ECLECTIC AUTOMATION GUIDED LITHERING EQUIPAGE (E.A.G.L.E)

PROJECT REPORT

Submitted by

S.SABARISAN	**- 2KEEE63**
M.VIJAYAKUMAR	**- 2KEEE67**
S.VINOD	**- 2KEEE58**
V.K.PRASANTH	**- 2KEEE30**

Guided by

Mrs.R.Mahalakhsmi, M.E,
Senior Lecturer, EEE Department.

In partial fulfillment of the requirements for the Award of the degree of BACHELOR OF ENGINEERING in ELECTRICAL AND ELECTRONICS ENGINEERING Branch of BHARATHIAR UNIVERSITY, COIMBATORE

DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING
KUMARAGURU COLLEGE OF TECHNOLOGY
COIMBATORE - 641 006

Estd-1984

ISO 9001:2000 Certified

2003- 2004

Estd-1984

KUMARAGURU COLLEGE OF TECHNOLOGY

COIMBATORE - 641 006

DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING

ISO 9001:2000
Certified

CERTIFICATE

This is to certify that the project report entitled

ECLECTIC AUTOMATION GUIDED LITHERING EQUIPAGE

(E.A.G.L.E)

Is the bonafide work done by

____

in partial fulfillment of the requirements for the Award of the degree of Bachelor of Engineering In Electrical and Electronics Engineering Branch of Bharathiar university, Coimbatore

R. Mehalakshmi

Guide
Date:

Head of the Department

Certified that the candidate ______________________ with

University Register No. ______________________ was examined in

Project work Viva-Voce Examination on ______________________

Internal Examiner

External Examiner

R. VENKATARAMANI
. General Manager (F & A)

INSTRUMENTATION LIMITED
(A GOVERNMENT OF INDIA ENTERPRISE)
Kanjikode west - 678 623, Palakkad, Kerala India.
Tel. Off : (0491) 2566127 to 2566130.
: 2566131 (Direct)
Resi : (0491) 2500340.
Fax : (0491) 2566135, 2566240.
E-mail : fa@ilpgt.com

05 March 2004

CERTIFICATE

This is to certify that the following students of Kumaraguru College of Technology-Coimbatore pursuing Final year BE Electrical and Electronics Engineering have Successfully completed their final year academic project in our company premises.

S.Sabarisan

M.Vijayakumar

S.Vinod

V.K.Prasanth

V. R. VENKATARAMANI
Dy. General Manager (F & A)
Instrumentation Limited
(A Govt of India Enterprise)
Kanjiko
Kerala

ACKNOWLEDGEMENT

ACKNOWLEDGEMENT

The satisfaction that accompanies the successful completion of any task would be but incomplete without mentioning the people whose constant guidance and encouragement crowns all efforts with success.

Our sincere thanks to Dr. T. Kameswaran, Head of the Department of Electrical and Electronics, Kumaraguru College of Technology for providing us this opportunity to take up the project work. He has been a source of great encouragement and inspiration throughout the curriculum.

We would like to express our gratitude to our guide, Mrs.R. Mahalakhsmi, Senior Lecturer, Department of Electrical and Electronics, Kumaraguru College of Technology, who guided us throughout the project and encouraged us to successfully complete our work. Our thanks to her for the immense help and guidance during the course of this project.

We take pleasure in thanking Mr.Millton and Mr.Venkataramani of INSTRUMENTATION LTD - PALAKAD, for providing us with technical assistance and guiding us through this project.

We also thank our staff members and friends for helping us with relevant details and especially their constant encouragement, without which this project would have not attained this form.

ABSTRACT

ABSTRACT

The impact of modernization is reflected in the day today life. Modern industries are now concentrating for automation of their machines by implementing robots. To implement robotics, industries are now seeking for good and reliable controllers. This project termed as the "E.A.G.L.E" which is expanded as "Eclectic Automation Guided Lithering Equipage" meaning 'source selecting automation guided slow moving vehicle' concentrates on developing a reliable controller. In industries robotics find two main applications, one is as a robotic arm structure and the other is as a guided vehicle. These days industries are importing such controllers from other countries which are costly, but their cost can be reduced if these controllers are manufactured in our country by the industries them self. This project is to give the industries the basic template of a modern controller. In this project the type of controllers are chosen based on their applications. For robotic arm the controller should be a noise free, portable, with a long distance span of control and for the vehicle the controller should be a process supportive, low distance span of control and which is of an affordable cost.

The controller for the robotic arm is achieved by using Digital Signal Processor (TMS320C25) and the controller for the vehicle is achieved by using a parallel port LPT1 interfaced computer controlled system. The robotic arm with a camera and its controller, mounted over the vehicle to give an appearance of an exploration vehicle.

CONTENTS

CONTENTS

INTRODUCTION

INTRODUCTION

Industries are now concentrating for automation of their machines by implementing robots. To implement robotics, industries are now seeking for good and reliable controllers. These days industries are importing such controllers from other countries which are costly, but their cost can be reduced if these controllers are manufactured in our country by the industries them self. An attempt has been made in this direction to design and develop the basic template of a modern controller. In this project the type of controllers are chosen based on their applications. For robotic arm the controller should be a noise free, portable, with a long distance span of control and for the vehicle the controller should be a process supportive, low distance span of control and which is of an affordable cost. The controller for the robotic arm is achieved by using Digital Signal Processor (TMS320C25) and the controller for the vehicle is achieved by using a parallel port LPT1 interfaced computer controlled system.

ROBOTIC ARM CONTROLLER

BASIC PRINCIPLE OF DSP BASED ROBOTIC ARM CONTROL

The Robotic Arm manipulation makes use of two stepper motors, one is for shoulder movement (Horizontal) and the other is for fist movement (Vertical).To control these motors DSP Controller (TMS320C25 Processor) is used. The stepper motor rotates based on the sequence of pulses given to it. If these pulse sequences are controlled then the motor rotation angle and the rotation direction can be changed.

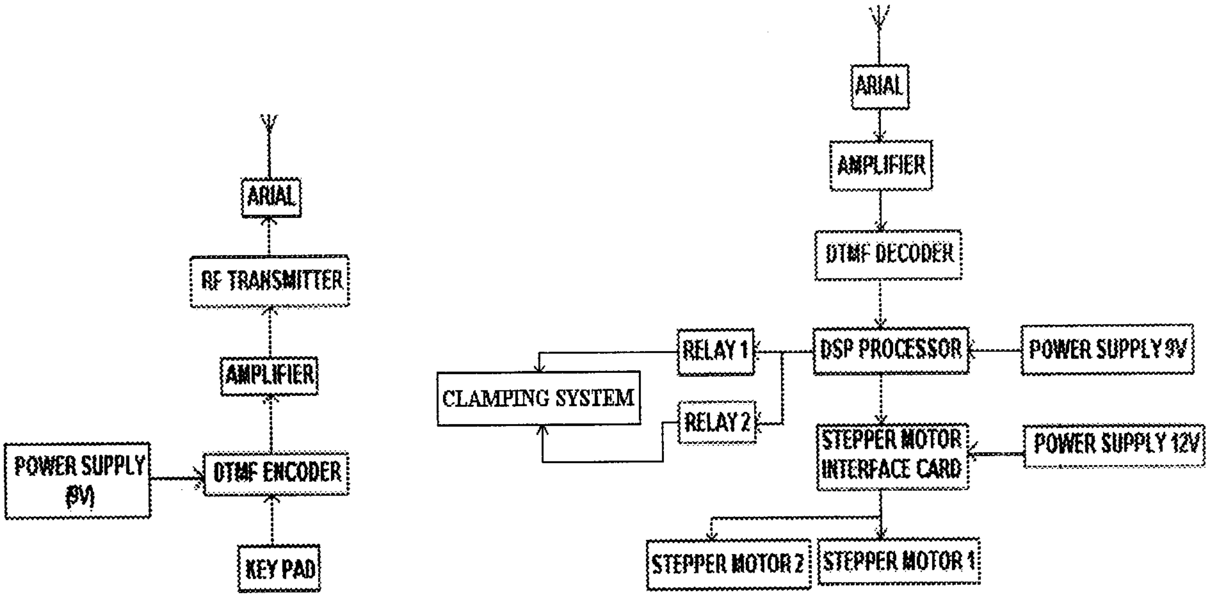
When the operator presses a control key then a keycode for the key pressed is generated, this code is then encoded by the DTMF generator. The encoded signal is then amplified by an amplifier circuit and is sent to the RF Transmitter. The RF Transmitter then modulates the frequency of the signal and transmits it through an arial.

The Receiver then senses the signal through an arial and demodulates the signal and sends it to the amplifier. The amplifier then increases the magnitude of the signal and the DTMF decoder converts the signal to keycode, this keycode generated consists of noise or interferences. This impure signal is then processed by the DSP Controller and the noises are eliminated.

Now the noise free keycode is recognized by the DSP Processor. Depending on the Keycode the Motor pulse sequence is generated. Since the pulse signal is +5V, but the stepper motor requires +12V for its operation the +5V pulse is used to switch ON a set of power transistors (Stepper Motor Interface Card) which promotes +12V to the stepper motors for its operation. An LCD display is provided to display the current status of the executed function. In case of relay energising this +5V pulse from the DSP Controller is used directly for the relay operation. The relays can be used to operate a clamping system for Pick and Place operation.

The Power supplies used are made available by a 9-0-9 V and a 12-0-12 V transformer with a diode bridge rectifier circuit.

