SREE NARAYANA MANGALAM INSTITUTE OF MANAGEMENT & TECHNOLOGY

MALIANKARA P.O., N. PARAVUR, ERNAKULAM - 683516

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

PROJECT REPORT

ON

ADAPTIVE METHOD FOR HAND GESTURE RECOGNITION

Submitted by

BILBEEYA C BABU (Reg. No. : 13017965)

GOPIKA SIDHARTHAN (Reg. No. : 13017970)

GOVIND RAJ(Reg. No. : 13017971)

MANASI K.A(Reg. No. : 13017975)

2016-2017

SREE NARAYANA MANGALAM INSTITUTE OF MANAGEMENT & TECHNOLOGY

MALIANKARA P.O., N. PARAVUR, ERNAKULAM - 683516

CERTIFICATE

Certified that this is the bonafide project report on

ADAPTIVE METHOD FOR HAND GESTURE RECOGNITION

submitted by

GOVIND RAJ (Reg. No. : 13017971)

during the year 2016-2017 in partial fulfilment of the requirement for the award of the Degree of Bachelor of Technology

in

Electronics & Communication Engineering, under Mahatma Gandhi University, Kerala

RAJAN N SEETHA GOPI V VISHNUPRIYA V	SREELAKSHMY R	RAJAN N
Project Co-ordinators	*Internal guide*	*Head of the Dept.*

ACKNOWLEDGEMENT

We express our heartfelt gratitude to almighty, the supreme guide for bestowing his blessings in our entire endeavor.

We wish to give deep sense of acknowledgement to our principal **Dr.P.A.RAJAN** for his constant encouragement and valuable advice throughout the course.

We are deeply indebted to **Mr. RAJAN N**, Head of the Department of Electronics and Communication Engineering, for his sincere and dedicated cooperation and encouragement throughout the duration of our Project.

We could also like to thank our Project coordinators **Mr. RAJAN N, Mrs. SEETHA GOPI V** and **Mrs. VISHNUPRIYA V** and Project Guide **Mrs. SREELAKSHMY R** Assistant Professors, Department of Electronics and Communication Engineering for their valuable advice and whole hearted cooperation without which this project would not have seen the light of the day. We express our sincere gratitude to all the faculty members of the Department of Electronics and Communication Engineering for their cooperation and support to our project.

We are highly obliged to our parents for their encouragement and we express our sincere thanks to each and every one who has directly or indirectly helped us during the period of this project.

ABSTRACT

In recent years, the gesture control technique has become a new developmental trend for many human-based electronics products. This technique let people can control devices more naturally, intuitively and conveniently. In this paper, a fast gesture recognition scheme is proposed to be an interface for the human-machine interaction (HMI) of systems. This paper presents some low-complexity algorithms and gestures to reduce the gesture recognition complexity and be more suitable for controlling real-time computer systems. Here we control the movements of a vehicle using some hand gestures.

In this project user's hand gestures are captured using a webcam and it is then subjected to gesture recognition process. Gestures recognition is done using MATLAB. By recognizing the input gestures corresponding commands and pass to the microcontroller in the vehicle through Zigbee wireless communication. Then the working of vehicle is controlled by the microcontroller.

We are trying to make a gesture to language conversion system. When a gesture is shown on the webcam, the camera takes the image. The features of the image are extracted by using MatLab. Then this compares with the database. And corresponding alphabet is displayed. And also this gesture can be used to control a device.

CONTENTS

CHAPTER 1

INTRODUCTION

Gestures are the motion of the body or physical action form by the user in order to convey some meaningful information. Gesture recognition is the process by which gesture made by the user is made known to the system. Through the use of computer vision or machine eye, there is great emphasis on using hand gesture as a substitute of new input modality in broad range applications. A primary goal of gesture recognition is to create a system which can identify specific human gestures and use them to convey information for device control and by implementing real time gesture recognition a user can control a computer by doing a specific gesture in front of a video camera linked to the computer.

With the massive influx of computers in society, human computer interaction, or HCI, has become an increasingly important part of daily lives. Current user interaction devices with keyboard, mouse and pen are not sufficient for physically challenged people and Virtual Environment (VE) which induce many new types of representation and interaction. Gesture, speech, and touch inputs are few possible methods of meeting such user's need to solve this problem. . To achieve nature human-computer interaction for VE application and the disabled people, the human hand could be an input device. Numerous approaches have been proposed for enabling hand gesture recognition. A common taxonomy is based on whether extra devices are required for raw data collecting. In this method, they are categorized into

1. Data glove based hand gesture recognition,

2. Vision based hand gesture recognition.

For digitizing hand and finger motions into multiparametric data, data-glove based methods use sensors. The extra sensors make it easy to collect hand configuration and movement. However, the extra devices are quite expensive and bring much cumbersome experience to the users. In contrast, the Vision Based methods require only a camera, thus realizing a natural interaction between humans and computers without the use of any extra devices.

1.1 Vision based hand gesture recognition

In vision based hand gesture recognition system, technology uses a bare hand to extract data for recognition. With the help of this technology user can directly interact with the system. In vision based hand gesture recognition system , the movement of the hand is recorded by video camera(s).Vision based technology deals with some image characteristics such as texture and color for acquiring data needed for gesture analyze.

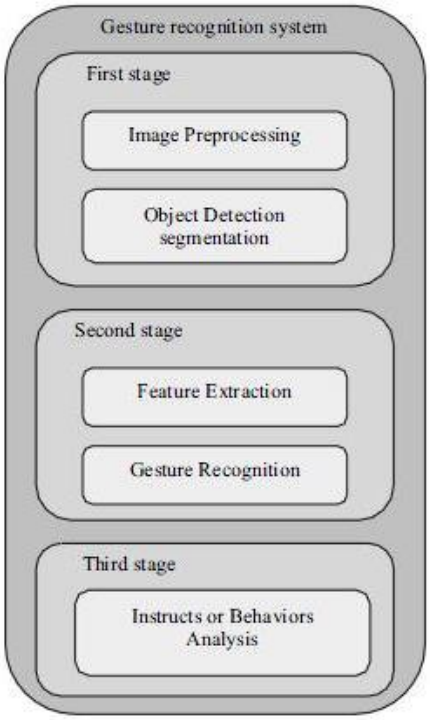


Fig 1.1 A common gesture recognition system [1]

Following is the figure showing a gesture recognition system using image processing. Figure 1.1. Common Gesture recognition system Most gesture recognition methods usually contain three major stages. The first stage is the object detection. The target of this stage is to detect hand objects in the digital images or videos. Many environment and image problems are needed to solve at this stage to ensure that the hand contours or regions can be extracted precisely to enhance the recognition accuracy. Common image problems contain unstable brightness, noise, poor resolution and contrast. The better environment and camera devices can effectively improve these problems. However, it is hard to control when the gesture recognition system is working in the real environment or is become a product. Hence, the image processing method is a better solution to solve these image problems to construct an adaptive and robust gesture recognition system. The second stage is object recognition. The detected hand objects are recognized to identify the gestures. At this stage, differentiated features and effective classifiers selection are a major issue in most researches.

CHAPTER 2

PROBLEM IDENTIFICATION

The researches done in this field are mostly done using a glove based system. In the glove based system, sensors such as potentiometer, accelerometers etc. are attached to each of the finger. Based on their readings the corresponding alphabet is displayed. Christopher Lee and Yangsheng Xu developed a glove-based gesture recognition system that was able to recognize 14 of the letters from the hand alphabet, learn new gestures and able to update the model of each gesture in the system in online mode. Over the years advanced glove devices have been designed such as the Sayre Glove, Dexterous Hand Master and Power Glove.

The main problem faced by this gloved based system is that it has to be recalibrate every time whenever a new user uses this system. Also the connecting wires restrict the freedom of movement. The present systems which focuses on vision based approach are implemented using kinect camera but these are more costly. This project focuses on reducing cost and improving robustness of system using simple web camera

CHAPTER 3

LITERATURE REVIEW

Not many Researches have been carried out in this particular field, especially in Binary Sign Language Recognition. Few researches have been done on this issue though and some of them are still operational, but nobody was able to provide a full fledged solution to the problem. Christopher Lee and Yangsheng Xu developed a glove-based gesture recognition system that was able to recognize 14 of the letters from the hand alphabet, learn new gestures and able to update the model of each gesture in the system in online mode, with a rate of 10Hz. Over the years advanced glove devices have been designed such as the Sayre Glove, Dexterous Hand Master and Power Glove

The most successful commercially available glove is by far the VPL Data Glove.It was developed by Zimmerman during the 1970's. It is based upon patented optical fiber sensors along the back of the fingers. Star-ner and Pentland developed a glove-environment system capable of recognizing 40 signs from the American Sign Language (ASL) with a rate of 5Hz.

Another research is by Hyeon-Kyu Lee and Jin H. Kim presented work on real-time hand-gesture recognition using HMM (Hidden Markov Model). Kjeldsen and Kendersi devised a technique for doing skin-tone segmentation in HSV space, based on the premise that skin tone in images occupies a connected volume in HSV space. They further developed a system which used a back-propagation neural network to recognize gestures from the segmented hand images.

