0 if successful and -1 if unsuccessful. Argument fid is a file identifier associated with an open file. (See fopen for a complete description of fid).

status = fclose('all') closes all open files (except standard input, output, and error), returning 0 if successful and -1 if unsuccessful
Future Scope

This system can be useful tool for both the buyer and the seller in order to increase the efficiency and quality of the product. Due to the low cost of the system it can be used in small scale industries-juice and jam making units.
Conclusion

A Scientific & economical analysis of the quality of a fruit or any product whose quality depends on the color is developed. By using this project we are not only analyzing the product but also controlling the final product quality with the help of which one can check the quality of the fruit and can easily control the quality of the final product.

So, this project can be a good tool in the hands of the farmers who are dealing with fruits or vegetables business. Moreover, the accuracy of this is better than the manual process and above all the speed of operation is very high.
References


Net links

Appendix

Color Content

Color Look Up Table

Color Histogram

RGB Image

Luminosity Image

Red/Green Image

Blue/Yellow Image