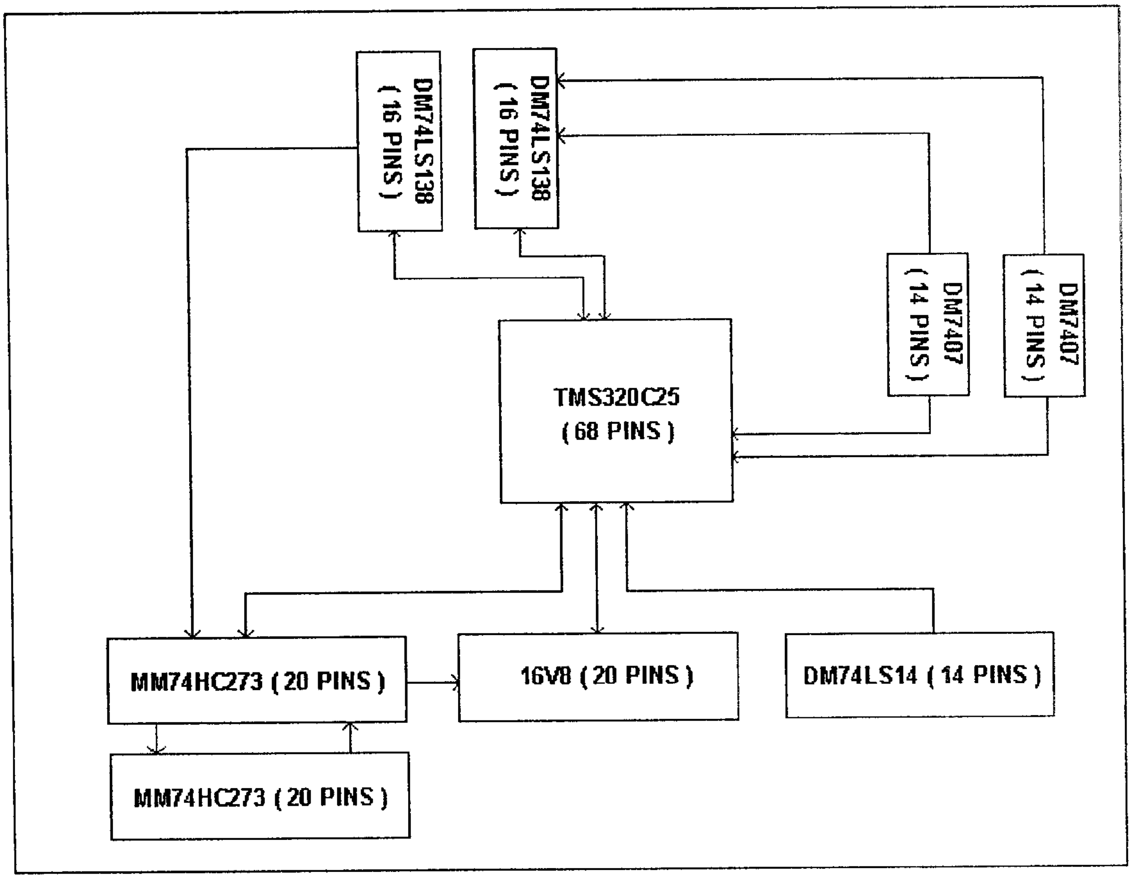
BLOCK DIAGRAM OF ROBOTIC ARM CONTROL



RF TRANSMITTER

RF RECEIVER

BLOCK DIAGRAM OF DSP CONTROLLER



DTMF CONTROL

Introduction

DTMF is the generic name for push button telephone signalling, equivalent to the Bell System's Touch Tone . Dual Tone Multi Frequency (DTMF) signalling is quickly replacing dial-pulse signalling in telephone networks worldwide. DTMF is becoming popular in interactive control applications, such as telephone banking or electronic mail systems, in which the user can select options from a menu by sending DTMF signals from a telephone.

To generate (encode) a DTMF signal, the ADSP – 2100 adds together two sinusoids, each created by software. For DTMF decoding, the ADSP – 2100 looks for the presence of two sinusoids in the frequency domain using modified Goertzel algorithms.

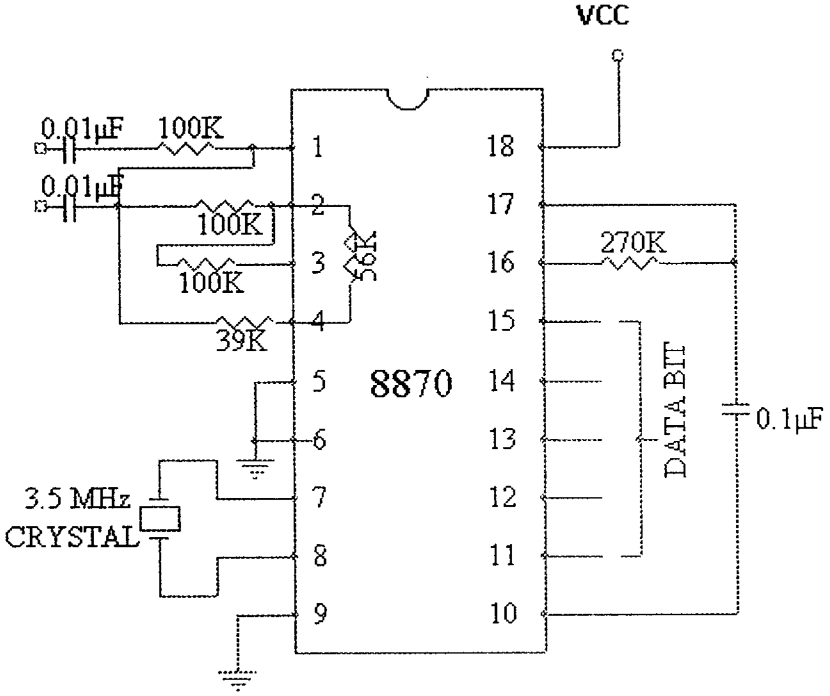
DTMF signals are interfaced to the analog via codec (coder / decoder) chips or linear analog to digital (A/D) converters and digital to analog (D/A) converters. Codec chips contain all the necessary A/D, D/A sampling and filtering circuitry for a bi-directional analog/digital interface. These codecs with on-chip filtering are sometimes called codecs /filter combo chips, or combo chips. They are referred to as codecs.

The codec channel used is band limited to pass only frequencies between 200Hz and 3400Hz. The codec also incorporates companding (audio compressing / expanding) circuitry for either of the two companding standards (A law and μ255 law). Companding is the process of logarithmically compressing a signal at the source and expanding it at the destination to maintain a high end to end dynamic range while reducing the dynamic range requirement within the communication channel.

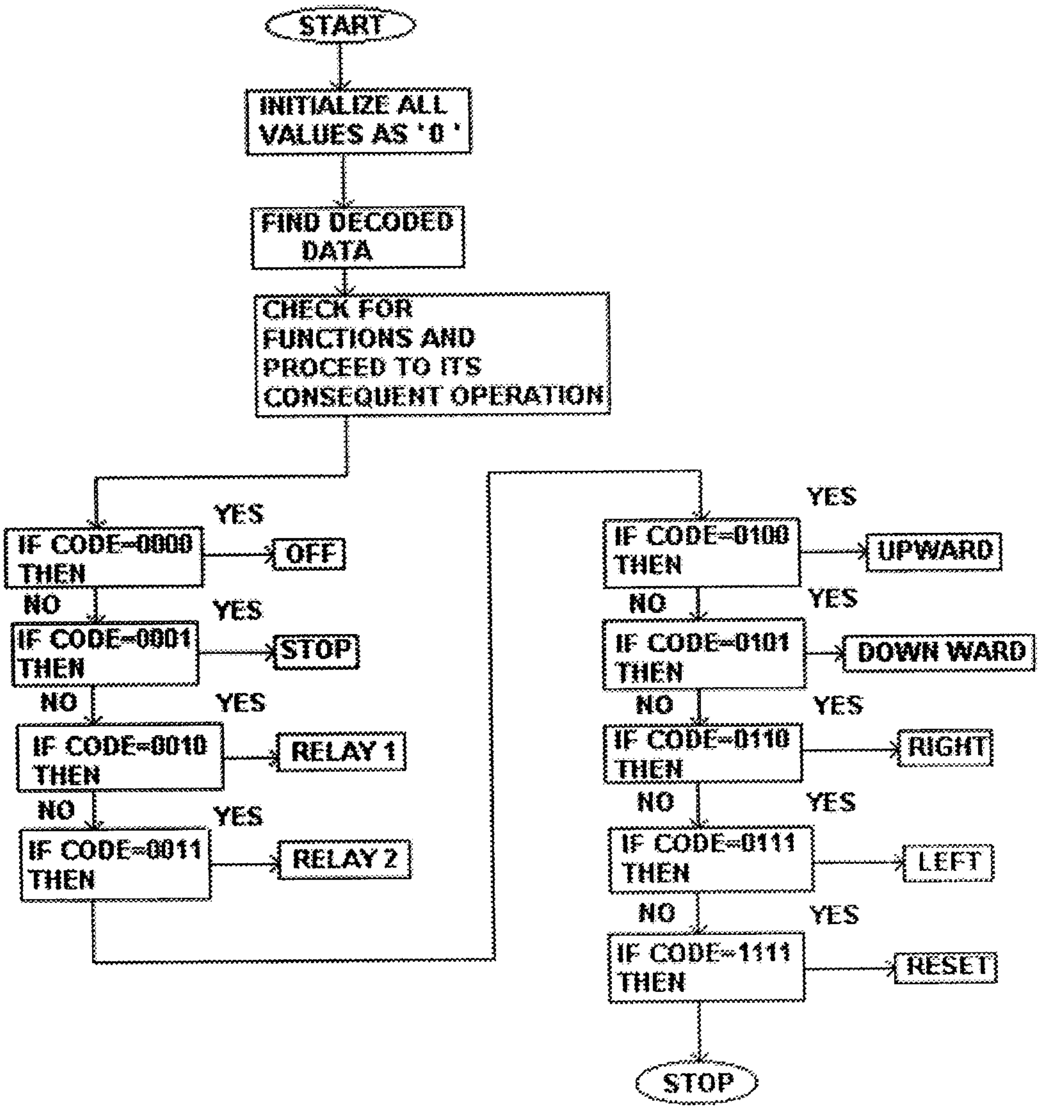
In the DTMF signal generation, the ADSP-2100 reads DTMF digits stored in data memory in a relocatable look up list. Alternatively, a DTMF keypad could be used for digit entry. In either case, the resultant DTMF tones are generated mathematically and added together. The values are logarithmically compressed and passed to the codec chip for converting it to analog signals. Multi channel DTMF signal generation is performed by simply time multiplexing the processor among the channels.

On the receiving end, the ADSP – 2100 reads the logarithmically compressed digital data from the codec's 8-bit parallel data bus and logarithmically expands it to its 16-bit linear format, performing a Goertzel algorithm (or) a fast DFT (discrete Fourier transform) calculation for detecting each tone, then passes the results through several tests to verify whether a valid DTMF digit is received. The result is coded and written to a memory mapped I/O port. Multi channel DTMF decoding is also performed by time-multiplexing the channels.