Etsuko Ueda and Yoshio Matsumoto presented a novel technique a hand-pose estimation that can be used for vision-based human interfaces, in this method, the hand regions are extracted from multiple images obtained by a multi viewpoint camera system, and constructing the "voxel Model" Ibraheem and

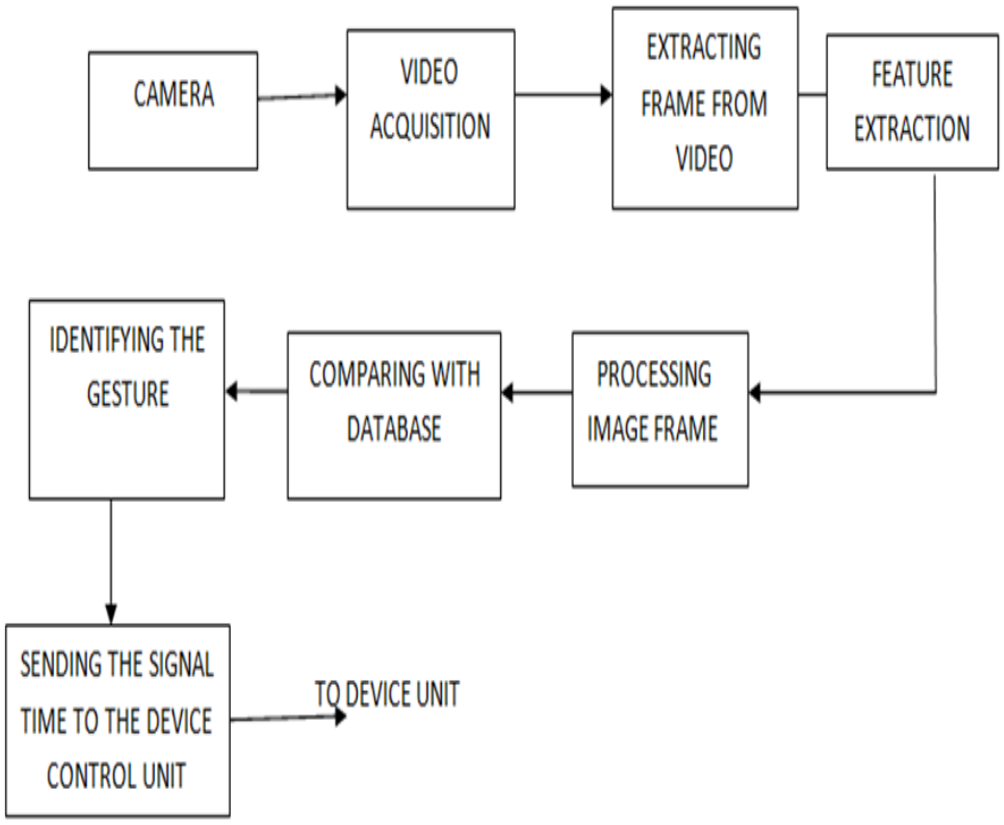
Khan have reviewed various techniques for gesture recognition and recent gesture recognition approaches.

Ghotkar et al. used Cam shift method and Hue, Saturation; Intensity (HSV) color for model for hand tracking and segmentation .For gesture recognition Genetic Algorithm is used. Paulraj M P et al. had developed a simple sign language recognition system that has been developed using skin color segmentation and Artificial Neural Network.

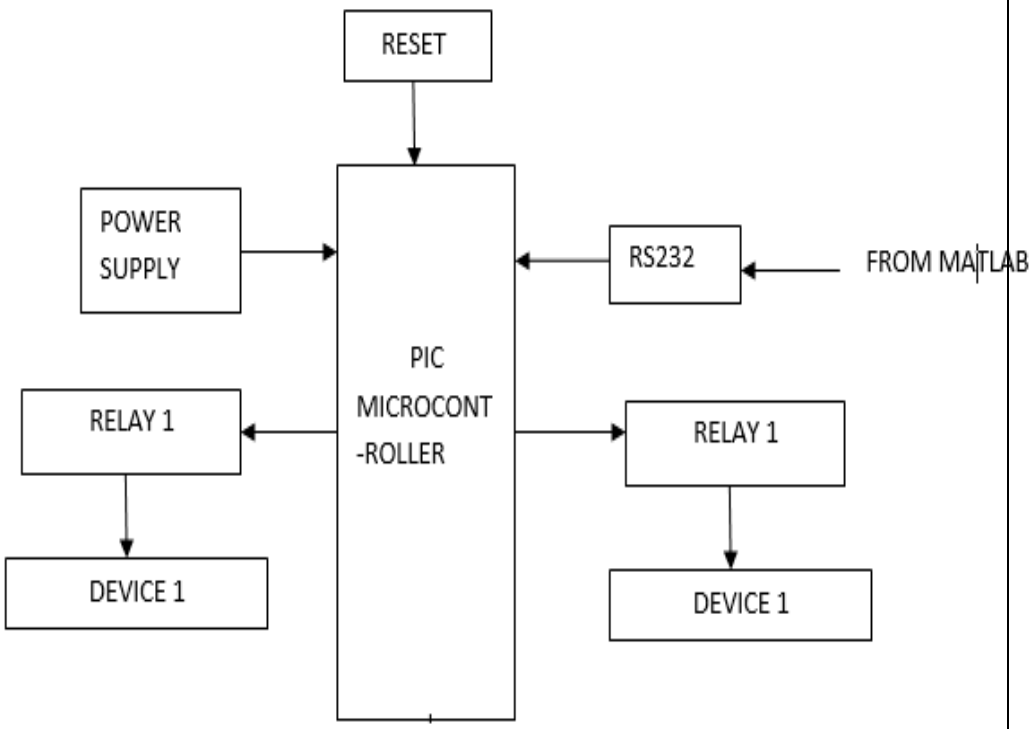
CHAPTER 4

BLOCK DIAGRAM

4.1 MATLAB SECTION



4.2 RECEIVER SECTION



4.3 BLOCK DIAGRAM DESCRIPTION

4.3.1 PIC Microcontroller

The main part of the system is PIC microcontroller. PIC (Peripheral Interface Controller) is a family of modified Harvard architecture microcontrollers made by Microchip Technology.

4.3.2 Power Supply

Here we use 7805 power supply circuit to provide 5V DC to the microcontroller. We can use 9V DC battery or 12V and 1A adapter to provide the supply to the circuit. 7805 is a 5V fixed three terminal positive voltage regulator IC. The IC has features such as safe operating area protection, thermal shut down, internal current limiting which makes the IC very rugged.

4.3.3 Zigbee Module

This is the wireless technology implanted in the project for the transmission purpose. Since Matlab system is placed at a distance from the display section data is to be transmitted from PIC microcontroller to Matlab system. Zigbee transmission is more secure than other wireless system which can be implemented like blue tooth.We can encrypt the data in Zigbee system if we needed. Easiness of installation is another reason for selecting Zigbee module.

4.3.4 Relay

They are often used to interface an electronic circuit (working at a low voltage) to an electrical circuit which works at very high voltage. Here 12V DC 1A dpdt miniature relay is used.

4.3.5 PC (Personal Computer)

It is used for data processing and recognizing. MATLAB programming is implemented within the PC

It involves the processes:

- Signal processing
- Feature selection
- Feature extractions

4.3.6 Camera

Input Gestures are captured using the web camera. The camera used here is iball CHD 20.0. It is a 720P High-Definition Webcam with exceptional image quality

4.3.7 RS-232

RS-232 is a standard communication protocol for linking computer and its peripheral devices to allow serial data exchange. In simple terms RS232 defines the voltage for the path used for data exchange between the devices. It specifies common voltage and signal level, common pin wire configuration and minimum, amount of control signals.

CHAPTER 5

HARDWARE DESCRIPTION

For designing this hardware many types of devices are used to make it perfectly working. All the devices are purchased from different manufacturers. These components are soldered on a soldering board. The following list of hardware is required for this system

- Power supply
- Microcontroller PIC16F877A
- Camera
- Zigbee
- Relay
- 12 V adapter
- Voltage Regulator
- MAX-232

5.1 PIC16F877A MICROCONTROLLER

PIC 16F877 is one of the most advanced microcontroller from Microchip. This controller is widely used for experimental and modern applications because of its low price, wide range of applications, high quality, and ease of availability. It is ideal for applications such as machine control applications, measurement devices, study purpose, and so on. It is one of the PICMicro Family microcontroller which is popular at this moment, start from beginner until all professionals because of its very easy using **PIC16F877A** and use FLASH memory technology so that can be write-erase until thousand times.

PIC16F877A perfectly fits many uses, from automotive industries and controlling home appliances to industrial instruments, remote sensors, electrical door locks and safety devices. It is also ideal for smart cards as well as for battery supplied devices because of its low consumption. EEPROM memory makes it easier to apply microcontrollers to devices where permanent storage of various parameters is needed (codes for transmitters, motor speed, receiver frequencies, etc.). Low cost, low consumption, easy handling and flexibility make **PIC16F877A**applicable even in areas where microcontrollers had not previously been considered (example: timer functions, interface replacement in larger systems, coprocessor applications, etc.). In System Programmability of this chip (along with using only two pins in data transfer) makes possible the flexibility of a product, after assembling and testing have been completed. This capability can be used to create assembly-line production, to store calibration data available only after final testing, or it can be used to improve programs on finished products.

5.1.1 Features of PIC16F877

The PIC16FXX series has more advanced and developed features when compared to its previous series. The important features of PIC16F877 series is given below.

- **General Features**

  - High performance RISC CPU.
  - Only 35 simple word instructions.
  - All single cycle instructions except for program branches which are two cycles.
  - Operating speed: clock input (200MHz), instruction cycle (200nS).

- Up to 368×8bit of RAM (data memory), 256×8 of EEPROM (data memory), 8k×14 of flash memory.
- Pin out compatible to PIC 16C74B, PIC 16C76, PIC 16C76.
- Eight level deep hardware stack.
- Interrupt capability (up to 14 sources).
- Different types of addressing modes (direct, Indirect, relative addressing modes).
- Power on Reset (POR).
- Power-Up Timer (PWRT) and oscillator start-up timer.
- Low power- high speed CMOS flash/EEPROM.
- Fully static design.
- Wide operating voltage range (2.0 – 5.56)volts.
- High sink/source current (25mA).
- Commercial, industrial and extended temperature ranges.
- Low power consumption (<0.6mA typical @3v-4MHz, 20µA typical @3v-32MHz and <1 A typical standby)

➢ **Peripheral Features**

- Timer 0: 8 bit timer/counter with pre-scalar.
- Timer 1:16 bit timer/counter with pre-scalar.
- Timer 2: 8 bit timer/counter with 8 bit period registers with pre-scalar and post-scalar.
- Two Capture (16bit/12.5nS), Compare (16 bit/200nS), Pulse Width Modules (10bit).
- 10bit multi-channel A/D converter

- Synchronous Serial Port (SSP) with SPI (master code) and I2C (master/slave).
- Universal Synchronous Asynchronous Receiver Transmitter (USART) with 9 bit address detection.
- Parallel Slave Port (PSP) 8 bit wide with external RD, WR and CS controls (40/46pin).
- Brown Out circuitry for Brown-Out Reset (BOR).

➢ **Key Features**

- Maximum operating frequency is 20MHz.
- Flash program memory (14 bit words), 8KB.
- Data memory (bytes) is 368.
- EEPROM data memory (bytes) is 256.
- 5 input/output ports.
- 3 timers.
- 2 CCP modules.
- 2 serial communication ports (MSSP, USART).
- PSP parallel communication port
- 10bit A/D module (8 channels)

➢ **Analog Features**

- 10bit, up to 8 channel A/D converter.
- Brown Out Reset function.
- Analog comparator module.

