LAMPIRAN A

Listing Program

LISTING PROGRAM PADA MIKROKONTROLER ATMEGA32
(PENAMPIL SIGNAL)
*PROGRAM UTAMA

```
#include <avr/io.h>
#include <avr/interrupt.h>
#include "glcd.c"

void USART_Init(unsigned int baudrate);
void timePause(unsigned long count);
void ADC_init (void);
void initTimer (void);

#ifndef F_CPU
#define F_CPU 16000000UL
#endif

#define msUp            1
#define msDwn           4
#define YposUp          0
#define YposDwn         3
#define freeze          2

#define AC              0
#define DC              1
#define SQUARE          2
#define TRUE            0
#define FALSE           1

unsigned int counter = 0;
unsigned int dataCounter = 0;
unsigned int timeDiv = 52;
unsigned int analogInput = 0;
unsigned char trigger = 2;
unsigned char cnt = 0;
unsigned char empty = 0;
unsigned int out = 0;
unsigned char findZero = 0;
unsigned char pressedButton = 0;
unsigned char upLimit = 0;
unsigned char dnLimit = 255;
unsigned char limitBkup = 0;
unsigned char tipeTeg = AC;
unsigned char complete = TRUE;

signed char Ypos   = 0;
signed char Ypos2   = 0;
signed char position  = 0;
```

```
int main (void)
{
        DDRC = 0b00000000;
        PORTC = 0b11111111;

        DDRA = 0b00000000;

        unsigned char temp1;
        unsigned int temp2;
        unsigned char i;

        glcdInit();
        ADC_init();
        createWelcomeScreen();
        showTheWave();
        timePause(10000000);

        for(;;)
        {
        //---------


        if (pressedButton == 0)
                        {
                if (~PINC & (1<<msUp) && (timeDiv <= 1000))
                                {
                                        if(timeDiv == 0)
                                                timeDiv = 52;
                                        else
                                                timeDiv += 145;
                                                pressedButton = 1;
                                }

                if (~PINC & (1<<msDwn) && (timeDiv >= 52))
                                {
                                        if(timeDiv <= 145)
                                                timeDiv = 0;
                                        else
                                                timeDiv -= 145;
                                                pressedButton = 1;
                                }

                if (~PINC & (1<<YposUp) && (Ypos2 <= 60))
                                Ypos2++;

                if (~PINC & (1<<YposDwn) && (Ypos2 >= -60))
                                Ypos2--;

                if (~PINC & (1<<freeze))
```

```
                        while (~PINC & (1<<freeze));

                }
        else
                {
                        temp1 = PINC;
                        if (temp1 == 255)
                                pressedButton = 0;
                        }

//--------v

dataCounter = 0;
findZero = 0;
upLimit = 0;
dnLimit = 255;


        for (i=99; i>0; i--)
                {
                        ADCSRA |= (1 << ADSC);
                        loop_until_bit_is_set(ADCSRA, ADIF);
                        temp1 = ADCL;
                        temp2 = ADCH;
                        timePause(timeDiv);

                        if (upLimit < temp2)
                                upLimit = temp2;

                        if (dnLimit > temp2)
                                dnLimit = temp2;

                        if (temp2 > 0)
                        {
                                temp2 += 5;
                                temp2 /= 5;
                                temp2 += 2;
                        }else temp2 = 2;

                        position = temp2 + Ypos2 +5;
                        if (position <= 63 && position >= 0
                                fillDataLcdBuffer(i,position);
                        else
                                fillDataLcdBuffer(i,0);

                }
                        if(upLimit != dnLimit)
                        trigger = (((upLimit - dnLimit)/2)+ dnLimit);
//--------^
```

```
createRaster();
createWave();
showTheWave();


                        dataCounter = 0;
                        do
                        {
                                limitBkup = temp2;

                                ADCSRA |= (1 << ADSC);
                                loop_until_bit_is_set(ADCSRA, ADIF);
                                temp1 = ADCL;
                                temp2 = ADCH;
                                //timerPause(timeDiv);