DTMF Generator



DSP PROGRAM – FLOW CHART



DSP PROGRAM FOR ROBOTIC ARM CONTROL

```
        ldpk   #100h
        lacc   #00h
        sacl   8018h
        sacl   8010h
        ldpk   #100h
        call   lcd_init
        ldpk   #06h
        lacl   #34h
        sacl   303h
        ldpk   #100h
        lack   #00h
        sacl   8020h
        ldpk   #6h
        sacl   3b0h
        sacl   310h
        sacl   315h
        ldpk   #100h
        call   display
        call   delay
set:    ldpk   #100h
        zals   8028h
        andk   #10h
        bnz    set
setl:   ldpk   #100h
        zals   8028h
        andk   #10h
        bz     setl
loop:   zals   8028h
        andk   #0fh
        ldpk   #6
        sacl   3b0h
        sacl   310h
        ldpk   #100h
        call   display1
        ldpk   #6
        lacl   3b0h
        subk   #1
        bz     relay1
        lacl   3b0h
        subk   #2
        bz     upward
        lacl   3b0h
        subk   #3
        bz     relay2
        lacl   3b0h
        subk   #4
        bz     left
        lacl   3b0h
        subk   #5
        bz     downward
        lacl   3b0h
        subk   #6
        bz     right
        lacl   3b0h
        subk   #7
        bz     stop
        b      set
stop:   ldpk   #100h
        lacl   #00h
        sacl   8010h
        b      set
```

```
upward:
          ldpk   #100h
          lacc   #5h
          sacl   8020h
          call   delay
          lacc   #9h
          sacl   8020h
          call   delay
          lacc   #0ah
          sacl   8020h
          call   delay
          lacc   #06h
          sacl   8020h
          call   delay
          ldpk   #100h
          zals   8028h
          andk   #10h
          bz     upward
          b      loop
downward:
          ldpk   #100h
          lacc   #06h
          sacl   8020h
          call   delay
          lacc   #0ah
          sacl   8020h
          call   delay
          lacc   #09h
          sacl   8020h
          call   delay
          lacc   #05h
          sacl   8020h
```

```
call    delay
          ldpk   #100h
          zals   8028h
          andk   #10h
          bz     downward
          b      loop
right:    ldpk   #100h
          lacc   #50h
          sacl   8020h
          call   delay
          lacc   #90h
          sacl   8020h
          call   delay
          lacc   #0a0h
          sacl   8020h
          call   delay
          lacc   #060h
          sacl   8020h
          call   delay
          ldpk   #100h
          zals   8028h
          andk   #10h
          bz     right
          b      loop
relay1:   ldpk   #100h
          lacl   #01h
          sacl   8010h
          b      set
relay2:   ldpk   #100h
          lacl   #02h
          sacl   8010h
          b      set
```

```
lcd_init: lacl  #038h
          sacl  8000h,0
          call  lcd_write
          call  delay
          lac   #038h
          sacl  8000h,0
          call  lcd_write
          call  delay
          lac   #038h
          sacl  8000h,0
          call  lcd_write
          call  delay
          lac   #01h
          sacl  8000h,0
          call  lcd_write
          call  delay
          lac   #0ch
          sacl  8000h,0
          call  lcd_write
          call  delay
          ret
display:  ldpk  #100h
          lac   #80h
          sacl  8000h,0
          call  lcd_write
          call  delay
          lac   #'R'
          sacl  8000h,0
          call  data_write
          call  delay
          lac   #'F'
          sacl  8000h
```

```
call  data_write
call  delay
lac   #'C'
sacl  8000h
call  data_write
call  delay
lac   #'O'
sacl  8000h
call  data_write
call  delay
lac   #'N'
sacl  8000h
call  data_write
call  delay
lac   #'T'
sacl  8000h
call  data_write
call  delay
lac   #'R'
sacl  8000h
call  data_write
call  delay
lac   #'O'
sacl  8000h
call  data_write
call  delay
lac   #'L'
sacl  8000h
call  data_write
call  delay
lac   #' '
sacl  8000h
```

```
          call   data_write
          call   delay
          lac    #'R'
          sacl   8000h
          call   data_write
          call   delay
          lac    #'O'
lcd_write:
          ldpk   #100h
          lacc   #04h
          sacl   8008h
          call   small_delay
          lacc   #00h
          sacl   8008h
          call   small_delay
          ret
delay:
          lar    ar2,#0ffffh
del:      sbrk   #01h
          banz   del, ar2
          ret
small_delay:
          lar    ar3,#0ffh
sdel:     sbrk   #01h
          banz   sdel,ar3
          ret
          .end
```

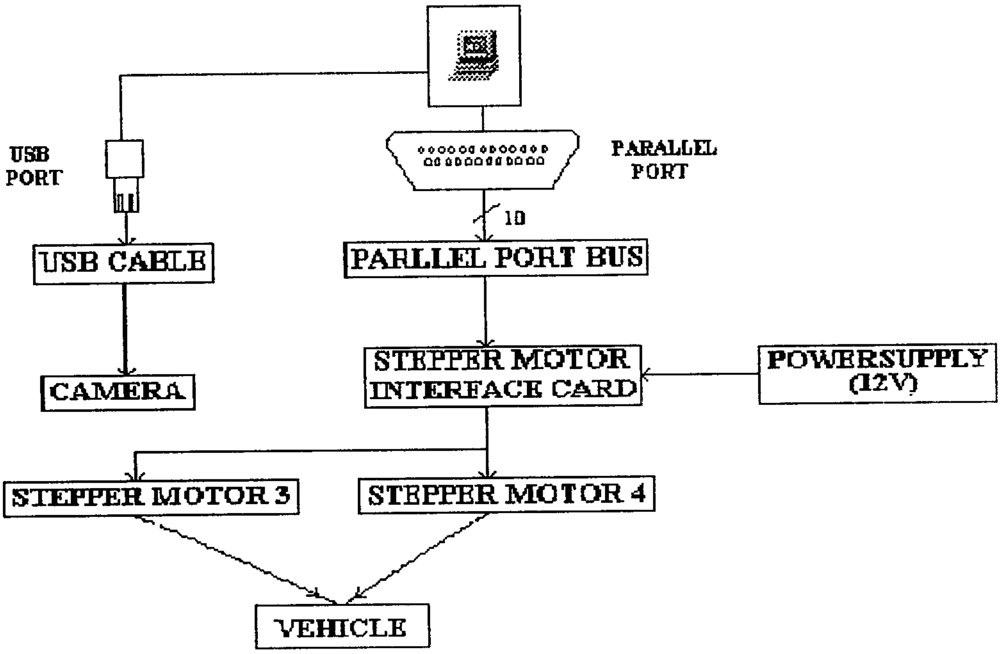
ROBOTIC VEHICLE

BASIC PRINCIPLE OF ROBOTIC VEHICLE

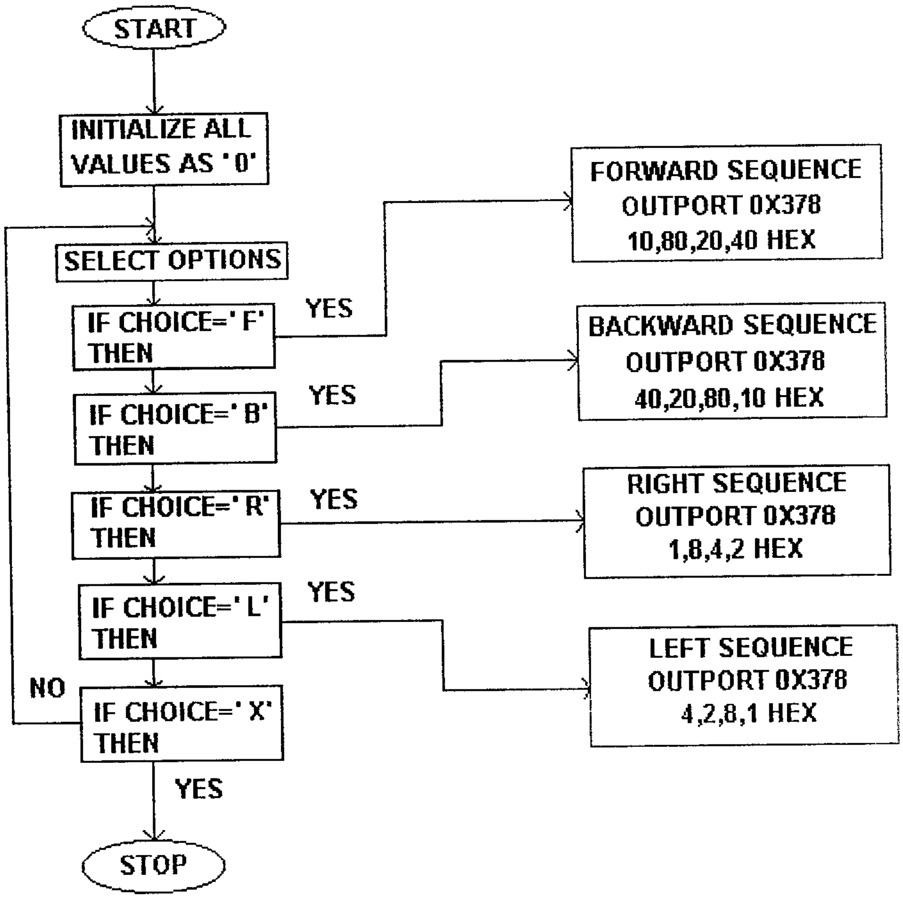
The Robotic Vehicle consists of two stepper motors one is for changing Right - Left directions and the other is for changing Forward – Reverse directions. The Vehicle is controlled by a Computer through a Parallel Port LPT1 interface. In a parallel port there are 25 pins 13 in the upper row and 12 in the lower row. In this project we make use of only the upper row. In the upper row the first three pins are select, earth and negative pins, the next eight pins are Data pins through which the stepper motor pulse sequence is transmitted.