➢ **Special Features**

- 100000 times erase/write cycle enhanced memory.
- 1000000 times erase/write cycle data EEPROM memory.
- Self programmable under software control.
- In-circuit serial programming and in-circuit debugging capability.
- Single 5V,DC supply for circuit serial programming
- WDT with its own RC oscillator for reliable operation.
- Programmable code protection.

5.1.2 Pin Diagrams

PIC16F877 chip is available in different types of packages. According to the type of applications and usage, these packages are differentiated. The pin diagrams of a PIC16F877 chip in different packages are shown in the figure below.

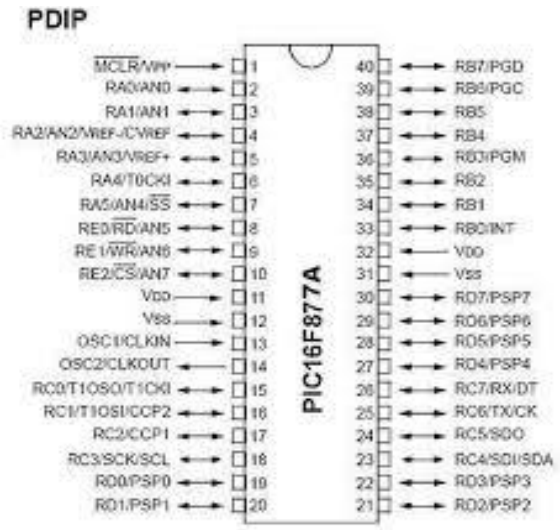


Fig 5.1: PIC16F877A Pin diagram [6]

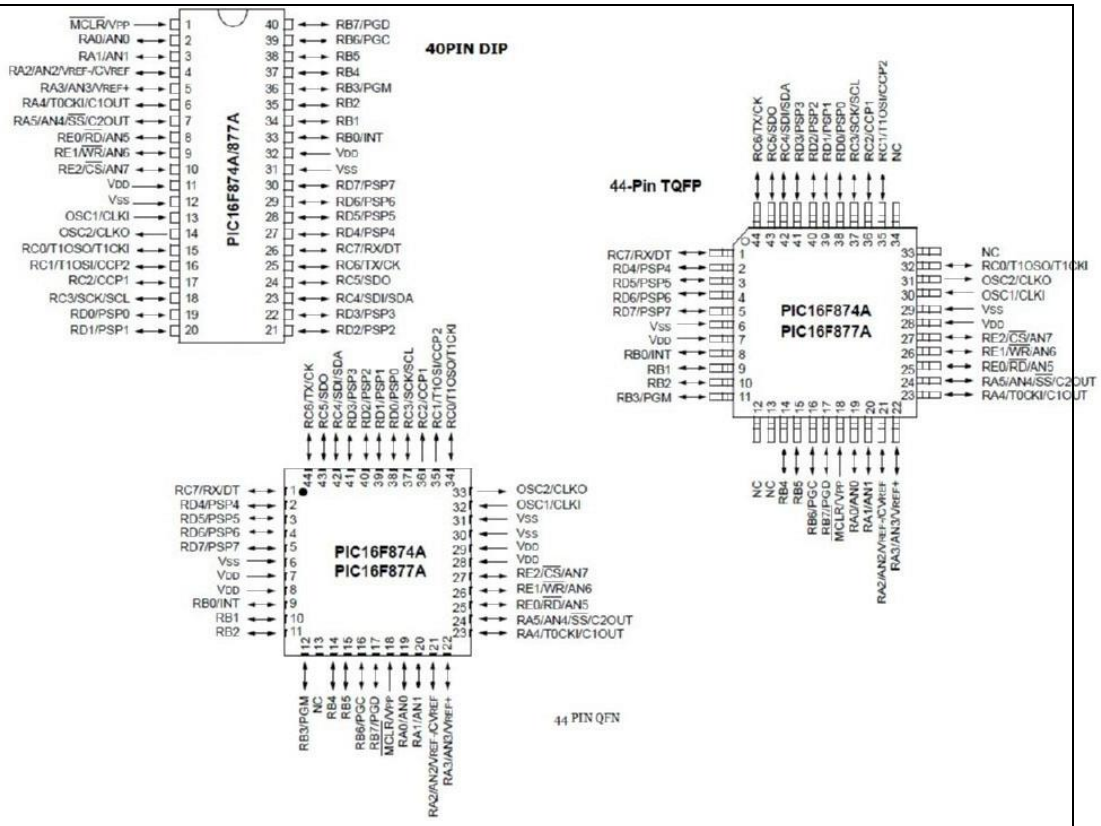


Fig 5.2: PIC16F877A Pin diagrams [6]

5.1.3 Input/output ports

PIC16F877 has 5 basic input/output ports. They are usually denoted by PORT A (R A), PORT B (RB), PORT C (RC), PORT D (RD), and PORT E (RE). These ports are used for input/ output interfacing. In this controller, “PORT A” is only 6 bits wide (RA-0 to RA-7), ”PORT B” , “PORT C”,”PORT D” are only 8 bits wide (RB-0 to RB-7,RC-0 to RC-7,RD-0 to RD-7), ”PORT E” has only 3 bit wide (RE-0 to RE-7).

All these ports are bi-directional. The direction of the port is controlled by using TRIS(X) registers (TRIS A used to set the direction of PORT-A, TRIS B used to set the direction for PORT-B, etc.). Setting a TRIS(X) bit ‘1’ will set the corresponding PORT(X) bit as input. Clearing a TRIS(X) bit ‘0’ will set the corresponding PORT(X) bit as output.

(If we want to set PORT A as an input, just set TRIS(A) bit to logical '1' and want to set PORT B as an output, just set the PORT B bits to logical '0'.)

Analog input port (AN0 TO AN7) : these ports are used for interfacing analog inputs.

TX and RX: These are the USART transmission and reception ports.

SCK: these pins are used for giving synchronous serial clock input.

SCL: these pins act as an output for both SPI and I2C modes.

DT: these are synchronous data terminals.

CK: synchronous clock input.

SD0: SPI data output (SPI Mode).

SD1: SPI Data input (SPI mode).

SDA: data input/output in I2C Mode.

CCP1 and CCP2: these are capture/compare/PWM modules.

OSC1: oscillator input/external clock.

OSC2: oscillator output/clock out.

MCLR: master clear pin (Active low reset).

Vpp: programming voltage input.

THV: High voltage test mode controlling.

Vref (+/-): reference voltage.

SS: Slave select for the synchronous serial port.

T0CK1: clock input to TIMER 0.

T1OSO: Timer 1 oscillator output.

T1OS1: Timer 1 oscillator input.

T1CK1: clock input to Timer 1.

PGD: Serial programming data.

PGC: serial programming clock.

PGM: Low Voltage Programming input.

INT: external interrupt.

RD: Read control for parallel slave port.

CS: Select control for parallel slave.

PSP0 to PSP7: Parallel slave port.

VDD: positive supply for logic and input pins.

VSS: Ground reference for logic and input/output pins

5.2 ZIGBEE

Zigbee is a wireless technology developed as an open global standard to address the unique needs of low cost, low power, wireless sensor networks. The standard takes full advantage of the IEEE 802.15.4 physical radio specification and operates in unlicensed bands worldwide at the following frequencies: 2.400-2.484 GHz, 902-928 MHz and 868-868.6MHz.The 802.15.4 specification was developed at the institute of Electrical and Electronics Engineers (IEEE). The specification is a packet based radio protocol that meets the need of low-

cost, battery operated devices. The protocol allows device to intercommunicate and be powered by batteries that last years instead of hours.

The ZigBee protocol was engineered by the zigbee alliance, a non-profit consortium of leading semiconductor manufacturers, technology providers, OEMs and end-users worldwide. The protocol was designed to provide OEMs and integrators with an easy-to-use wireless data solution characterized by low power consumption, support for multiple network structures and secure connections.

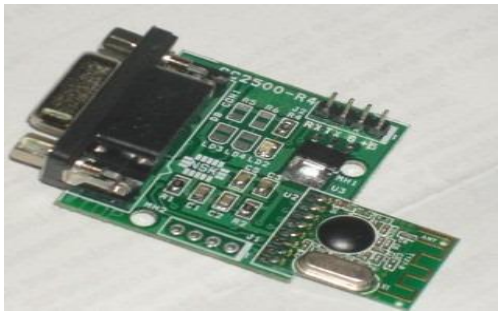


Fig 5.3 Zigbee [6]

The CC2500 is a low-cost 2.4 GHz transceiver designed for very low-power wireless applications. The circuit is intended for the 2400-2483.5 MHz ISM (Industrial, Scientific and Medical) and SRD (Short Range Device) frequency band.

The RF transceiver is integrated with a highly configurable baseband modem. The modem supports various modulation formats and has a configurable data rate up to 500 kBaud.CC2500 provides extensive hardware support for packet handling, data buffering, burst transmissions, clear channel assessment, link quality indication and wake-on-radio.

The main operating parameters and the 64-byte transmit/receive FIFOs of CC2500 can be controlled via an SPI interface. In a typical system, the CC2500 will be used together with a microcontroller and a few additional passive components.

- **Features**

  - RF Performance
  - High sensitivity (-104 dBm at 2.4 kBaud, 1% packet error rate)
  - Low current consumption (13.3 mA in RX, 250 kBaud, input well above sensitivity limit)
  - Programmable output power up to +1 dBm
  - Excellent receiver selectivity and blocking performance
  - Programmable data rate from 1.2 to 500 kBaud
  - Frequency range: 2400 - 2483.5 MHz

- **Technical Specification**

  - Baud rates (9600)
  - Works on ISM band (2.4 GHz)
  - Designed to be as easy to use as cables
  - No external Antenna required.
  - Plug and play device.
  - Works on 5 DC supply.
  - Range 20 Mtrs
  - Both TTL and RS232 Outputs

5.3 RELAY 12V-DPDT

Fig 5.4 Relay[6]

A relay is an electrically operated switch used to isolate one electrical circuit from another. In its simplest form, a relay consists of a coil used as an electromagnet to open and close switches contacts. Since the two circuits are isolated from one another, a lower voltage circuit can be used to trip a relay, which will control a separate circuit that requires a higher voltage or amperage. Relays can be found in early telephone exchange equipment, in industrial control circuits, in car audio systems, in automobiles, on water pumps, in high-power audio amplifiers and as protection devices.