                                if(limitBkup == temp2)
                                {
                                        dataCounter++;
                                        if(dataCounter >= 500)
                                                tipeTeg = DC;
                                        else
                                                tipeTeg = AC;
                                }

                                if((tipeTeg == AC) && ((temp2 == trigger) &&
(limitBkup < temp2)))
                                                complete = TRUE;
                                else
                                if((tipeTeg == DC) && (limitBkup == temp2) &&
(upLimit != dnLimit))
                                {
                                        dataCounter = 0;
                                        do
                                        {
                                                ADCSRA |= (1 << ADSC);
                                                loop_until_bit_is_set(ADCSRA, ADIF);
                                                temp1 = ADCL;
                                                temp2 = ADCH;
                                                complete = TRUE;
                                                dataCounter++;
                                        }while ((temp2 > trigger) && (dataCounter
<1000));

                                        dataCounter = 0;
                                        do
                                        {
                                                ADCSRA |= (1 << ADSC);
                                                loop_until_bit_is_set(ADCSRA, ADIF);
```

```
                                        temp1 = ADCL;
                                        temp2 = ADCH;
                                        complete = TRUE;
                                        dataCounter++;
                                }while  ((temp2  <  trigger)  &&  (dataCounter
<1000));
                                }
                        else
                                if((tipeTeg == DC) && (limitBkup == temp2))
                                        complete = TRUE;
                                else
                                        complete = FALSE;


                } while(complete == FALSE);

        }
}

//================================================================
//
//================================================================

void ADC_init (void)
{
        ADMUX = 0b01100000;
        ADCSRA = 0b11000100;
}


//================================================================
//
//================================================================
void timePause(unsigned long count)
{
   while(count--);
}

//================================================================
//
//================================================================
```

***SUB PROGRAM GLCD.C**

```
#include "glcd.h"
#include <util/delay.h>


void glcdInit (void)
{
 data_port_pins = 0;  //DATA Port Low
 data_port_ddr = 0xff; //Make DATA port output

 ctrl_port = 0;   //CONTROL Port Low
 ctrl_port_ddr = 0xff; //Make CONTROL port outputs

 ctrl_port |= (1<<lcdrst);
 ctrl_port |= (1<<lcdcs1);   //Enable the CS1 of the display
      ctrl_port |= (1<<lcdcs2);   //Enable the CS2 of the display
      ctrl_port &= ~(1<<lcdrs);        //Clear RS  \_
      ctrl_port &= ~(1<<lcdrw);        //Clear RW/  Command mode

 //delayTimexx(10000);
 _delay_us(42);
 data_port = 0b11000000;   //Display start line = 0 (0-63)
      eStrobe();


 //delayTimexx(10000);
 _delay_us(42);
      data_port = 0b01000000;   //Set address = 0 (0-63)
      eStrobe();

 //delayTimexx(10000);
 _delay_us(42);
      data_port = 0b10111000;   //Set page = 0 (0-7)
      eStrobe();

 _delay_us(42);
 //delayTimexx(10000);
      data_port = 0b00111111;   //Display ON
      eStrobe();

      ctrl_port &= ~(1<<lcdcs1);  //Disable the CS1 of display
      ctrl_port &= ~(1<<lcdcs2);  //Disable the CS2 of display

 createRaster(); // raster...
 showTheWave(); // LCD.

}
```

```
//==================================================================
//  create "welcome" screen
//==================================================================
void createWelcomeScreen (void)
{
unsigned int size;


 lcdAddress = 0;
 flashAddress =0;
 const char *data = welcomeScreen;

 for (size = 0; size<1024; size++)
   lcdBuffer[lcdAddress++] = pgm_read_byte(data++);

}
//==================================================================
// Copy the 1024 bytes from flash to RAM.