The Pulse sequence for the stepper motor is first observed, then the pulse sequence is converted into a Hexa Decimal value and is outported to the parallel port by a C-Program. The pulse signal is +5V, but the stepper motor requires +12V for its operation so, this +5V pulse is used to switch ON a set of power transistors(Stepper Motor Interface Card) which promotes +12V to the stepper motors for its operation. The Power supply used are made available by a 12-0-12 V transformer with a diode bridge rectifier circuit. A Camera with a USB Cable transmission is mounted on the arm to capture the view ahead. The captured view can be stored as a movie profile or as a snap shot picture.

BLOCK DIAGRAM OF ROBOTIC VEHICLE CONTROL



VEHICLE CONTROL PROGRAM - FLOW CHART


START
INITIALIZE ALL VALUES AS '0'
SELECT OPTIONS
IF CHOICE=' F' THEN
YES
FORWARD SEQUENCE OUTPORT 0X378 10,80,20,40 HEX
IF CHOICE=' B' THEN
YES
BACKWARD SEQUENCE OUTPORT 0X378 40,20,80,10 HEX
IF CHOICE=' R' THEN
YES
RIGHT SEQUENCE OUTPORT 0X378 1,8,4,2 HEX
IF CHOICE=' L' THEN
YES
LEFT SEQUENCE OUTPORT 0X378 4,2,8,1 HEX
NO
IF CHOICE=' X' THEN
YES
STOP

PROGRAM FOR ROBOTIC VEHICLE CONTROL

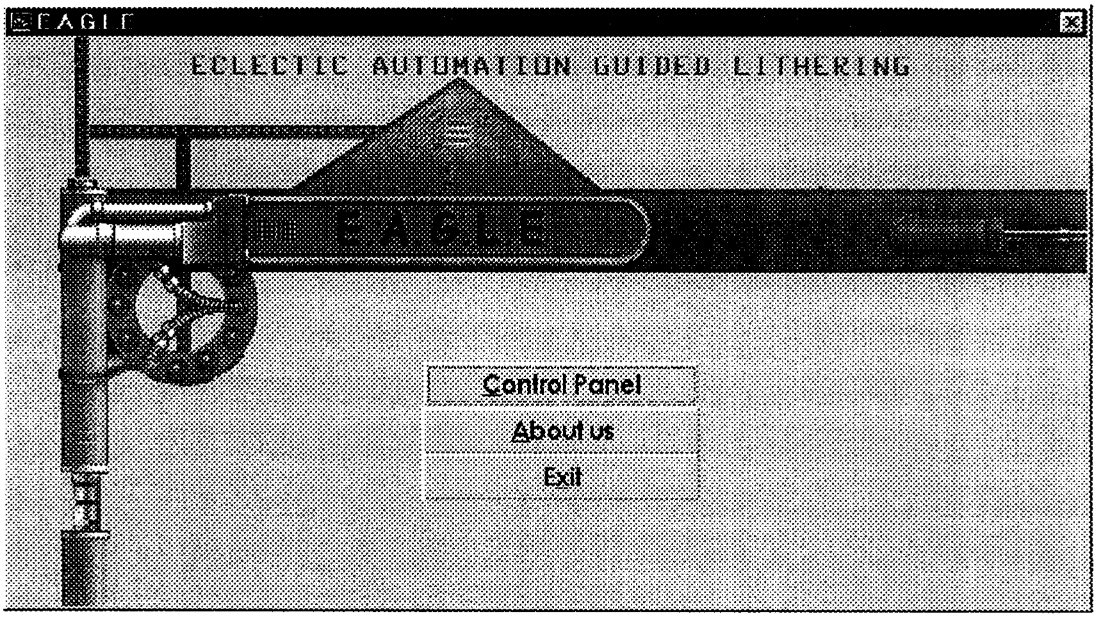
```
#include <stdio.h>
#include <conio.h>
#include<dos.h>
#include<ctype.h>
#include<string.h>
void main()
{
 char choice;
 int i;
 flushall();
 clrscr();
 sab:
 textcolor(WHITE);
 gotoxy(23,1);cprintf("ÛÛÛÛÛÛÛÛÛÛÛÛÛÛÛÛÛÛÛÛÛÛÛÛÛÛÛÛÛÛÛÛÛ");
 gotoxy(23,2);cprintf("ÛÛ                                                        ÛÛ");
 gotoxy(23,3);cprintf("ÛÛ                                                        ÛÛ");
 gotoxy(23,4);cprintf("ÛÛ                                                        ÛÛ");
 gotoxy(23,5);cprintf("ÛÛÛÛÛÛÛÛÛÛÛÛÛÛÛÛÛÛÛÛÛÛÛÛÛÛÛÛÛÛÛÛÛ");
 textcolor(WHITE+BLINK);
 textcolor(WHITE);
 gotoxy(27,3);cprintf("E.A.G.L.E CONTROL PROGRAM");
 gotoxy(28,9); printf("ÉÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍ»");
 gotoxy(28,10);printf("º F -> Forward     º");
 gotoxy(28,11);printf("º B -> Backward    º");
 gotoxy(28,12);printf("º R -> Right       º");
 gotoxy(28,13);printf("º L -> Left        º");
 gotoxy(28,14);printf("º S -> Stop        º");
 gotoxy(28,15);printf("º X -> Exit        º");
 gotoxy(28,16);printf("ÈÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍ¼");
 gotoxy(28,18);printf("Enter Your Choice : ");
```

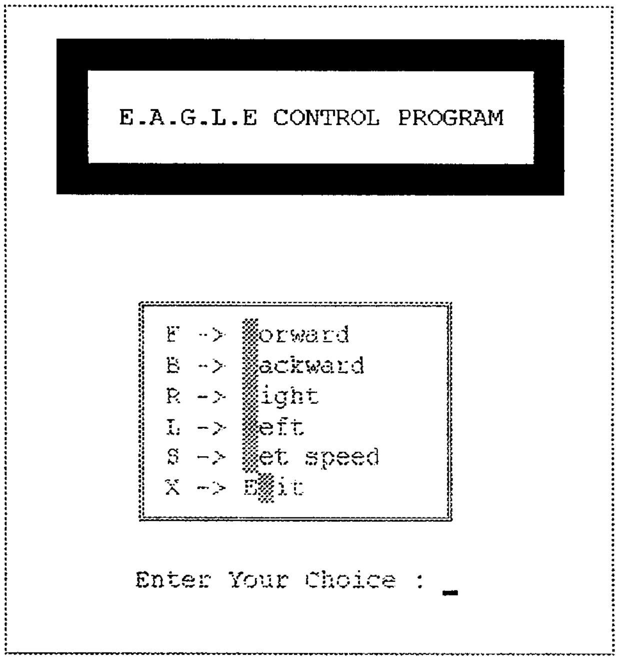
```
textcolor(5+BLINK);
gotoxy(35,10);cprintf("F");
gotoxy(35,11);cprintf("B");
gotoxy(35,12);cprintf("R");
gotoxy(35,13);cprintf("L");
gotoxy(35,14);cprintf("S");
gotoxy(36,15);cprintf("X");
textcolor(WHITE);
gotoxy(48,18);scanf("%c",&choice);
choice=toupper(choice);
if(choice =='S')
        {       printf("STOP");
                while(!kbhit())
                {
                outportb(0x378,0x0);
                }
                clrscr();
                goto sab;               }
    else if(choice=='F')
        {       printf("FORWARD");
                while(!kbhit())
                {
                outportb(0x378,0x40);
                delay(100);
                outportb(0x378,0x20);
                delay(100);
                outportb(0x378,0x80);
                delay(100);
                outportb(0x378,0x10);
                delay(100);
                outportb(0x378,0x0);            }
                clrscr();
```

```
goto sab;            }
    else if(choice=='B')
       {
              printf("BACKWARD");
              while(!kbhit())
              {
              outportb(0x378,0x10);
              delay(100);
              outportb(0x378,0x80);
              delay(100);
              outportb(0x378,0x20);
              delay(100);
              outportb(0x378,0x40);
              delay(100);
              outportb(0x378,0x0);
              }
              clrscr();
              goto sab;
              }
     else if(choice=='R')
     {
               printf("RIGHT");
               while(!kbhit())
               {
               outportb(0x378,0x1);
               delay(100);
               outportb(0x378,0x8);
               delay(100);
               outportb(0x378,0x2);
               delay(100);
               outportb(0x378,0x4);
               delay(100);
outportb(0x378,0x0);
                }
                clrscr();
                goto sab;
                }
    else if(choice=='L')
        {
                printf("LEFT");
                while(!kbhit())
                {
                outportb(0x378,0x4);
                delay(100);
                outportb(0x378,0x2);
                delay(100);
                outportb(0x378,0x8);
                delay(100);
                outportb(0x378,0x1);
                delay(100);
                outportb(0x378,0x0);        }
                clrscr();
                goto sab;
                }
    else if(choice=='X')
        {
                {
                clrscr();
                normvideo();
                exit(0);
                }
        }
 goto sab;
 }
```

E.A.G.L.E CONTROL PANEL

E.A.G.L.E CONTROL PANEL





```
E.A.G.L.E CONTROL PROGRAM

F -> Forward
B -> Backward
R -> Right
L -> Left
S -> Set speed
X -> Exit

Enter Your Choice : _
```

TECHNICAL DETAILS

GENERAL DESCRIPTION OF ICs AND COMPONENTS

IC TMS320C25

Digital Signal Processors

- 80-ns Instruction Cycle Time
- 544 Words of On-Chip Data RAM
- 4K Words of On-Chip Secure Program EPROM (TMS320E25)
- 4K Words of On-Chip Program ROM (TMS320C25)
- 128K Words of Data/Program Space
- 32-Bit ALU/Accumulator
- 16 × 16-Bit Multiplier With a 32-Bit Product
- Block Moves for Data/Program Management
- Repeat Instructions for Efficient Use of Program Space
- Serial Port for Direct Codec Interface
- Synchronization Input for Synchronous Multiprocessor Configurations
- Wait States for Communication to Slow Off-Chip Memories/Peripherals
- On-Chip Timer for Control Operations
- Single 5-V Supply
- Packaging: 68-Pin PGA, PLCC, and CER-QUAD
- 68-to-28 Pin Conversion Adapter Socket for EPROM Programming

NMOS Technology

TMS32020 200-ns cycle time

CMOS Technology

TMS320C25 100-ns cycle time
TMS320E25 100-ns cycle time
TMS320C25-50 80-ns cycle time

General Description

The TMS320 family of 16/32-bit single-chip digital signal processors combines the flexibility of a high-speed controller with the numerical capability of an array processor, thereby offering an inexpensive alternative to multichip bit-slice processors. The highly paralleled architecture and efficient instruction set provide speed and flexibility to produce a MOS microprocessor family that is capable of executing more than 12.5 MIPS (million instructions per section). The TMS320 family optimizes speed by implementing functions in hardware that other processors implement through microcode or software. This hardware-intensive approach provides the design engineer with processing power previously unavailable on any other chip.