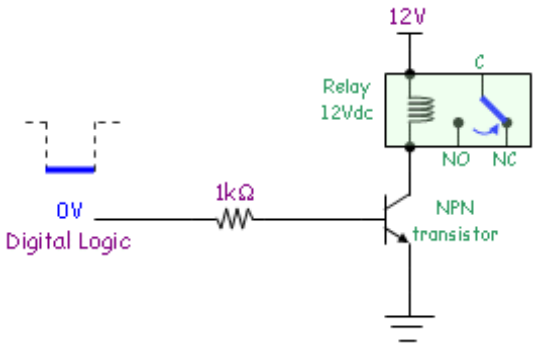


Fig 5.5 Relay circuit[6]

5.3.1 12V DC 1A DPDT Miniature Relay

High sensitivity, low power consumption BT type 47 equivalent relay, which is designed for direct PCB mounting & can also be mounted in a standard 16 pin DIL socket.

Table 5.1: Relay Specifications [6]

Maximum current	1A
Maximum Voltage	120Vac, 24Vdc
Maximum switching power	120VA resistive, 24W resistive
Contact resistance	<50mΩ
Operating time	4ms max.
Release time	2ms max.
Insulation resistance	500MΩ at 500V
Mechanical life	>10 million operations
Electrical life	>200,000 operations at full load
Contact material	Gold overlay silver alloy
Coil Details	
Nominal voltage	Coil resistance

5Vdc	180Ω±10%
12Vdc	720Ω±10%
Size of base	21.0 x 10.0mm

5.4 VOLTAGE REGULATOR

7805 is a voltage regulator integrated circuit. It is a member of 78xx series of fixed linear voltage regulator ICs. The voltage source in a circuit may have fluctuations and would not give the fixed voltage output. The voltage regulator IC maintains the output voltage at a constant value. The xx in 78xx indicates the fixed output voltage it is designed to provide. 7805 provides +5V regulated power supply. Capacitors of suitable values can be connected at input and output pins depending upon the respective voltage levels.

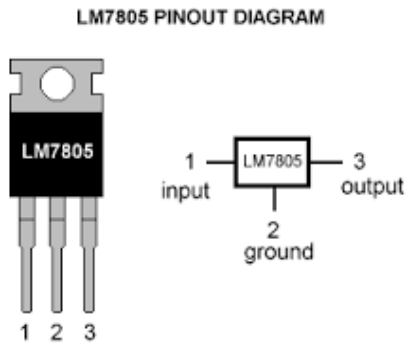


Fig 5.6: LM7805 Pinout [6]

- **Features**
  - 3-Terminal Regulators
  - Output Current up to 1.5
  - A Internal Thermal-Overload Protection

- High Power-Dissipation Capability
- Internal Short-Circuit Current Limiting
- Output Transistor Safe-Area Compensation

➢ **Technical Specification**

- Input voltage range 7V- 35V
- Current rating I c =1A
- Output voltage range V max =5.2, V min =4.8

5.5 12V POWER ADAPTER

An AC adapter, AC/DC adapter, or AC/DC converter is a type of external power supply, often enclosed in a case similar to an AC plug. AC adapters are used with electrical devices that require power but do not contain internal components to derive the required voltage and power from mains power. The internal circuitry of an external power supply is very similar to the design that would be used for a built-in or internal supply.

Fig 5.7: 12V Adapter[6]

Originally, most AC/DC adapters were linear power supplies, containing a transformer to convert the mains electricity voltage to a lower voltage,

a rectifier to convert it to pulsating DC, and a filter to smooth the pulsating waveform to DC, with residual ripple variations small enough to leave the powered device unaffected. Size and weight of the device was largely determined by the transformer, which in turn was determined by the power output and mains frequency. Ratings over a few watts made the devices too large and heavy to be physically supported by a wall outlet. The output voltage of these adapters varied with load; for equipment requiring a more stable voltage, linear voltage regulator circuitry was added. Losses in the transformer and the linear regulator were considerable; efficiency was relatively low, and significant power dissipated as heat even when not driving a load.

In the early twenty-first century, switched-mode power supplies (SMPSs) became almost ubiquitous for this purpose. Mains voltage is rectified to a high direct voltage driving a switching circuit, which contains a transformer operating at a high frequency and outputs direct current at the desired voltage. The high-frequency ripple is more easily filtered out than mains-frequency. The high frequency allows the transformer to be small, which reduces its losses; and the switching regulator can be much more efficient than a linear regulator. The result is a much more efficient, smaller, and lighter device. Safety is ensured, as in the older linear circuit, because there is still a transformer which electrically isolates the output from the mains.

A linear circuit must be designed for a specific, narrow range of input voltages (e.g., 220–240VAC) and must use a transformer appropriate for the frequency (usually 50 or 60 Hz), but a switched-mode supply can work efficiently over a very wide range of voltages and frequencies; a single 100–240VAC unit will handle almost any mains supply in the world.

➢ **Features**

- Input - Ac Input 100-240V 50 Hz
- Output voltage- 12Volt
- Output current 1Amp

5.6 CAMERA

A web camera is used to capture the gestures made by the user. And these gestures are then sent for processing. The camera used in this project is iball CHD20.0

Fig 5.8: Camera [6]

➢ Features

- 720P High-Definition Webcam with exceptional image quality
- Interpolated 20M pixels still image & 2.1M pixels video resolution
- HD 720p (1280 x 720) Widescreen resolution
- Clear and richer picture with 5G Wide angle lens
- 6 LEDs for night vision, with brightness controller.
- Built-in high sensitive USB microphone
- Snapshot button for still image capture.

- Multi-utility camera base (can be used on Desktop, Laptop & LCD)
- Auto face tracking & Digital zoom.
- Driverless – Just plug-n-play

5.7 POWER SUPPLY

It consists of step down transformer, bridge rectifier, capacitors and voltage regulator ICs. 230V AC is converted to 12V DC using transformer and bridge rectifier. This 12V DC is further reduced to 5V DC using voltage regulator IC.

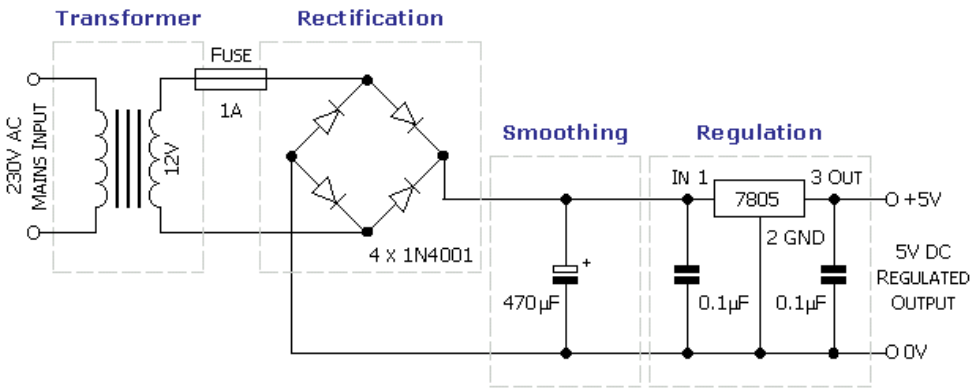


Figure 5.9: Power Supply [6]

5.8 MAX232

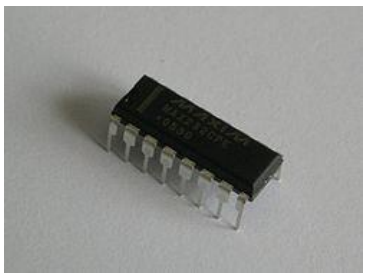


Fig 5.10: MAX232 IC [6]

The MAX232 is an IC, first created in 1987 by Maxim Integrated Products, that converts signals from an RS-232 serial port to signals suitable for use in TTL compatible digital logic circuits. The MAX232 is a dual driver/receiver and typically converts the RX, TX, CTS and RTS signals. The MAX232 (A) has two receivers (converts from RS-232 to TTL voltage levels), and two drivers (converts from TTL logic to RS-232 voltage levels). This means only two of the RS-232 signals can be converted in each direction. Typically, a pair of a driver/receiver of the MAX232 is used for TX and RX signals, and the second one for CTS and RTS signals.

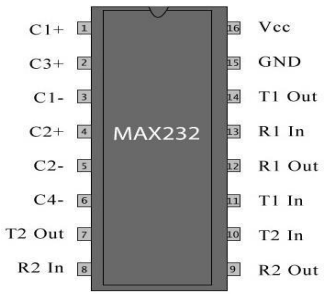


Fig 5.11: MAX232 Pin diagram[6]

It is helpful to understand what occurs to the voltage levels. When a MAX232 IC receives a TTL level to convert, it changes TTL logic 0 to between +3 and +15 V, and changes TTL logic 1 to between -3 to -15 V, and vice versa for converting from RS232 to TTL. This can be confusing when you realize that the RS232 data transmission voltages at a certain logic state are opposite from the RS232 control line voltages at the same logic state

CHAPTER 6

SOFTWARE DESCRIPTION

6.1 EMBEDDED SYSTEM

An embedded system is a special purpose computer system, which is completely encapsulated by the device it controls. It is called "embedded" because the microcontroller is inside some other systems. An embedded system has specific requirements and performs predefined tasks unlike a general purpose personal computer. An embedded system is a combination of hardware and software and perhaps additional mechanical or other parts, designed to perform a dedicated function.

- **Features of embedded system**

- It is a combination of hardware and software
- It is a system that has a computed device embedded into it.
- They are designed around a microcontroller which integrates memory and peripherals

- **Characteristics and benefits of embedded system**

- Sophisticated functionality
- Real time operation
- Low manufacturing cost
- Low power consumption

- Eliminates necessity of complex circuitry
- Smarter products
- Smaller size
- User friendly
- State of the art technology

6.2 MPLAB IDE

MPLAB® X IDE is a software program that runs on a PC (Windows®, Mac OS®, Linux®) to develop applications for Microchip microcontrollers and digital signal controllers. It is called an Integrated Development Environment (IDE), because it provides a single integrated "environment" to develop code for embedded microcontrollers.