//==================================================================
void createRaster (void)
{
 unsigned int size;


 lcdAddress = 0;
 flashAddress =0;
 const char *data = LcdRaster;

 for (size = 0; size<1024; size++)
  lcdBuffer[lcdAddress++] = pgm_read_byte(data++);

}
//==================================================================
//
//==================================================================
void changeLine (unsigned char lineData)
{
  ctrl_port |= (1<<lcdcs1); //Enable the CS1 of the display
        ctrl_port |= (1<<lcdcs2); //Enable the CS2 of the display
        ctrl_port &= ~(1<<lcdrs);   //Clear RS. Command mode
  ctrl_port &= ~(1<<lcdrw);   //Clear RW. Command mode

  lineData += 0b10111000;
        data_port = lineData;  //Set page = 0 (0-7)
        eStrobe();

        data_port = 0b01000000;  //Set address = 0 (0-63)
        eStrobe();

  ctrl_port |= (1<<lcdrs); //Set RS. Data mode
```

```
}

//============================================================
//
//============================================================

void fillDataLcdBuffer (unsigned char address, unsigned char data)
{

 dataLcdBuffer[address] = data;
}

//============================================================
//
//============================================================
void showTheWave (void)
{
 for(lcdAddress = 0; lcdAddress < 1024 ;lcdAddress++)
 {

  if (line == 8)
   line = 0;

  if (column == 128)
  {
   column = 0;
   line++;
   if (line == 8)
    line = 0;
   changeLine(line);
  }

  if (column <= 63)
   enable_cs1();

  if (column == 64)
   enable_cs2();

  //delayTimexx(10);
   _delay_us(3);
  ctrl_port |= (1<<lcdrs); // "DATA SEND" mode
  ctrl_port &= ~(1<<lcdrw);
  data_port = lcdBuffer[lcdAddress];
  eStrobe();

  column++;     // increase column (maximum 128).
 }

}
```

```
//==============================================================
// Write the data for 1 waveform to the buffer dataLcdBuffer.
//==============================================================
void createWave (void)
{
 unsigned char data;
 unsigned char byte;
 unsigned char i = 0;

 for (i=0; i<128; i++)
 {
  byte = 0b10000000;
  lcdAddress = 996;

  if(i<100)
        {
   data = dataLcdBuffer[i];
   for (; data>7; data-=8)
    lcdAddress -= 128;

   lcdAddress -= i;

   for(; data>0; data--)
    byte >>= 1;
   lcdBuffer[lcdAddress] |= byte;
  }
 }
}

//==============================================================
//
//==============================================================
void enable_cs1 (void)
{
 ctrl_port |= (1<<lcdcs1);    //Enable the CS1 of the display
 ctrl_port &= ~(1<<lcdcs2);    //Disable the CS2 of the display
}

//==============================================================
//
//==============================================================
void enable_cs2 (void)
{
 ctrl_port |= (1<<lcdcs2);    //Enable the CS2 of the display
 ctrl_port &= ~(1<<lcdcs1);    //Disable the CS1 of the display
}
```

```
//==============================================================
//  Wait for graphics LCD to be unbusy
//==============================================================
void glcdWait (void)
{
 unsigned char dataIn;

 data_port_ddr = 0;   //Make portB all inputs
 ctrl_port |= (1<<lcdrw); //Set r/w pin to read
 ctrl_port &= ~(1<<lcdrs); //Set register select to command

 do
 {
  eStrobe();
  dataIn = data_port_pins; //Read busy flag
 }while bit_is_set(dataIn, 7); // loop until bit7 of 'dataIn' will be cleared.

 data_port_ddr = 0xff;  //Make portB all outputs
 ctrl_port &= ~(1<<lcdrw); //Set r/w pin to write
 ctrl_port |= (1<<lcdrs); //Set register select to data


}
//==============================================================
//
//==============================================================
void eStrobe (void)
{
	ctrl_port |= (1<<lcde);   //lcd 'E' pin high
  //delayTimexx(10);
   _delay_us(3);