The TMS320 family consists of three generations of digital signal processors. The first generation contains the TMS32010 and its series. The second generation includes the TMS32020, TMS320C25, and TMS320E25. The TMS320C25 is a floating-point DSP device designed for even higher performance. Many features are common among the TMS320 processors. Specific features are added in each processor to provide different cost/performance tradeoffs. Software compatibility is maintained throughout the family to protect the user's investment in architecture. Each processor has software and hardware tools to facilitate rapid design.

The TMS32010, the first NMOS digital signal processor in the TMS320 family, was introduced in 1983. Its powerful instruction set, inherent flexibility, high-speed number-crunching capabilities, and innovative architecture have made this high performance, cost-effective processor the ideal solution to many telecommunications, computer, commercial, industrial, and military applications.The second generation of the TMS320 family (referred to as TMS320C2x) includes four members, the TMS32020, TMS320C25, TMS320C25-50, and TMS320E25. The architecture of these devices is based upon that of the TMS32010.

The TMS320C20, processed in NMOS technology, is source-code compatible with the TMS32010 and in many applications is capable of two times the throughput of the first-generation devices. Its enhanced instruction set (109 instructions), large on-chip data memory (544 words), large memory spaces, on-chip serial port, and hardware timer makes the TMS320C20 a powerful addition to the TMS320 family.

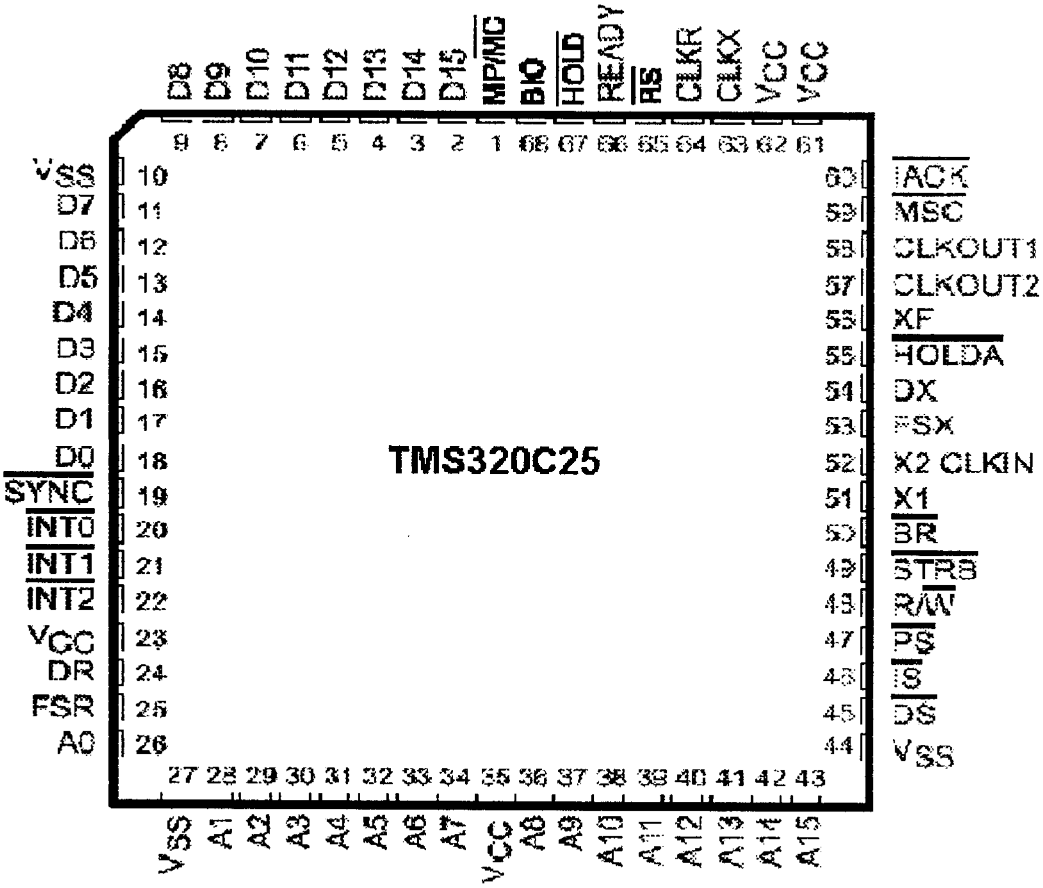
The TMS320C25 is the second member of the TMS320 second generation. It is processed in CMOS technology, is capable of an instruction cycle time of 100 ns, and is pin-for-pin and object-code compatible with the TMS32020. The TMS320C25's enhanced feature set greatly increases the functionality of the device over the TMS32020. Its enhancement includes, 24 additional instructions (133 total), eight auxiliary registers, an eight-level hardware stack, 4K words of on-chip program ROM, a bit-reversed indexed addressing mode, and the low-power dissipation inherent to the CMOS process. An extended-temperature range version (TMS320C25GBA) is also available.

The TMS320C25-50 is a high-speed version of the TMS320C25. It is capable of an instruction cycle time of less than 80 ns. It is architecturally identical to the original 40MHz version of the TMS320C25 and thus, it is pin-for-pin and object-code compatible with the TMS320C25. The TMS320E25 is identical to the TMS320C25, with the exception that the on-chip 4K-word program ROM is replaced with a 4K-word on-chip program EPROM. On-chip EPROM allows real-time code development and modification for immediate evaluation of system performance.

Features of ADSP 2105

- 16 bit fixed point DSP microprocessors with on chip memory
- Enhanced Harvard architecture for three bus performance : instruction bus and dual data buses
- Independent computational units : ALU, multiplier/accumulator and shifter
- Single cycle instruction execution and multifunction instructions
- On chip program memory and data memory
- Separate on chip buses for program and data memory
- Program memory stores both program and data
- Dual data address generators with modulo and bit reverse addressing

Pin Diagram of TMS320C25



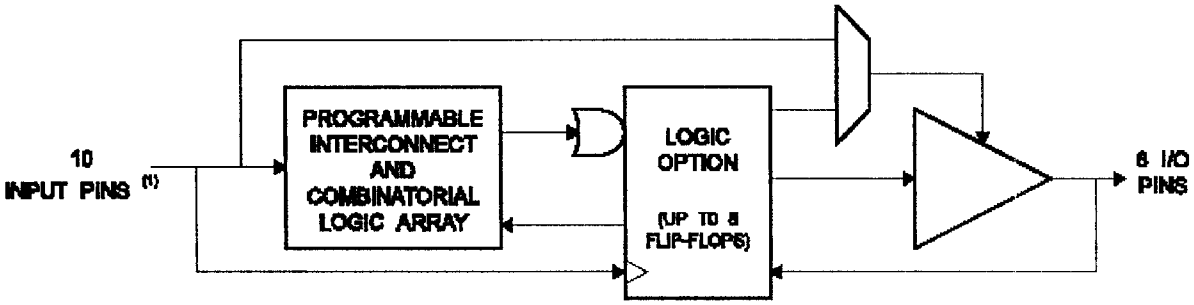
IC 16V8

General Description

The ATF16V8C is a high-performance EECMOS Programmable Logic Device that utilizes Atmel's proven electrically-erasable Flash memory technology. Speeds down to 5 ns and a 100 µA pin-controlled power-down mode option are offered. All speed ranges are specified over the full 5V ± 10% range for industrial temperature ranges and 5V ± 5% for commercial range 5-volt devices.

The ATF16V8C incorporates a superset of the generic architectures, which allows direct replacement of the 16R8 family and most 20-pin combinatorial PLDs. There are eight outputs which are allocated eight product terms individually. Three different modes of operations are configured automatically with the soft-ware and allows highly complex logic functions to be realized. The ATF16V8C can significantly reduce total system power, thereby enhancing system reliability and reducing power supply costs. When pin 4 is configured as the power-down control pin, supply current drops to less than 100 µA whenever the pin is high. If the power-down feature isn't required for a particular application then pin 4 may be used as a logic input. Also, the pin keeper circuits eliminate the need for internal pull-up resistors along with their attendant power consumption.

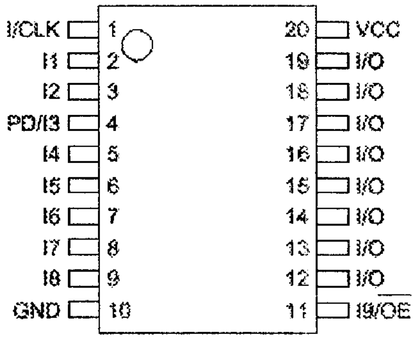
Block Diagram of 16V8



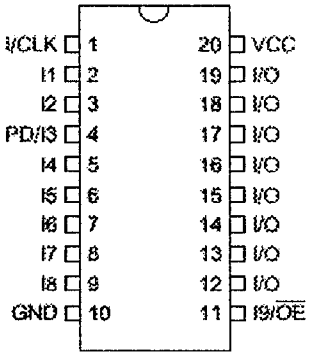
Pin Configuration

Pin Name	Function
CLK	Clock
I	Logic Inputs
I/O	Bidirectional Buffers
$\overline{OE}$	Output Enable
VCC	+5V Supply
PD	Power-down

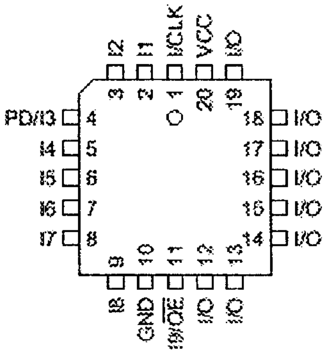
TSSOP



DIP/SOIC



PLCC



IC DM7407

Hex Buffers with High Voltage Open-Collector Outputs

General Description

This device contains six independent gates each of which performs a buffer function. The open-collector outputs require external pull-up resistors for proper logical operation.