MPLAB® X Integrated Development Environment brings many changes to the PIC® microcontroller development tool chain. Unlike previous versions of the MPLAB® IDE which were developed completely in-house, MPLAB® X IDE is based on the open source NetBeans IDE from Oracle. Taking this path has allowed us to add many frequently requested features very quickly and easily, while also providing us with a much more extensible architecture to bring you even more new features in the future.

MPLAB X is the first version of the IDE to include cross-platform support for Mac OS X and Linux operating systems, in addition to Microsoft Windows.

MPLAB X supports the following compilers:

- MPLAB XC8 — C compiler for 8-bit PIC devices
- MPLAB XC16 — C compiler for 16-bit PIC devices
- MPLAB XC32 — C++ compiler for 32-bit PIC devices
- HI-TECH C — C compiler for 8-bit PIC devices
- SDCC — open-source C compiler

6.2.1 Components of MPLAB IDE

The MPLAB IDE has both built-in components and plug-in modules to configure the system for a variety of software and hardware tools. 1.6.1 MPLAB IDE Built-In Components The built-in components consist of:

• Project Manager- The project manager provides integration and communication between the IDE and the language tools.

• Editor- The editor is a full-featured programmer's text editor that also serves as a window into the debugger.

• Assembler/Linker and Language Tools -The assembler can be used standalone to assemble a single file, or can be used with the linker to build a project from separate source files, libraries and recompiled objects. The linker is responsible for positioning the compiled code into memory areas of the target microcontroller.

• Debugger -The Microchip debugger allows breakpoints, single-stepping, watch windows and all the features of a modern debugger for the MPLAB IDE. It works in conjunction with the editor to reference information from the target being debugged back to the source code

. • Execution Engines- There are software simulators in MPLAB IDE for all PICmicro and dsPIC devices. These simulators use the PC to simulate the instructions and some peripheral functions of the PICmicro and dsPIC devices.

Optional in-circuit emulators and in-circuit debuggers are also available to test code as it runs in the applications hardware.

What is MPLAB® IDE? □ 2005 Microchip Technology Inc. DS51519A-page 19 1.6.2 Additional Optional Components for MPLAB IDE Optional components can be purchased and added to the MPLAB IDE:

• Compiler Language Tools MPLAB C17, MPLAB C18 and MPLAB C30 from Microchip provide fully integrated, optimized code. Along with compilers from HI-TECH, IAR, microEngineering Labs, CCS and Byte Craft, they are invoked by the MPLAB IDE project manager to compile code that is automatically loaded into the target debugger for instant testing and verification.

• Programmers PICSTART Plus, PRO MATE II, MPLAB PM3 as well as MPLAB ICD 2 can program code into target microcontrollers. MPLAB IDE offers full control over programming both code and data, as well as the configuration bits to set the various operating modes of the target microcontrollers

• In-Circuit Emulators MPLAB ICE 2000 and MPLAB ICE 4000 are full-featured emulators for the PICmicro and dsPIC devices. They connect to the PC via I/O ports and allow full control over the operation of microcontroller in the target applications.

• In-Circuit Debugger MPLAB ICD 2 provides an economic alternative to an emulator. By using some of the on-chip resources, MPLAB ICD 2 can download code into a target microcontroller inserted in the application, set

6.3 HITECH C COMPILER

The HI-TECH C Compiler for PIC10/12/16 MCUs is a free-standing, optimizing ANSI C compiler. It supports all PIC10, PIC12 and PIC16 series

devices, as well as the PIC14000 device and the enhanced Mid-Range PIC® MCU architecture. The compiler is available for several popular operating systems, including 32 and 64-bit Windows®, Linux and Apple OS X. The compiler can run in one of three operating modes: Lite, Standard or PRO. The Standard and PRO operating modes are licensed modes and require a serial number to enable them. Lite mode is available for unlicensed customers. The basic compiler operation, supported devices and available memory are identical across all modes. The modes only differ in the level of optimization employed by the compiler.

6.3.1 Device description

This compiler supports Microchip PIC devices with Baseline, Mid-Range and Enhanced Mid-Range cores. All are 8-bit devices. The Baseline core uses a 12-bit wide instruction set and is available in PIC10, PIC12 and PIC16 part numbers. The Mid-Range core utilizes a 14-bit wide instruction set that includes additional instructions to those provided by Baseline parts. Its data memory banks and program memory pages are larger than those on Baseline devices. It is available in PIC12, PIC14 and PIC16 part numbers. The Enhanced Mid-Range core also uses a 14-bit wide instruction set, but incorporates additional instructions and features over the Mid-Range devices. There are both PIC12 and PIC16 part numbers that are based on the Enhanced Mid-Range core.

6.4 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

- 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 noninteractive language such as C or Fortran.

The name MATLAB stands for matrix laboratory. MATLAB was originally written to provide easy access to matrix software developed by the LINPACK and EISPACK projects, which together represent the state-of-the-art in software for matrix computation.

MATLAB has evolved over a period of years with input from many users. In university environments, it is the standard instructional tool for introductory and advanced courses in mathematics, engineering, and science. In industry, MATLAB is the tool of choice for high-productivity research, development, and analysis.

MATLAB features a family of 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.

6.4.1 The MATLAB System

The MATLAB system consists of five main parts:

- The MATLAB language.

This is a high-level matrix/array language with control flow statements, functions, data structures, input/output, and object-oriented programming features. It allows both "programming in the small" to rapidly create quick and dirty throw-away programs, and "programming in the large" to create complete large and complex application programs.

- The MATLAB working environment

This is the set of tools and facilities that you work with as the MATLAB user or programmer. It includes facilities for managing the variables in your workspace and importing and exporting data. It also includes tools for developing, managing, debugging, and profiling M-files, MATLAB's applications.

- Handle Graphics

This is the MATLAB graphics system. It includes high-level commands for two-dimensional and three-dimensional data visualization, image processing, animation, and presentation graphics. It also includes low-level commands that allow you to fully customize the appearance of graphics as well as to build complete Graphical User Interfaces on your MATLAB applications.

- The MATLAB mathematical function library

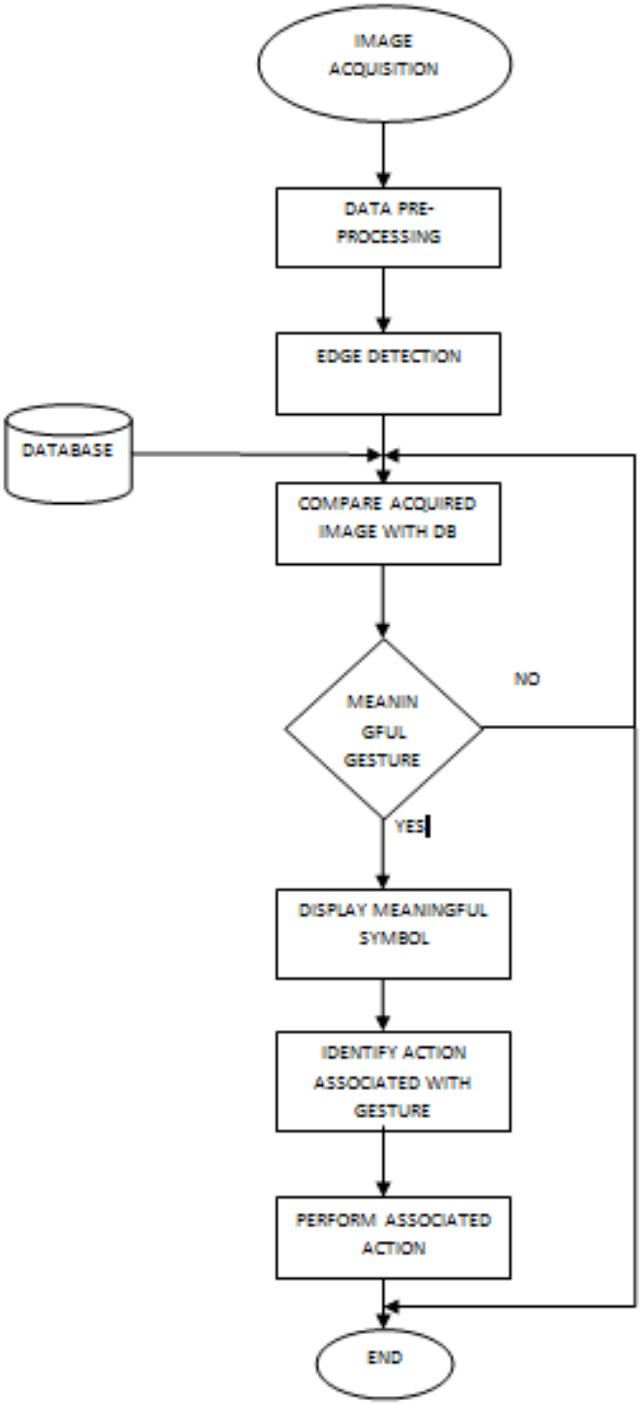
This is a vast collection of computational algorithms ranging from elementary functions like sum, sine, cosine, and complex arithmetic, to more sophisticated functions like matrix inverse, matrix eigenvalues, Bessel functions, and fast Fourier transforms.

- The MATLAB Application Program Interface (API)

This is a library that allows you to write C and Fortran programs that interact with MATLAB. It include facilities for calling routines from MATLAB (dynamic linking), calling MATLAB as a computational engine, and for reading and writing MAT-files.

CHAPTER 7

FLOWCHART



CHAPTER 8

PCB FABRICATION

8.1 PRINTED CIRCUIT BOARD

ARES (Advanced Routing and Editing Software) forms the PCB layout module of the proteus system and offers net listed based PCB design and complete with a suite of high performance design automation tools. The latest version is compatible with Windows 998/Me/2k/XP and later. It includes a brand new Auto-placer, improved auto-placing, automatic gate swap optimization and even more support for power plans.

8.2 PCB FABRICATION

The various steps involved in the preparation of PCB are explained below:

1) PCB fabrication

2) Soldering of components

1) PCB fabrication

Materials required: Copper clad sheet, paint, ferric chloride solution, drilling machine.