	ctrl_port &= ~(1<<lcde);  //lcd 'E' pin low
  //delayTimexx(10);
   _delay_us(3);
}

//==============================================================
```

***SUB PROGRAM GLCD.H**

```
#define lcdrs     0             // LCD's RS pin is connected to Pin0 of AVR
#define lcdrw     1             // LCD's r/w pin is connected to Pin1 of AVR
#define lcde      2     // LCD's e pin is connected to Pin2 of AVR
#define lcdcs1    5     // LCD's CS1 pin is connected to Pin5 of AVR
#define lcdcs2    4     // LCD's CS2 pin is connected to Pin4 of AVR
#define lcdrst    3     // LCD's RST pin is connected to Pin3 of AVR


#define ctrl_port           PORTB
#define ctrl_port_ddr   DDRB
#define ctrl_port_pins  PINB

#define data_port           PORTD
#define data_port_ddr   DDRD
#define data_port_pins  PIND
#define pgm_read_byte

void glcdInit (void);
void createWelcomeScreen (void);
void createRaster (void);
void delayTime (unsigned long counter);
void glcdWait (void);
void eStrobe (void);
void enable_cs1 (void);
void enable_cs2 (void);
void showTheWave (void);
void changeLine (unsigned char data);
void createWave (void);
void fillDataLcdBuffer (unsigned char address, unsigned char data);


unsigned char column = 0;
unsigned char line = 0;
unsigned char lcdBuffer[1024];
unsigned int  lcdAddress = 0;
unsigned int  flashAddress = 0;
static unsigned char dataLcdBuffer[128];
unsigned int  backupLcdAddress = 0;
```

```
const char LcdRaster[] __attribute__ ((progmem)) = {
255,1,1,1,1,1,1,1,1,3,1,1,1,1,1,1,
1,1,1,3,1,1,1,1,1,1,1,1,1,3,1,1,
1,1,1,1,1,1,1,3,1,1,1,1,1,1,1,1,
1,171,1,1,1,1,1,1,1,1,1,3,1,1,1,1,
1,1,1,1,1,3,1,1,1,1,1,1,1,1,1,3,
1,1,1,1,1,1,1,1,1,3,1,1,1,1,1,1,
0,0,0,0,255,0,0,60,66,66,60,0,64,0,68,98
82,76,0,60,74,74,48,0,112,8,112,8,112,0,0,0
255,0,0,0,0,0,0,0,0,8,0,0,0,0,0,0,
0,0,0,8,0,0,0,0,0,0,0,0,0,8,0,0,
0,0,0,0,0,0,0,8,0,0,0,0,0,0,0,0,
8,170,8,0,0,0,0,0,0,0,0,8,0,0,0,0,
0,0,0,0,0,8,0,0,0,0,0,0,0,0,0,8,
0,0,0,0,0,0,0,0,0,8,0,0,0,0,0,0,
0,0,0,0,255,0,0,30,33,161,30,0,32,0,39,165
165,25,0,0,0,0,0,0,56,4,56,4,56,0,0,0
255,0,0,0,0,0,0,0,0,32,0,0,0,0,0,0,
0,0,0,32,0,0,0,0,0,0,0,0,0,32,0,0,
0,0,0,0,0,0,0,32,0,0,0,0,0,0,0,0,
32,170,32,0,0,0,0,0,0,0,0,32,0,0,0,0,
0,0,0,0,0,32,0,0,0,0,0,0,0,0,0,32,
0,0,0,0,0,0,0,0,0,32,0,0,0,0,0,0,
0,0,0,0,255,0,0,0,145,223,16,0,16,0,207,80
80,79,0,0,0,0,0,0,28,2,28,2,28,0,0,0
255,128,0,128,0,128,0,128,0,192,0,128,0,128,0,128,
0,128,0,192,0,128,0,128,0,128,0,128,0,192,0,128,
0,128,0,128,0,128,0,192,0,128,0,128,0,128,0,128,
128,234,128,128,0,128,0,128,0,128,0,192,0,128,0,128,
0,128,0,128,0,192,0,128,0,128,0,128,0,128,0,192,
0,128,0,128,0,128,0,128,0,192,0,128,0,128,0,128,
0,0,0,0,255,0,0,64,40,47,200,0,8,0,201,41
41,198,0,0,0,0,0,0,14,129,14,129,14,0,0,0
255,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,
0,0,0,1,0,0,0,0,0,0,0,0,0,1,0,0,
0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,
0,171,0,0,0,0,0,0,0,0,0,1,0,0,0,0,
0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,1,
0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,
0,0,0,0,255,0,0,36,22,149,100,0,4,0,115,84
84,147,0,0,0,0,0,0,135,64,135,64,135,0,0,0
255,0,0,0,0,0,0,0,0,2,0,0,0,0,0,0,
0,0,0,2,0,0,0,0,0,0,0,0,0,2,0,0,
0,0,0,0,0,0,0,2,0,0,0,0,0,0,0,0,
2,170,2,0,0,0,0,0,0,0,0,2,0,0,0,0,
0,0,0,0,0,2,0,0,0,0,0,0,0,0,0,2,
0,0,0,0,0,0,0,0,0,2,0,0,0,0,0,0,
0,0,0,0,255,0,0,10,43,90,138,0,2,0,242,10
10,241,0,0,0,0,0,0,195,32,195,32,195,0,0,0
255,0,0,0,0,0,0,0,0,8,0,0,0,0,0,0,
```

```
0,0,0,8,0,0,0,0,0,0,0,0,0,8,0,0,
0,0,0,0,0,0,0,8,0,0,0,0,0,0,0,0,
8,170,8,0,0,0,0,0,0,0,0,8,0,0,0,0,
0,0,0,0,0,8,0,0,0,0,0,0,0,0,0,8,
0,0,0,0,0,0,0,0,0,8,0,0,0,0,0,0,
0,0,0,0,255,0,0,133,149,173,68,0,129,0,156,149
149,100,0,0,0,0,0,0,225,16,225,16,225,0,0,0
255,128,128,128,128,128,128,128,128,160,128,128,128,128,128,128,
128,128,128,160,128,128,128,128,128,128,128,128,128,160,128,128,
128,128,128,128,128,128,128,160,128,128,128,128,128,128,128,128,
160,170,160,128,128,128,128,128,128,128,128,160,128,128,128,128,