Pull-Up Resistance Equation

$$R_{MAX} = \frac{V_O\ (Min) - V_{OH}}{N_1\ (I_{OH}) + N_2\ (I_{IH})}$$

$$R_{MIN} = \frac{V_O\ (Max) - V_{OL}}{I_{OL} - N_3\ (I_{IL})}$$

Where: N1 (IOH) = total maximum output high current for all

outputs tied to pull-up resistor

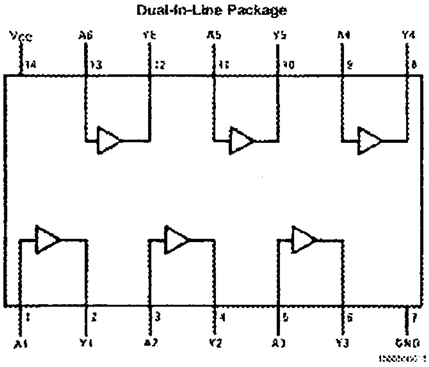
N2 (IIH) = total maximum input high current for all

inputs tied to pull-up resistor

N3 (IIL) = total maximum input low current for all in-puts

tied to pull-up resistor

Connection Diagram



Function Table

Y = A

Input	Output
A	Y
L	L
H	H

H = High Logic Level
L = Low Logic Level

IC MM74HC273

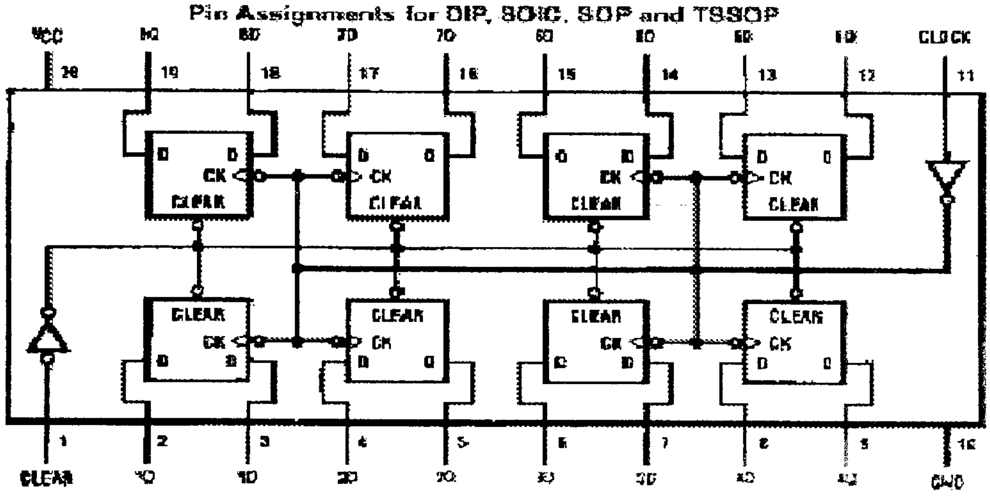
Octal D-Type Flip-Flops with Clear

General Description

The MM74HC273 edge triggered flip-flops utilize an advanced silicon-gate CMOS technology to implement D-type flip-flops. They posses high noise immunity, low power, and speeds comparable to low power Schottky TTL circuits. This device contains 8 master-slave flip-flops with a common clock and common clear. Data on the
D input having the specified setup and hold times is transferred to the Q output on the LOW to HIGH transition of the CLOCK input.

The CLEAR input when becomes LOW, sets all outputs to a low state. Each output can drive 10 low power Schottky TTL equivalent loads. The MM74HC273 is functionally as well as pin compatible to the 74LS273. All inputs are protected from damage due to static discharge by diodes connected betweenVCC and ground.

Connection Diagram



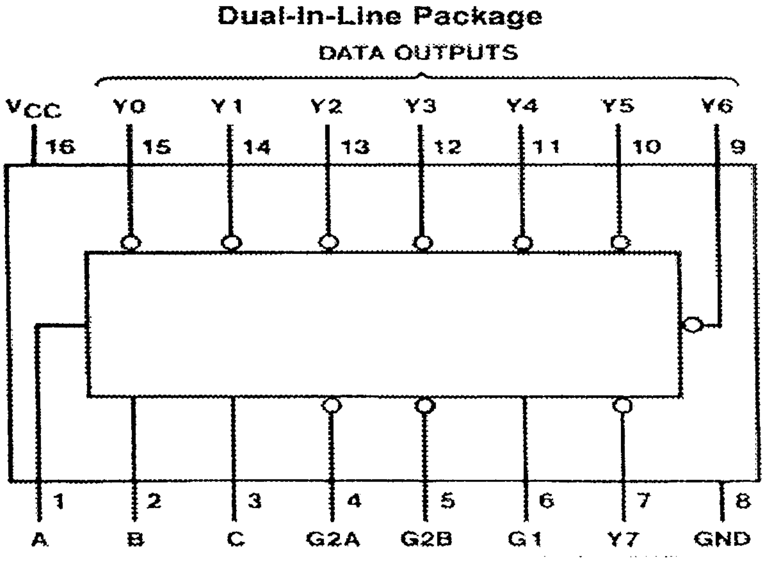
IC DM74LS138

Decoders / Demultiplexers

General Description

These Schottky-clamped circuits are designed to be used in high-performance memory-decoding or data-routing applications, requiring very short propagation delay times. In high-performance memory systems these decoders can be used to minimize the effects of system decoding. When used with high-speed memories, the delay times of these decoders are usually less than the typical access time of the memory. This means that the effective system delay introduced by the decoder is negligible. The LS138 decodes one-of-eight lines, based upon the conditions at the three binary select inputs and the three enable inputs. Two active-low and one active-high enable inputs reduce the need for external gates or inverters during its expansion. A 24-line decoder can be implemented without external inverters, but a 32-line decoder requires only one inverter. An enable input can be used as a data input for demultiplexing applications.

Connection Diagram



IC DM74LS14

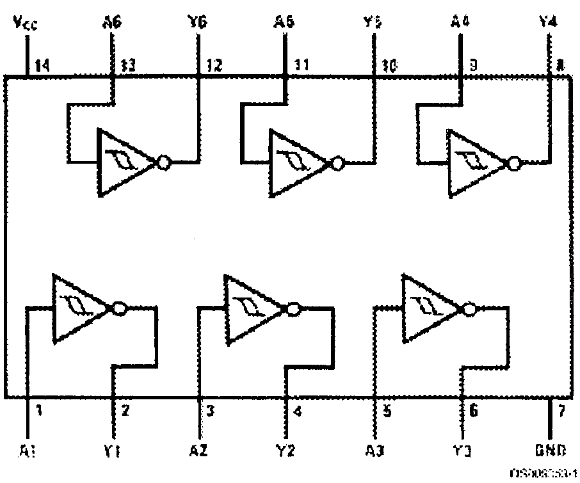
Hex Inverters with Schmitt Trigger Inputs

General Description

This device contains six independent gates each of which performs the logic INVERT function. Each input has hysteresis which increases the noise immunity and transforms a slowly changing input signal to a fast changing, noise free output.

Connection Diagram

Dual-In-Line Package



Function Table

$Y = \overline{A}$

Input	Output
A	Y
L	H
H	L

H = High Logic Level
L = Low Logic Level

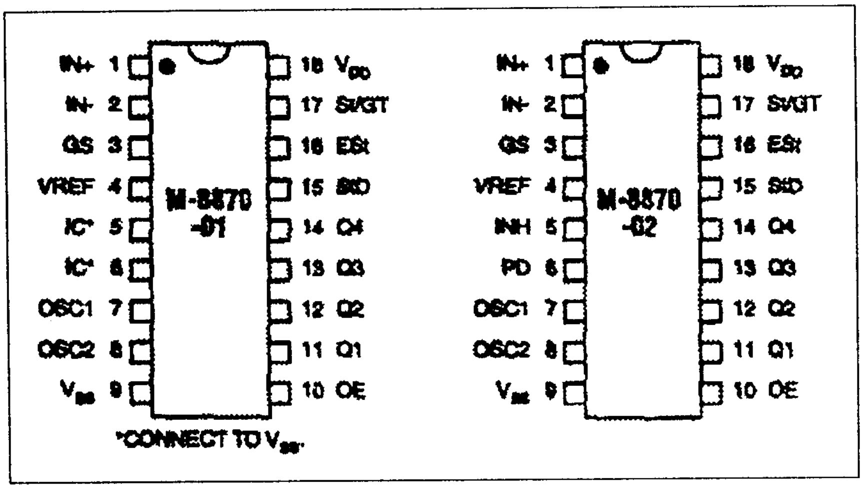
IC M-8870

DTMF Receiver

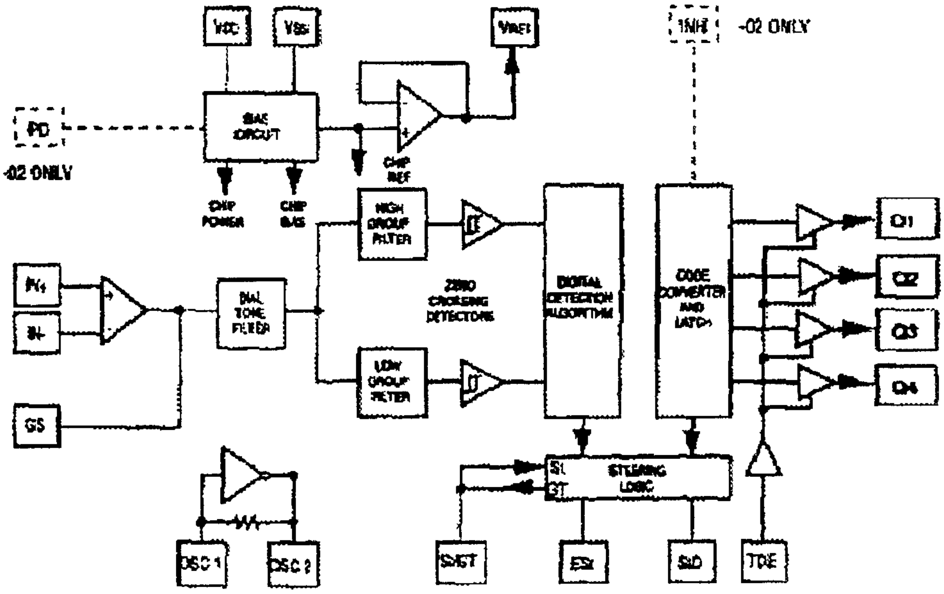
General Description

The IC M-8870 is a full DTMF receiver that integrates both band split filter and decoder functions into a single 18 pin DIP or SOIC package. Manufactured using state of the art CMOS process technology, the M-8870 offers low power consumption (35 mw max) and precise data handling. Its filter section uses switched capacitor technology for both the high and low group filters and for dial tone rejection. Its decoder uses digital counting techniques to detect and decode all 16 DTMF tone pairs into a 4-bit code. External component count is minimized by provision of an on-chip differential input amplifier, clock generator, and latched tri-state interface bus. Minimal external components required include a low-cost 3.579545 MHz color burst crystal ,a timing resistor, and a timing capacitor.