The steps involved in making PCB:

a) **Preparation of lay out of the track**: The track lay out of the circuit is drawn using PCB designing software. The layout should be made in such a way that the parts are in easy roots which will be done automatically by the

use of software. This enables PCB to be more economical and compact. The softwares commonly used are ORCAD, Proteus, PCB express etc

b) Transferring of layout to the copper clad: Plain, copper clad PC board can be purchased from many different retailers. If necessary, cut down the board to the approximate size. A table saw with a diamond blade or a shear are the best choices. In any case, be sure to wear goggles and a breathing mask. Always clean the board with water (perhaps some cleaning solution) and a scouring pad. The scrubbing removes grime and oxidation, thus preparing the surface for a crisp transfer and ultimately firm soldering. After drying the copper board, place the blue transfer paper with the image and toner side against the copper. Carefully position the image. The clinging of the plastic to the clean copper helps.

c) Etching: Ferric chloride solution is popularly used as the etching solution. The ferric chloride powder is made in to a solution using water and kept in a plastic tray. Now carefully immerse the marked copper clad sheet in the solution for a period of 1-2 hours. Due to reaction the solution becomes weak and it can't be used for further etching process. The copper in the unmarked area will be etched out. Take out the etched sheet from the tray containing the solution and wash it in clean water and dry out in sunlight for an hour. Later remove the paint over the tracks using turpentine.

d) Drilling: The holes are made in the etched out copper clad sheet using a drilling machine with drill bits of size 0.5mm, 0.6mm, 0.9mm etc are depending on the leads of the components used in the circuit.

8.3 SOLDERING OF COMPONENTS

Soldering is the process of joining two or more dissimilar metals by melting another metal having low melting point. In order to make the surfaces accept the soldering readily the component terminals should be oxides and other

obstructing films. Soldering flux cleans the oxides from the surfaces of the metal; the lead should be cleaned chemically or by scrapping using a blade or knife. Small amount of lead should be coated on the cleaned position of leads and the bits of soldering iron. This process is called tinning. Zinc chloride, ammonium chloride, rosin are called fluxes. The solder is used for joining two metals at temperature below the melting point. The popularly used alloys of tin and lead melts at 375 degree F and solidifies when it cools. Most of the Solder wires are flux core type. Soldering iron is the tool used to melt the solder and apply at the joint of the circuit. It operates at 230 v main supply. The power ratings of the soldering iron are 10W, 25W, 35W, 65W, 125W.

8.4 HOW TO SOLDER

Make a layout according to the connection of components in the circuit. Plug the soldering iron in the mains supply to get it heated up. Clean the components leads using a blade or knife and bent them according to the needs. Apply a little flux on the leads. Mount the components on the PCB, apply flux on the joints and solder the joint. Soldering must be done in the minimum time to avoid the dry heating of solder and heating up of the components. Cut the excess leads after soldering using a wire stripper or sniper.

CHAPTER 9

RESULTS AND DISCUSSION

We have implemented a total of 4 gestures for which corresponding actions are produced. A database of 26 characters is built. We have created GUI for both the database creation as well as for the character recognition.

The GUI developed are shown below:

Fig 9.1: GUI For database creation

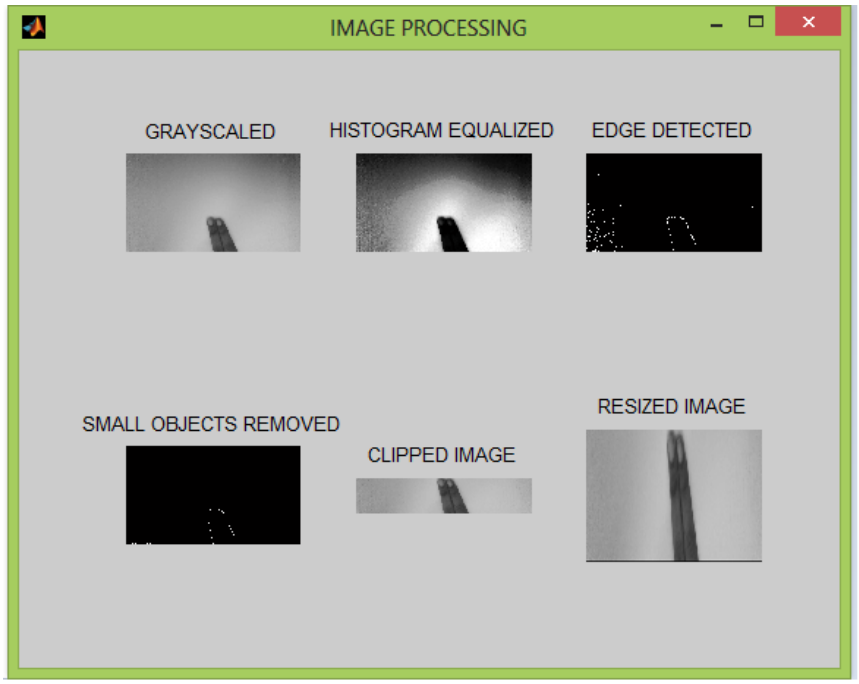


Fig 9.2: Image processing steps undergone

Fig 9.3: GUI for gesture recognition

Taking accuracy and efficiency into consideration, the proposed system is simulated using MATLAB software. System shows the hand gesture recognition technique to interact with machines. The hand gesture output is tested for various inputs. It improves the performance and makes human computer interaction more convenient.

CHAPTER 10

ADVANTAGES

A. Real time functioning.

The output of the system will be displayed in the text form in real time. This makes the system more efficient. The images captured through web camera are compared and the result of comparison is displayed at the same time. Thus this feature of the system makes it very simple and delay free.

B. Portable.

The entire system becomes portable and can be taken anywhere. This feature facilitates the user to take the system anywhere and everywhere and overcomes the barrier of restricting him/herself to communicate without a desktop or laptop.

C. Secure

Hand gesture from one user varies from another, this clearly helps in making this system secure.

D. Does not get damage through use.

As no special sensors are used in this system, the system is less likely to get damaged.

CHAPTER 11

DISADVANTAGES

There were some disadvantages that we come across this system. First of all, for a very efficient result perfect lighting was required. Any disturbance in the lighting will hinder the output. Following that, preventing shadows from falling in the view of the camera was a tedious task. The shadows had a very big impact on the output as they were labelled along with other components creating confusion for the algorithm in determining which the required and non-required components are. In another case, if we had used concepts like Haar wavelet transform with Hue, Saturation values to extract skin tone and other features it would make the design more complex in MATLAB. Efficiency would be less as the results are more likely to be distorted due to improper lighting.

CHAPTER 12

APPLICATIONS

The project that has been introduced here can be used for variety of applications

- Aid to physically challenged
- Immersive game technology
- Automatic translation of gesture based natural languages
- Remote and virtual controllers

CHAPTER 13

FUTURE SCOPE

- Developing the codes in programming languages like OpenCV will have better performance and efficiency.
- Another advancement that can be made is getting the inputs dynamically i.e. setting a perfect interval for each input gesture to get recognized, comparing and determining the output thereby making the system fully automated.
- By integrating our system with voice recognition system to embedded it in robots

CHAPTER 14

CONCLUSION

Gestures shown by the user were successfully interpreted using adaptive method for hand gesture recognition. The main advantage of using this method was it involved no complications like the Hue, saturation, values and skin tone detection that are most likely to vary due to lighting conditions. Therefore, for a simple web camera this method produced sufficient results and is capable of producing better results with high resolutions cameras as well.

MATLAB has proved itself to be an efficient application. MATLAB was employed to perform the simulation of all the results. It has a lot of built-in functions that helps lessen the amount of time spent on writing the simulation code.

REFERENCES

[1] Namita Agarwal, S.M.Hambarde, "Static hand gesture's voice conversion system using Vision based approach for mute people." Proceedings of 26th IRF international conference, 10th may 2015

[2] Rajat Garg, N.Shriram, Vikrant Gupta, and Vineet Agrawal "A Biometric Security Based Electronic Gadget Control Using Hand Gestures", International Journal of Electronics and Computer Science Engineering, ISSN 2277-1956/V1N3-1103-1107

[3] Meenakshi Panwar and Pawan Singh Mehra , "Hand Gesture Recognition for Human Computer Interaction", in Proceedings of IEEE International Conference on Image Information Processing(ICIIP 2011), Waknaghat, India, November 2011

[4] Ms. Rashmi D. Kyatanavar, Prof. P. R. Futane, Comparative Study ofSign Language Recognition Systems, International Journal of Scientificand Research Publications, Volume 2, Issue 6, June 2012

[5] J. Singha and K. Das, "Indian Sign Language Recognition Using Eigen Value Weighted Euclidean Distance Based Classification Technique",(IJACSA) International Journal of Advanced Computer Science and Applications, Volume 4, no. 2 , pp. 188-195,July 2013