128,128,128,128,128,160,128,128,128,128,128,128,128,128,128,160,
128,128,128,128,128,128,128,128,128,160,128,128,128,128,128,128,
0,0,0,0,255,0,0,24,30,16,126,0,64,0,60,66
66,60,0,0,0,0,0,0,112,8,112,8,112,0,0,0};


const char welcomeScreen[] __attribute__ ((progmem)) = {
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,255,1,6,8,6,1,255,0,0,1,2
,252,2,1,0,0,0,0,0,0,126,129,129,129,126,0,70
,137,137,137,114,0,126,129,129,129,102,0,129,255,129,0,255
,128,128,128,0,255,128,128,128,0,126,129,129,129,126,0,70
,137,137,137,114,0,126,129,129,129,102,0,126,129,129,129,126
,0,255,17,17,17,14,0,255,137,137,129,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,255,137,137,137,129,0,255
,128,128,128,128,0,255,137,137,137,129,0,255,8,20,34,193
,0,1,1,255,1,1,0,255,17,49,81,142,0,126,129,129
,129,126,0,0,0,0,0,255,2,4,2,255,0,252,34,33
,34,252,0,255,17,49,81,142,0,252,34,33,34,252,0,255
,2,12,48,255,0,252,34,33,34,252,0,1,1,255,1,1
,0,255,8,8,8,255,0,252,18,17,18,252,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
```

,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,252,254,3,3,3,3,3,254
,252,0,0,0,192,240,60,15,3,3,255,255,0,0,60,62
,3,3,131,195,195,126,60,0,60,62,3,3,131,195,195,126
,60,0,4,254,255,0,0,60,62,3,3,131,195,195,126,60
,0,60,126,195,195,195,195,254,252,0,0,0,0,0,0,0
,0,0,0,0,255,192,160,144,136,136,144,160,192,128,128,128
,128,128,128,128,128,192,160,144,136,136,144,160,192,128,128,128
,128,128,128,128,128,128,128,128,128,128,128,0,0,0,0,0
,0,0,0,0,0,0,0,0,15,31,48,48,48,48,48,31
,15,0,2,3,3,3,3,3,3,3,63,63,0,0,56,60
,54,51,49,48,48,56,56,0,56,60,54,51,49,48,48,56
,56,0,48,63,63,48,0,56,60,54,51,49,48,48,56,56
,0,12,24,48,48,48,48,31,15,0,0,0,0,0,0,0
,0,0,0,0,127,0,0,0,0,0,0,0,0,1,2,4
,8,8,4,2,1,0,0,0,0,0,0,0,0,1,2,4
,8,8,4,2,1,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0};

LISTING PROGRAM PADA MIKROKONTROLER ATMEGA16 (PENAMPIL ANGKA)

```
Project :
Version :
Date    : 5/19/2009
Author  : F4CG
Company : F4CG
Comments:


Chip type           : ATmega16
Program type        : Application
Clock frequency     : 16.000000 MHz
Memory model        : Small
External SRAM size  : 0
Data Stack size     : 256
*****************************************************/

#include <mega16.h>
#include <delay.h>
#include <stdio.h>
// Alphanumeric LCD Module functions
#asm
   .equ __lcd_port=0x15 ;PORTC
#endasm
#include <lcd.h>
unsigned long int a,frek;
int text[33];



// External Interrupt 0 service routine
interrupt [EXT_INT0] void ext_int0_isr(void)
```

```
{a++;
// Place your code here

}

// Timer 1 output compare A interrupt service routine
interrupt [TIM1_COMPA] void timer1_compa_isr(void)
{ frek=a;
a=0;
// Place your code here

}

#define ADC_VREF_TYPE 0xC0

// Read the AD conversion result
unsigned int read_adc(unsigned char adc_input)
{
ADMUX=adc_input | (ADC_VREF_TYPE & 0xff);
// Start the AD conversion
ADCSRA|=0x40;
// Wait for the AD conversion to complete
while ((ADCSRA & 0x10)==0);
ADCSRA|=0x10;
return ADCW;
}

// Declare your global variables here
 float v2;

 float v;
 int adc;
 int adc2;
```

```
void main(void)
{
// Declare your local variables her
// Input/Output Ports initialization
// Port A initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T
PORTA=0x00;
DDRA=0x00;

// Port B initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T
PORTB=0x00;
DDRB=0x00;

// Port C initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T
PORTC=0x00;
DDRC=0x00;

// Port D initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T
PORTD=0x00;
DDRD=0x00;

// Timer/Counter 0 initialization
// Clock source: System Clock
// Clock value: Timer 0 Stopped
// Mode: Normal top=FFh
// OC0 output: Disconnected
TCCR0=0x00;
```

```
TCNT0=0x00;
OCR0=0x00;

// Timer/Counter 1 initialization
// Clock source: System Clock
// Clock value: 15.625 kHz
// Mode: CTC top=OCR1A
// OC1A output: Discon.