Pin Configuration



Block Diagram



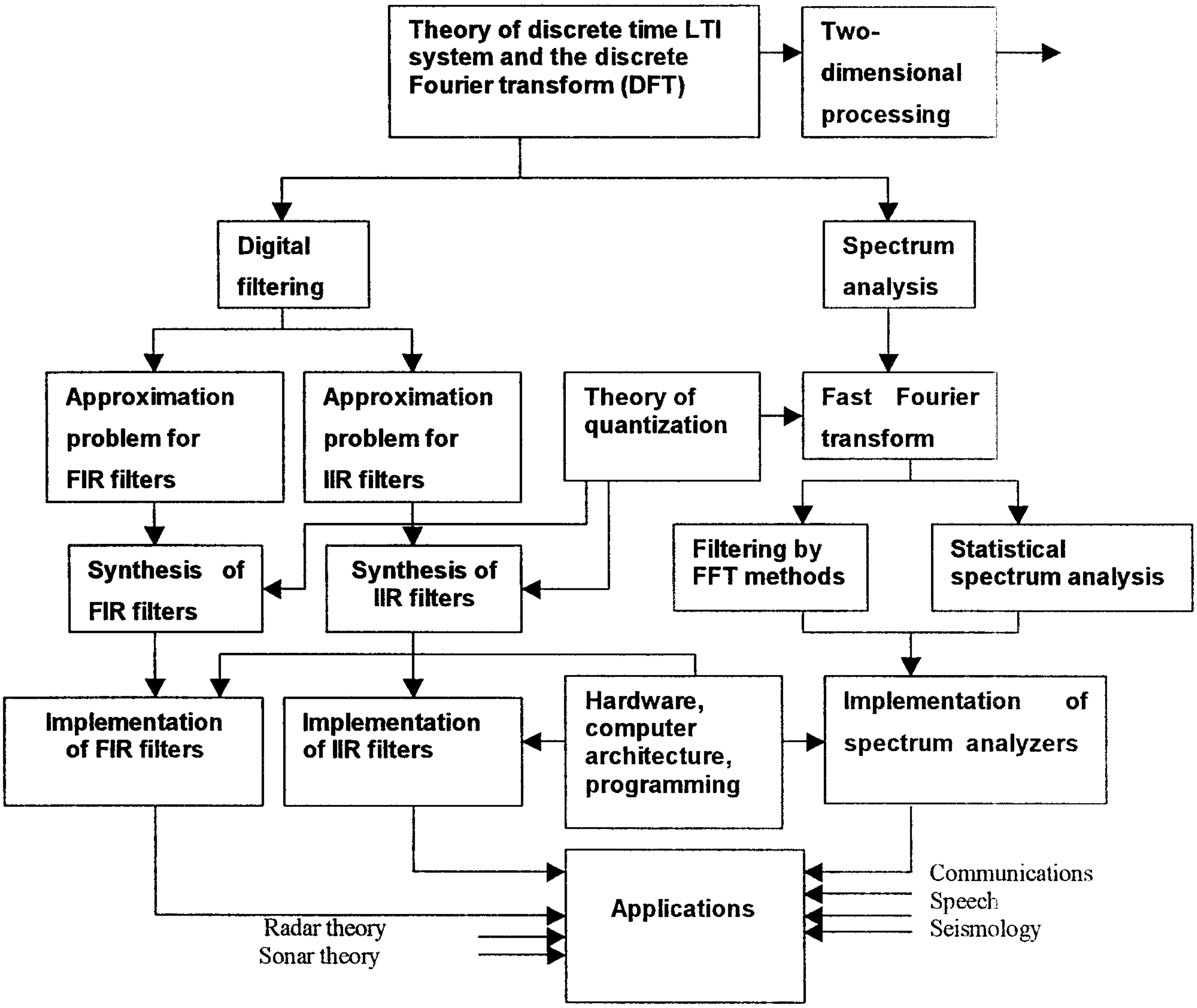
DIGITAL SIGNAL PROCESSING

The field of digital processing has grown enormously in the past decade to encompass and provide firm theoretical backgrounds for a large number of individual areas. Since digital signal processing relies on the theory of discrete-time linear time-invariant systems, we show this as a major unifying influence for the entire field.

The major subdivisions of the field of digital signal processing are digital filtering and spectrum analysis. The field of digital filtering is further divided into finite impulse response (FIR) filters and infinite impulse response (IIR) filters. The field of spectrum analysis is broken into calculation of spectra via discrete Fourier transform (DFT) and via statistical techniques as in the case of random signals e.g., quantization of noise in a digital system. The fast Fourier transform (FFT) and the related area of fast convolution are almost exclusively used in practical spectrum analysis techniques.

Two-dimensional signal processing is a relatively new area. The remaining aspects of digital signal processing, as in Fig are the important topics of implementations of digital systems and application areas. Almost all the spectrum analysers would be of little value, if a good understanding of the issues involved in practical implementations of such systems in finite precision software or hardware were available. Thus the area of implementation is first concerned with the mathematical basis of quantization in discrete systems then, depending on whether a software or hardware implementation is desired, it is important to understand the inherent limitations and advantages in both these implementations.

BLOCK DIAGRAM OF DSP


Theory of discrete time LTI system and the discrete Fourier transform (DFT)
Two-dimensional processing
Digital filtering
Spectrum analysis
Approximation problem for FIR filters
Approximation problem for IIR filters
Theory of quantization
Fast Fourier transform
Synthesis of FIR filters
Synthesis of IIR filters
Filtering by FFT methods
Statistical spectrum analysis
Implementation of FIR filters
Implementation of IIR filters
Hardware, computer architecture, programming
Implementation of spectrum analyzers
Applications
Radar theory
Sonar theory
Communications
Speech
Seismology

STEPPER MOTOR

A unique type of motor useful for moving things in small increments is the stepper motor. Instead of rotating smoothly as most motors do, stepper motors rotate on steps from one fixed position to the next. Common step sizes range from 0.9° to 30°. A stepper motor is stepped from one position to the next by changing the currents through the fields in the motor. The two common field connections are referred to as two phase and four phase.

In our project we are using four phase stepper motors. The buffers are inverting, a high on an output pin turns on current to a winding. The switching sequence to step a motor in a clockwise direction or a counter clockwise direction is shown below.

STEP	SW4	SW3	SW2	SW1
1	0	0	1	1
2	1	0	0	1
3	1	1	0	0
4	1	0	0	1
1	0	0	1	1

Consider the four phases as SW1, SW2, SW3 & SW4.If SW1 and SW2 are turned on. Turning off SW2 and turning on SW4 will cause the motor to rotate one step of 1.8° clockwise. Changing to SW4 and SW3 on will cause the motor to rotate another 1.8° clockwise. Changing to SW3 and SW2 on will cause another step. After that, changing to SW2 and SW1 on again will cause another step clockwise. We can repeat the sequence until the motor has rotated as many steps clockwise as we want. To step the motor counter clockwise, work through the switch sequence in the reverse direction. In either case the motor will be held in its last position by the current through the coils.

To take the first step clockwise from SW2 and SW1 being on, the pattern of 1's and 0's is simply rotated one bit position around to the right. The 1 from SW1 is rotated around into bit 4. To take the next step the switch pattern is rotated one more bit position. To step counter clockwise the switch pattern is rotated left one bit position for each step desired. Suppose 00110011 is loaded into ACC and output this to the switches. Duplicating the switch pattern in the upper half of ACC will make stepping easy. To step the motor clockwise, just rotate this pattern right one bit position and output it to the switches. To step counter clockwise, rotate the switch pattern left one bit position and output it. After output one-step code just waits for a few milliseconds before output another step command, because the motor can only step so fast. Maximum stepping rates for different types of steppers vary from a few hundred steps per second to several thousand steps per second.

To achieve high stepping rates the stepping rate is slowly increased to the maximum, then it is decreased as the desired number of steps is approached. To add a stepper motor to the robot add the clamp diodes across each winding to save the transistors from inductive kick and the current limiting resistors (R1 & R2). The motor we used here has a nominal voltage rating of 5.5V. This means that we could have designed the circuit to operate with a voltage of about 6.5V on the emitters of the driver transistors. For low stepping rates, this would work fine. However for higher stepping rates and more torque while stepping, we use a higher supply voltage and current-limiting resistors R1 and R2.

The motor we used here has a nominal voltage rating of 5.5V. This means we could have designed the circuit to operate with voltage of about 6.5V on the emitters of the driver transistors (5.5V for the motor plus 1V for the drop across the transistor) for low stepping rates, this would work fine. However, for higher stepping rates and more torque while stepping, we use a higher supply voltage and current -limiting resisters as shown. The point of this is that by adding series resistance, we decrease L/R time constant.

This allows the current to change more rapidly in the windings. For the motors we used, the current per winding is 0.88A. Since only one winding on each resistor is ever on at a time, 6.5V/0.88A gives a resistor value of 6.25Ω. To be conservative we used 8Ω, 10-W resistors. The optional transistor switch and diode connection to the +5V supply are used as follows. When the motor is not stepping, the switch to +12V is off so the motor is held in position by the current from the +5V supply. Before we send a step command, we turn on the transistor to +12V to give the motor more current for stepping. When stepping is done we turn off the switch to +12V, and drop back to the +5V supply which cuts the power dissipation.