[6] https://www.google.co.in

APPENDIX: PROGRAM

CODE FOR DATABASE CREATION

```
function varargout = database_craete(varargin)
% DATABASE_CRAETE MATLAB code for database_craete.fig
%      DATABASE_CRAETE, by itself, creates a new DATABASE_CRAETE or raises the existing
%      singleton*.
%
%      H = DATABASE_CRAETE returns the handle to a new DATABASE_CRAETE or the handle to
%      the existing singleton*.
%
%      DATABASE_CRAETE('CALLBACK',hObject,eventData,handles,...) calls the local
%      function named CALLBACK in DATABASE_CRAETE.M with the given input arguments.
%
%      DATABASE_CRAETE('Property','Value',...) creates a new DATABASE_CRAETE or raises the
%      existing singleton*.  Starting from the left, property value pairs are
%      applied to the GUI before database_craete_OpeningFcn gets called.  An
%      unrecognized property name or invalid value makes property application
%      stop.  All inputs are passed to database_craete_OpeningFcn via varargin.
%
%      *See GUI Options on GUIDE's Tools menu.  Choose "GUI allows only one
%      instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES
```

```
% Edit the above text to modify the response to help database_craete

% Last Modified by GUIDE v2.5 02-Mar-2017 12:31:13

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @database_craete_OpeningFcn, ...
                   'gui_OutputFcn',  @database_craete_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT



% --- Executes just before database_craete is made visible.
function database_craete_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB

% handles    structure with handles and user data (see GUIDATA)
```

```
% varargin   command line arguments to database_craete (see VARARGIN)

% Choose default command line output for database_craete
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes database_craete wait for user response (see UIRESUME)
% uiwait(handles.figure1);
global vid h
clc
imaqreset;
pause(0.5);
% vid = videoinput('winvideo', 1, 'RGB24_320x240');
% vid = videoinput('winvideo', 1, 'RGB24_640x480');
vid = videoinput('winvideo', 1, 'MJPG_640x360');

src = getselectedsource(vid);
vid.FramesPerTrigger = 1;
h= imshow(zeros(480,640),'Parent',handles.axes1);
set(handles.edit1,'String','');



% --- Outputs from this function are returned to the command line.
function varargout = database_craete_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
```

```
varargout{1} = handles.output;
 function edit1_Callback(hObject, eventdata, handles)
% hObject    handle to edit1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit1 as text
%        str2double(get(hObject,'String')) returns contents of edit1 as a double


% --- Executes during object creation, after setting all properties.
function edit1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called
% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
   set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

global vid h
gesturename=get(handles.edit1,'string');
```

```
if isempty(gesturename)
    a=msgbox('PLEASE SPECIFY NAME OF GESTURE');
    uiwait(a);
else
    disp('place your hand in place and press enter')
    preview(vid,h);
    pause
    image_in = getsnapshot(vid);
    if size(image_in,3)==3
        image_in=rgb2gray(image_in);
    end
    image_hist=histeq(image_in);
    image_edge=edge(image_hist,'sobel');
    image_edge_thresh=bwareaopen(image_edge,30);
    [row,col]=find(image_edge_thresh);
    template_img=image_in(min(row):max(row),min(col):max(col));
    template_img_resized=imresize(template_img,[240,320]);
    path = pwd;
    cd  database
    imwrite(template_img_resized,[gesturename '.jpg']);
    pause(0.02);
    cd ..
    f=figure('name','IMAGE PROCESSING','menubar','none','numbertitle','off');
    subplot(2,3,1)
    imshow(image_in);
    title('GRAYSCALED')

    subplot(2,3,2)
    imshow(image_hist);
    title('HISTOGRAM EQUALIZED')
```

```
    subplot(2,3,3)
    imshow(image_edge);
    title('EDGE DETECTED')

    subplot(2,3,4)
    imshow(image_edge_thresh);
    title('SMALL OBJECTS REMOVED')

    subplot(2,3,5)
    imshow(template_img);
    title('CLIPPED IMAGE')

    subplot(2,3,6)
    imshow(template_img_resized);
    title('RESIZED IMAGE')
    set(handles.edit1,'String','');
    stoppreview(vid);
    h=imshow(zeros(480,640),'Parent',handles.axes1);
    clc;
end
```

CODE FOR MAIN GUI

```
function varargout = main_gui(varargin)
% MAIN_GUI MATLAB code for main_gui.fig
%      MAIN_GUI, by itself, creates a new MAIN_GUI or raises the existing
%      singleton*.
%
%      H = MAIN_GUI returns the handle to a new MAIN_GUI or the handle to
%      the existing singleton*.
%
%      MAIN_GUI('CALLBACK',hObject,eventData,handles,...) calls the local
%        function named CALLBACK in MAIN_GUI.M with the given input arguments.
%
%      MAIN_GUI('Property','Value',...) creates a new MAIN_GUI or raises the
%      existing singleton*.  Starting from the left, property value pairs are
%      applied to the GUI before main_gui_OpeningFcn gets called.  An
%      unrecognized property name or invalid value makes property application
%      stop.  All inputs are passed to main_gui_OpeningFcn via varargin.
%
%       *See GUI Options on GUIDE's Tools menu.  Choose "GUI allows only one
%      instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES
% Edit the above text to modify the response to help main_gui

% Last Modified by GUIDE v2.5 12-Mar-2017 06:30:35

% Begin initialization code - DO NOT EDIT

gui_Singleton = 1;
```

```
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @main_gui_OpeningFcn, ...
                   'gui_OutputFcn',  @main_gui_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT



% --- Executes just before main_gui is made visible.
function main_gui_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to main_gui (see VARARGIN)

% Choose default command line output for main_gui
handles.output = hObject;

% Update handles structure

guidata(hObject, handles);
```

```
% UIWAIT makes main_gui wait for user response (see UIRESUME)
% uiwait(handles.figure1);
global vid h ser
clc
imaqreset;
instrreset;
 ser = serial('COM4');
 fopen(ser);
 pause(0.5);
% vid = videoinput('winvideo', 1, 'RGB24_320x240');
% vid = videoinput('winvideo', 1, 'RGB24_640x480');
% vid = videoinput('winvideo', 2, 'MJPG_640x360');
vid = videoinput('winvideo', 1, 'MJPG_640x360');

src = getselectedsource(vid);
vid.FramesPerTrigger = 1;
h= imshow(zeros(480,640),'Parent',handles.axes1);
set(handles.pushbutton1,'string','START');
set(handles.text4,'Visible','off');
set(handles.text3,'Visible','off');

% --- Outputs from this function are returned to the command line.
function varargout = main_gui_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
 % Get default command line output from handles structure
varargout{1} = handles.output;

 % --- Executes on button press in pushbutton1.
```

```
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

global vid h ser
action = get(handles.pushbutton1,'string');

if strcmp(action,'START')
   set(handles.pushbutton1,'string','STOP');
   disp('place your hand in place and press enter')
   preview(vid,h);
   pause
   image_in = getsnapshot(vid);
   if size(image_in,3)==3
      image_in=rgb2gray(image_in);
   end
   image_hist=histeq(image_in);
   image_edge=edge(image_hist,'sobel');
   image_edge_thresh=bwareaopen(image_edge,30);
   [row,col]=find(image_edge_thresh);
   template_img=image_in(min(row):max(row),min(col):max(col));
   template_img_resized=imresize(template_img,[240,320]);

   keys = [] ;
   cd database
   a = dir('*.jpg');
   sz = size(dir('*.jpg'),1)
   for lp = 1:sz
      temp = imread(a(lp).name);

      key = corr2(template_img_resized,temp);
```

```
        keys = [keys key ];
    end
    keys
    mx = max(keys);
     if(mx > .68)
      indx = find(keys==mx)
      letter = a(indx).name;
      letter = letter(1:end-4)  % letter is the recognised alphabet after
      set(handles.text4,'Visible','on');
      set(handles.text3,'Visible','on');
      set(handles.text3,'String',letter);
      fprintf(ser,letter);
     else
        set(handles.text4,'Visible','on');
         set(handles.text3,'Visible','on');
         set(handles.text3,'String','No match');

     end
    cd ..                % the matching the gest with curnt database


else
     set(handles.pushbutton1,'string','START');
     stoppreview(vid);
     set(handles.text4,'Visible','off');
     set(handles.text3,'Visible','off');
     h=imshow(zeros(480,640),'Parent',handles.axes1);
     clc;

end

function edit1_Callback(hObject, eventdata, handles)
```

```
% hObject    handle to edit1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit1 as text
%        str2double(get(hObject,'String')) returns contents of edit1 as a double


% --- Executes during object creation, after setting all properties.
function edit1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
   set(hObject,'BackgroundColor','white');
end
```

CODE FOR RECEIVER SECTION

```
#include<pic.h>
#include"uart.h"
#include"delay.h"

void main()
{
        char a;
        TRISE=0X00;
        ADCON1=0X07;
        TRISC=0X80;
        UART_init();
        while(1)
        {
                a=UART_receive();
                if(a=='A')
                        RE0=1;
                else if(a=='B')
                        RE0=0;
                if(a=='C')
                        RE1=1;
                else if(a=='D')
                        RE1=0;
        }


}
```

APPENDIX: DATASHEETS

PIC16F877A

MICROCHIP

PIC16F87XA

28/40/44-Pin Enhanced Flash Microcontrollers

Devices Included in this Data Sheet:

- PIC16F873A
- PIC16F874A
- PIC16F876A
- PIC16F877A

High-Performance RISC CPU:

- Only 35 single-word instructions to learn
- All single-cycle instructions except for program branches, which are two-cycle
- Operating speed: DC – 20 MHz clock input
  DC – 200 ns instruction cycle
- Up to 8K x 14 words of Flash Program Memory,
  Up to 368 x 8 bytes of Data Memory (RAM),
  Up to 256 x 8 bytes of EEPROM Data Memory
- Pinout compatible to other 28-pin or 40/44-pin PIC16CXXX and PIC16FXXX microcontrollers

Peripheral Features:

- Timer0: 8-bit timer/counter with 8-bit prescaler
- Timer1: 16-bit timer/counter with prescaler, can be incremented during Sleep via external crystal/clock
- Timer2: 8-bit timer/counter with 8-bit period register, prescaler and postscaler
- Two Capture, Compare, PWM modules
  - Capture is 16-bit, max. resolution is 12.5 ns
  - Compare is 16-bit, max. resolution is 200 ns
  - PWM max. resolution is 10-bit
- Synchronous Serial Port (SSP) with SPI™ (Master mode) and I^2C™ (Master/Slave)
- Universal Synchronous Asynchronous Receiver Transmitter (USART/SCI) with 9-bit address detection
- Parallel Slave Port (PSP) – 8 bits wide with external $\overline{RD}$, $\overline{WR}$ and $\overline{CS}$ controls (40/44-pin only)
- Brown-out detection circuitry for Brown-out Reset (BOR)

Analog Features:

- 10-bit, up to 8-channel Analog-to-Digital Converter (A/D)
- Brown-out Reset (BOR)
- Analog Comparator module with:
  - Two analog comparators
  - Programmable on-chip voltage reference (V_{REF}) module
  - Programmable input multiplexing from device inputs and internal voltage reference
  - Comparator outputs are externally accessible

Special Microcontroller Features:

- 100,000 erase/write cycle Enhanced Flash program memory typical
- 1,000,000 erase/write cycle Data EEPROM memory typical
- Data EEPROM Retention > 40 years
- Self-reprogrammable under software control
- In-Circuit Serial Programming™ (ICSP™) via two pins
- Single-supply 5V In-Circuit Serial Programming
- Watchdog Timer (WDT) with its own on-chip RC oscillator for reliable operation
- Programmable code protection
- Power saving Sleep mode
- Selectable oscillator options
- In-Circuit Debug (ICD) via two pins

CMOS Technology:

- Low-power, high-speed Flash/EEPROM technology
- Fully static design
- Wide operating voltage range (2.0V to 5.5V)
- Commercial and Industrial temperature ranges
- Low-power consumption