// OC1B output: Discon.
// Noise Canceler: Off
// Input Capture on Falling Edge
// Timer 1 Overflow Interrupt: Off
// Input Capture Interrupt: Off
// Compare A Match Interrupt: On
// Compare B Match Interrupt: Off
TCCR1A=0x00;
TCCR1B=0x0D;
TCNT1H=0x00;
TCNT1L=0x00;
ICR1H=0x00;
ICR1L=0x00;
OCR1AH=0x3D;
OCR1AL=0x09;
OCR1BH=0x00;
OCR1BL=0x00;

// Timer/Counter 2 initialization
// Clock source: System Clock
// Clock value: Timer 2 Stopped
// Mode: Normal top=FFh
// OC2 output: Disconnected
ASSR=0x00;
TCCR2=0x00;
TCNT2=0x00;
```

```
OCR2=0x00;

// External Interrupt(s) initialization
// INT0: On
// INT0 Mode: Falling Edge
// INT1: Off
// INT2: Off
GICR|=0x40;
MCUCR=0x02;
MCUCSR=0x00;
GIFR=0x40;

// Timer(s)/Counter(s) Interrupt(s) initialization
TIMSK=0x10;

// Analog Comparator initialization
// Analog Comparator: Off
// Analog Comparator Input Capture by Timer/Counter 1: Off
ACSR=0x80;
SFIOR=0x00;

// ADC initialization
// ADC Clock frequency: 1000.000 kHz
// ADC Voltage Reference: Int., cap. on AREF
// ADC Auto Trigger Source: None
ADMUX=ADC_VREF_TYPE & 0xff;
ADCSRA=0x84;

// LCD module initialization
lcd_init(16);

// Global enable interrupts
#asm("sei")
```

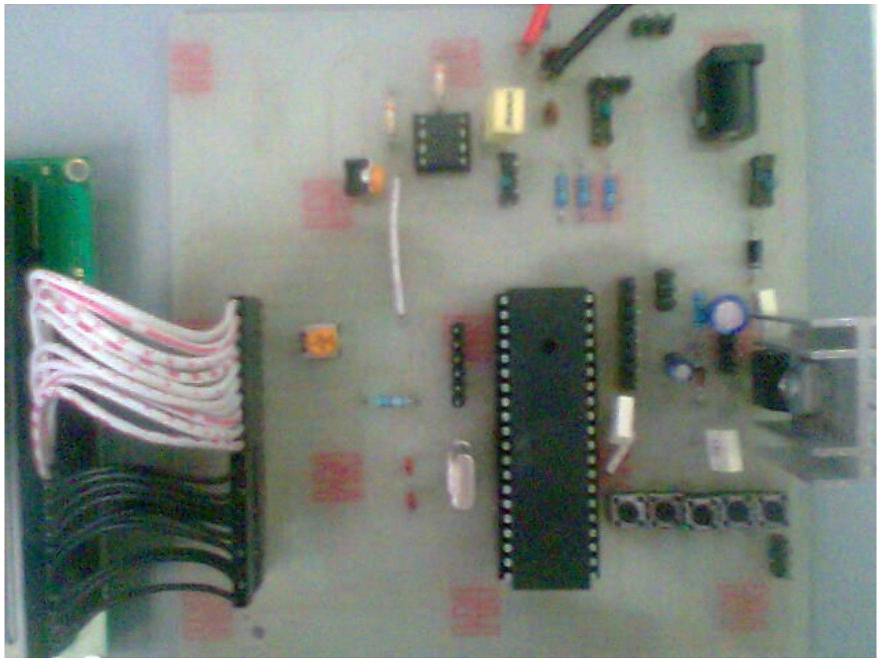
```