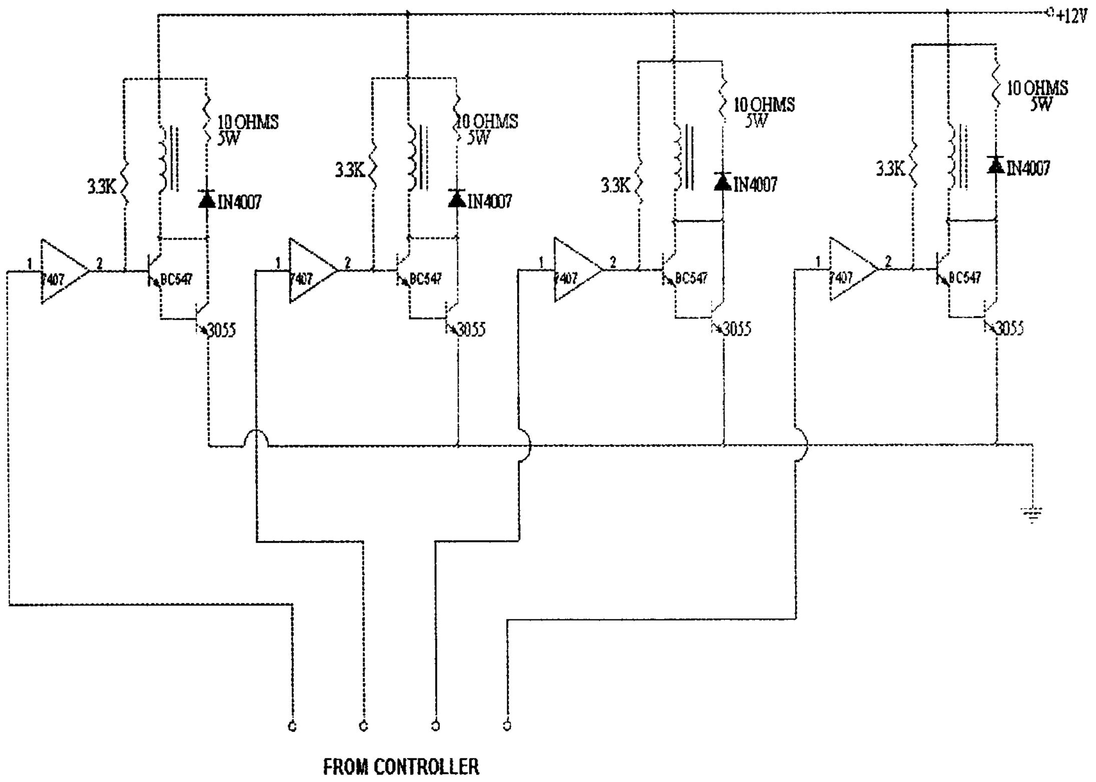
STEPPER MOTOR INTERFACING

Stepper motor has four energizing coils. When energizing in alternate phases of stepper motor, the shaft of Stepper motor starts running in forward Direction and when energized in opposite phases, the motor starts running in reverse direction.

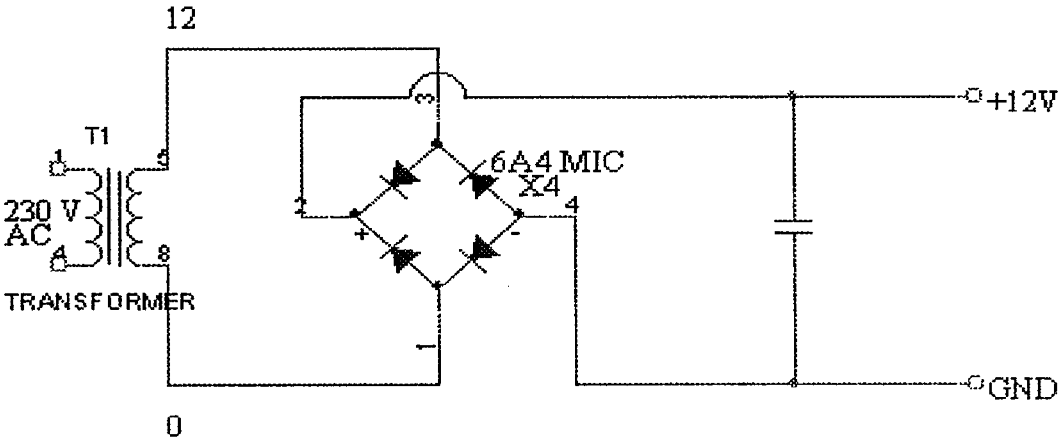
Four coils of stepper motor are energized by using ports of the processor. We cannot directly drive the coils of the stepper motor, so by using NPN transistor, power transistor, we energize the coils of the stepper motor. The NPN transistors are switched by sending pulse from the processor.

Power transistor is switched by using NPN transistor. Resistors are used for current limiting purpose. Diodes are used for avoiding reverse current. When sending corresponding data for energizing the coils of the stepper motor from the processor, first NPN transistors are switched, then we can switch on the Power transistors. By switching power transistors we can energize the coils of the stepper motor.

STEPPER MOTOR INTERFACING CIRCUIT



STEPPER MOTOR POWER SUPPLY



RELAYS

A relay is a switch operated by an electromagnet. It is used to switch ON a circuit to control another circuit containing devices such as lamps or electric motors which requires high current, or to operate several switch contacts simultaneously.

When the controlling current flows through the coil, the soft iron core is magnetizes and attracts the L-shaped soft iron armature. This rocks on its pivot and changes over the electrical contacts in the circuit being controlled.

The current needed to operate a relay is called the pull-in current. The dropout current is the current in the coil when the relay just stops working. If the coil resistance R of a relay is 185 Ω and its operating voltage V is 12V, then the pull-in current I is given by:

$$I = \frac{V}{R} = \frac{12}{185} = 0.065\text{A} = 65\text{mA}$$

PCB DESIGNING

PCB DESIGNING

Design and Fabrication of Printed circuit boards

Introduction

Printed circuit boards, or PCBs, forms the core of an electronic equipment both in domestic and industrial applications. Some of the areas where PCBs are intensively used are computers, process control, telecommunications and instrumentation.

Manufacturing

The manufacturing process consists of two methods: print and etch, plate and etch. The single sided PCBs are usually made using the print and etch method. The double sided plate through hole (PTH) boards are made by the print plate and etch method. The production of multi layer boards uses both these methods. The inner layers are printed and etch while the outer layers are produced by plate and etch after pressing the inner layers.

Sofware

The software used in our project to obtain the schematic layout is MICROSIM.

Panelisation

Here the schematic is transformed in to working positive/negative films. The circuit is repeated conveniently to accommodate economically as many circuits as possible in a panel, which can be operated in every sequence of subsequent steps in the PCB process. This is called panalisation. For the PTH boards, the next operation is drilling.

Drilling

PCB drilling is a state of the art operation. Very small holes are drilled with high speed CNC drilling machines, giving a wall finish or no smear or epoxy finish.

Plating

It is the heart of PCB manufacturing process. The holes drilled in the board are treated both mechanically and chemically before depositing copper by electrolysis copper platting process.

Etching

Once a multiplayer board is drilled and electrolysis copper is deposited, the image available in the form of a film is transferred on to the out side by photo printing using a dry film printing process. The boards are then electrolytic plated on to the circuit pattern with copper and tin. The tin-plated deposit serves as an etch resistance when copper in the unwanted area is removed by the conveyorised spray etching machine with chemical etchants. The etching machines are attached to an automatic dosing equipment, which analyses and controls etchants concentrations.

Soldermask

As, the PCB design may call for very close spacing between conductors, a solder mask has to be applied on both sides of the circuitry to avoid bridging of conductors. The solder mask ink is applied by screening. The ink is dried, exposed to UV, developed in a mild alkaline solution and finally cured by both UV and thermal energy.

Hot Air Levelling

After applying the solder mask, the circuit pads are soldered using the hot air levelling process. The bare bodies fluxed and dipped into a molten solder bath. While removing the board from the solder bath, hot air is blown on both sides of the board through air knives in the machine, leaving the board soldered and levelled. This is one of the common finishes given to the boards. Thus the double side is plated through and the printed circuit board is manufactured .Now it is ready to solder the components.

TEST RESULTS

TEST RESULTS

The Robotic Arm was tested in the laboratory using direct cable transmission for a variable distance of about 50 Feet and the arm was controlled precisely.

The Robotic Arm was then tested using RF transmission for a variable distance within 50 Feet and still was found to give a good performance.

The Vehicle control using Computer parallel port interface was tested by changing the direction and the speed of movement and found to give an accurate control.

CONCLUSION

CONCLUSION

We here by conclude that the Controllers designed by us will be useful for various applications as listed below:

- It canbe used as a model controller for manufacturing robots.
- It canbe used as a Pick and Place Robot.
- It canbe used as a Load shifting vehicle.
- It canbe used as a Material Handler in Laboratories.
- It canbe used as an exploration vehicle for rough terrains.

Scope for Future Development

- The Arm can be made more flexible by attaching an elbow setup.
- The computer interface can also be made as a RF Controlled one.
- The Clamp capacity can be increased to lift more load.
- The power supply can be made as a battery powered one for mobility.
- Higher capacity motors can be used in case of heavy duty purpose.

REFERENCES

REFERENCES

[1]P. Vinciarelli, “DSP Filters” *U.S. Patent # 4,415,959 (1993).*

[2]K. H. Liu and F. C. Lee, “Noise elimination in RF”*International TelecommunicationsEnergy Conference; New Orleans, 1994.*

[3]K. H. Lieu, R. Oruganti, F. C. Lee, “Intelligent systems,”*IEEE PESC 1995 (France).*

[4]A. Heyman, “Exploration Vehicles hard binding”*IEEE 1997.*

[5]M. Schlect, L. Casey, “DSP Applications, *"IEEE APEC Conference, 1997.*

[6]TMS320C25 Handbook by Texas Instruments.

[7]Lampex LCD Manual.

[8]PCB Design and Technology by Misra