Device	Program Memory		Data SRAM (Bytes)	EEPROM (Bytes)	I/O	10-bit A/D (ch)	CCP (PWM)	MSSP		USART	Timers 8/16-bit	Comparators
	Bytes	# Single Word Instructions						SPI	Master I^2C			
PIC16F873A	7.2K	4096	192	128	22	5	2	Yes	Yes	Yes	2/1	2
PIC16F874A	7.2K	4096	192	128	33	8	2	Yes	Yes	Yes	2/1	2
PIC16F876A	14.3K	8192	368	256	22	5	2	Yes	Yes	Yes	2/1	2
PIC16F877A	14.3K	8192	368	256	33	8	2	Yes	Yes	Yes	2/1	2

PIC16F87XA

1.0 DEVICE OVERVIEW

This document contains device specific information about the following devices:

- PIC16F873A
- PIC16F874A
- PIC16F876A
- PIC16F877A

PIC16F873A/876A devices are available only in 28-pin packages, while PIC16F874A/877A devices are available in 40-pin and 44-pin packages. All devices in the PIC16F87XA family share common architecture with the following differences:

- The PIC16F873A and PIC16F874A have one-half of the total on-chip memory of the PIC16F876A and PIC16F877A
- The 28-pin devices have three I/O ports, while the 40/44-pin devices have five
- The 28-pin devices have fourteen interrupts, while the 40/44-pin devices have fifteen
- The 28-pin devices have five A/D input channels, while the 40/44-pin devices have eight
- The Parallel Slave Port is implemented only on the 40/44-pin devices

The available features are summarized in Table 1-1. Block diagrams of the PIC16F873A/876A and PIC16F874A/877A devices are provided in Figure 1-1 and Figure 1-2, respectively. The pinouts for these device families are listed in Table 1-2 and Table 1-3.

Additional information may be found in the PICmicro® Mid-Range Reference Manual (DS33023), which may be obtained from your local Microchip Sales Representative or downloaded from the Microchip web site. The Reference Manual should be considered a complementary document to this data sheet and is highly recommended reading for a better understanding of the device architecture and operation of the peripheral modules.

TABLE 1-1: PIC16F87XA DEVICE FEATURES

Key Features	PIC16F873A	PIC16F874A	PIC16F876A	PIC16F877A
Operating Frequency	DC – 20 MHz	DC – 20 MHz	DC – 20 MHz	DC – 20 MHz
Resets (and Delays)	POR, BOR (PWRT, OST)	POR, BOR (PWRT, OST)	POR, BOR (PWRT, OST)	POR, BOR (PWRT, OST)
Flash Program Memory (14-bit words)	4K	4K	8K	8K
Data Memory (bytes)	192	192	368	368
EEPROM Data Memory (bytes)	128	128	256	256
Interrupts	14	15	14	15
I/O Ports	Ports A, B, C	Ports A, B, C, D, E	Ports A, B, C	Ports A, B, C, D, E
Timers	3	3	3	3
Capture/Compare/PWM modules	2	2	2	2
Serial Communications	MSSP, USART	MSSP, USART	MSSP, USART	MSSP, USART
Parallel Communications	—	PSP	—	PSP
10-bit Analog-to-Digital Module	5 Input channels	8 Input channels	5 Input channels	8 Input channels
Analog Comparators	2	2	2	2
Instruction Set	35 Instructions	35 Instructions	35 Instructions	35 Instructions
Packages	28-pin PDIP 28-pin SOIC 28-pin SSOP 28-pin QFN	40-pin PDIP 44-pin PLCC 44-pin TQFP 44-pin QFN	28-pin PDIP 28-pin SOIC 28-pin SSOP 28-pin QFN	40-pin PDIP 44-pin PLCC 44-pin TQFP 44-pin QFN

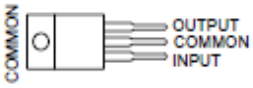
VOLTAGE REGULATOR

µA7800 SERIES
POSITIVE-VOLTAGE REGULATORS

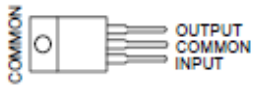
SLVS056J – MAY 1976 – REVISED MAY 2003

- 3-Terminal Regulators
- Output Current up to 1.5 A
- Internal Thermal-Overload Protection
- High Power-Dissipation Capability
- Internal Short-Circuit Current Limiting
- Output Transistor Safe-Area Compensation

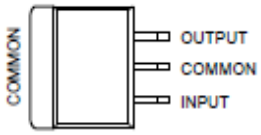
KC (TO-220) PACKAGE
(TOP VIEW)



KCS (TO-220) PACKAGE
(TOP VIEW)



KTE PACKAGE
(TOP VIEW)



description/ordering information

This series of fixed-voltage integrated-circuit voltage regulators is designed for a wide range of applications. These applications include on-card regulation for elimination of noise and distribution problems associated with single-point regulation. Each of these regulators can deliver up to 1.5 A of output current. The internal current-limiting and thermal-shutdown features of these regulators essentially make them immune to overload. In addition to use as fixed-voltage regulators, these devices can be used with external components to obtain adjustable output voltages and currents, and also can be used as the power-pass element in precision regulators.

ORDERING INFORMATION

T_J	$V_{O(NOM)}$ (V)	PACKAGE†		ORDERABLE PART NUMBER	TOP-SIDE MARKING
0°C to 125°C	5	POWER-FLEX (KTE)	Reel of 2000	µA7805CKTER	µA7805C
		TO-220 (KC)	Tube of 50	µA7805CKC	µA7805C
		TO-220, short shoulder (KCS)	Tube of 20	µA7805CKCS	
	8	POWER-FLEX (KTE)	Reel of 2000	µA7808CKTER	µA7808C
		TO-220 (KC)	Tube of 50	µA7808CKC	µA7808C
		TO-220, short shoulder (KCS)	Tube of 20	µA7808CKCS	
	10	POWER-FLEX (KTE)	Reel of 2000	µA7810CKTER	µA7810C
		TO-220 (KC)	Tube of 50	µA7810CKC	µA7810C
	12	POWER-FLEX (KTE)	Reel of 2000	µA7812CKTER	µA7812C
		TO-220 (KC)	Tube of 50	µA7812CKC	µA7812C
		TO-220, short shoulder (KCS)	Tube of 20	µA7812CKCS	
	15	POWER-FLEX (KTE)	Reel of 2000	µA7815CKTER	µA7815C
		TO-220 (KC)	Tube of 50	µA7815CKC	µA7815C
		TO-220, short shoulder (KCS)	Tube of 20	µA7815CKCS	
	24	POWER-FLEX (KTE)	Reel of 2000	µA7824CKTER	µA7824C
		TO-220 (KC)	Tube of 50	µA7824CKC	µA7824C

† Package drawings, standard packing quantities, thermal data, symbolization, and PCB design guidelines are available at www.ti.com/sc/package.

MAX232

TEXAS INSTRUMENTS

MAX232, MAX232I

SLLS047M – FEBRUARY 1989 – REVISED NOVEMBER 2014

MAX232x Dual EIA-232 Drivers/Receivers

1 Features

- Meets or Exceeds TIA/EIA-232-F and ITU Recommendation V.28
- Operates From a Single 5-V Power Supply With 1.0-µF Charge-Pump Capacitors
- Operates up to 120 kbit/s
- Two Drivers and Two Receivers
- ±30-V Input Levels
- Low Supply Current: 8 mA Typical
- ESD Protection Exceeds JESD 22
  - 2000-V Human-Body Model (A114-A)
- Upgrade With Improved ESD (15-kV HBM) and 0.1-µF Charge-Pump Capacitors is Available With the MAX202 Device

2 Applications

- TIA/EIA-232-F
- Battery-Powered Systems
- Terminals
- Modems
- Computers

3 Description

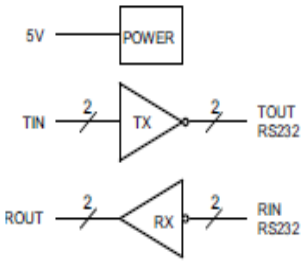
The MAX232 device is a dual driver/receiver that includes a capacitive voltage generator to supply TIA/EIA-232-F voltage levels from a single 5-V supply. Each receiver converts TIA/EIA-232-F inputs to 5-V TTL/CMOS levels. These receivers have a typical threshold of 1.3 V, a typical hysteresis of 0.5 V, and can accept ±30-V inputs. Each driver converts TTL/CMOS input levels into TIA/EIA-232-F levels.

Device Information(1)

ORDER NUMBER	PACKAGE (PIN)	BODY SIZE
MAX232x	SOIC (16)	9.90 mm × 3.91 mm
	SOIC (16)	10.30 mm × 7.50 mm
	PDIP (16)	19.30 mm × 6.35 mm
	SOP (16)	10.3 mm × 5.30 mm

(1) For all available packages, see the orderable addendum at the end of the datasheet.

4 Simplified Schematic



IBALL CHD20.0

- 720P High-Definition Webcam with exceptional image quality
- Interpolated 20M pixels still image & 2.1M pixels video resolution
- Video calling & High Quality still picture

MRP : Rs. 1,499.00

Warranty : 2 Years

Features / Specifications

Features :	HD 720p (1280 x 720) Widescreen resolution Clear and richer picture with 5G Wide angle lens 6 LEDs for night vision, with brightness controller. Built-in high sensitive USB microphone Snapshot button for still image capture. Multi-utility camera base (can be used on Desktop, Laptop & LCD) Auto face tracking & Digital zoom. Driverless – Just plug-n-play (However, to get best results use with provided drivers) High-speed USB 2.0 interface. Free tripod stand bundled
Max. Video Resolution :	1920 x1080 pixels
Max. Image Resolution :	5500 x 3640 pixels
Frame Rates :	30 Frames per second
Interface :	USB 2.0. Backward compatible with USB 1.1
Focus :	Manual Focus
Microphone :	Built-in USB Mic
OS Compatibility :	Windows XP, Vista, 7, 8. (Mac and Linux with limited functions)
Sensor resolution :	1 MP CMOS
Interpolated Resolution :	20 MP (Photo), 2.1 MP (Video)
Lens :	5G wide angle lens
Cable Length :	1.5m
LED's :	6 LEDs for night vision, with brightness controller
Software Features :	Camera driver