while (1)
      {


       adc=read_adc(1);
      if(frek>=5) v=((float)adc*(5/1.075)/1024);
             else  v=((float)adc*5/1024);
       frek=(frek/1.45);
       sprintf(text,"frek = %li \n V = %0.3f V ",frek,v);
       lcd_puts(text);
       delay_ms(1000);
       lcd_clear();


      };
}
```

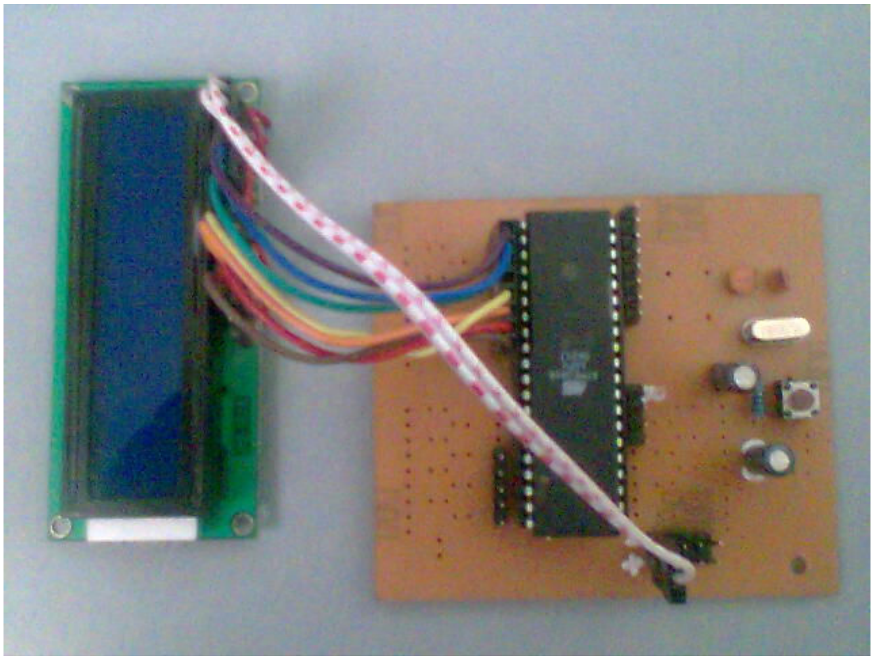
LAMPIRAN B

- Penampil Sinyal Tampak Atas

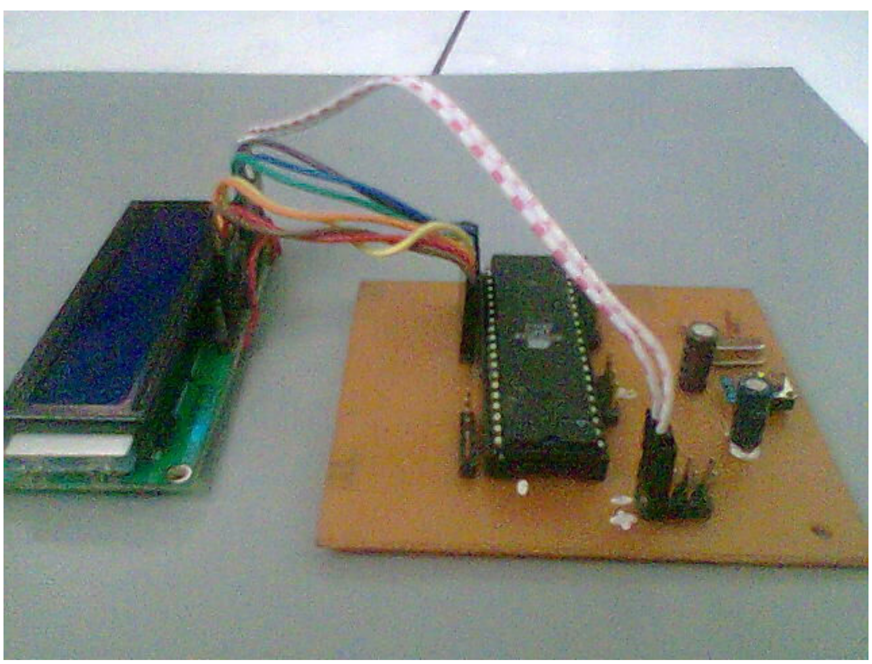


- Penampil Sinyal Tampak Samping

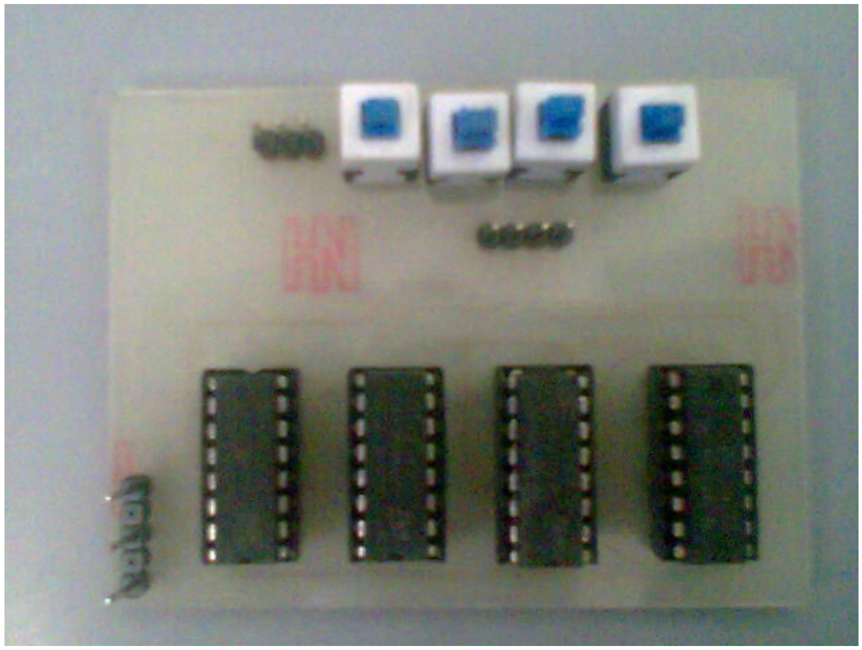
- Penampil Angka Tampak Atas



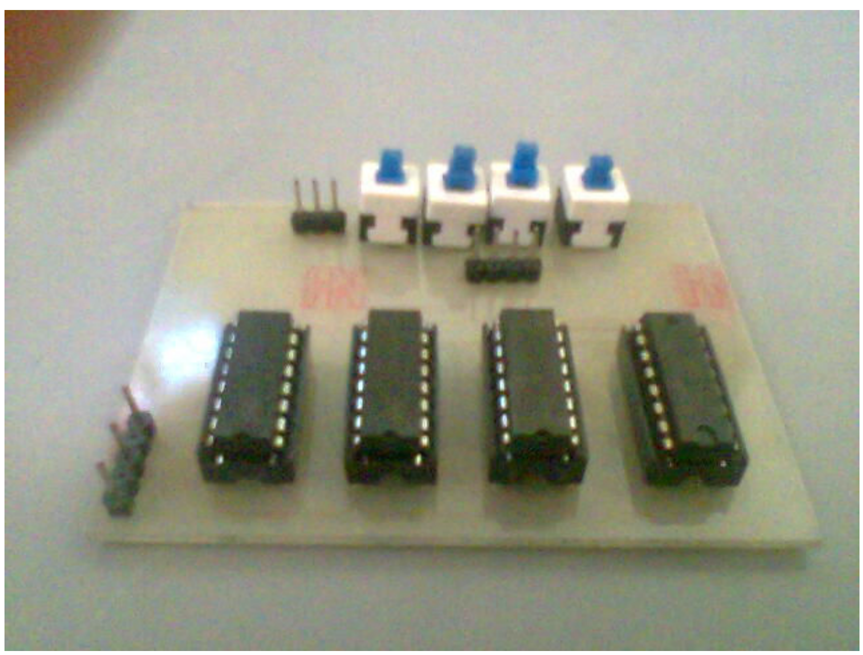
- Penampil Angka Tampak Samping



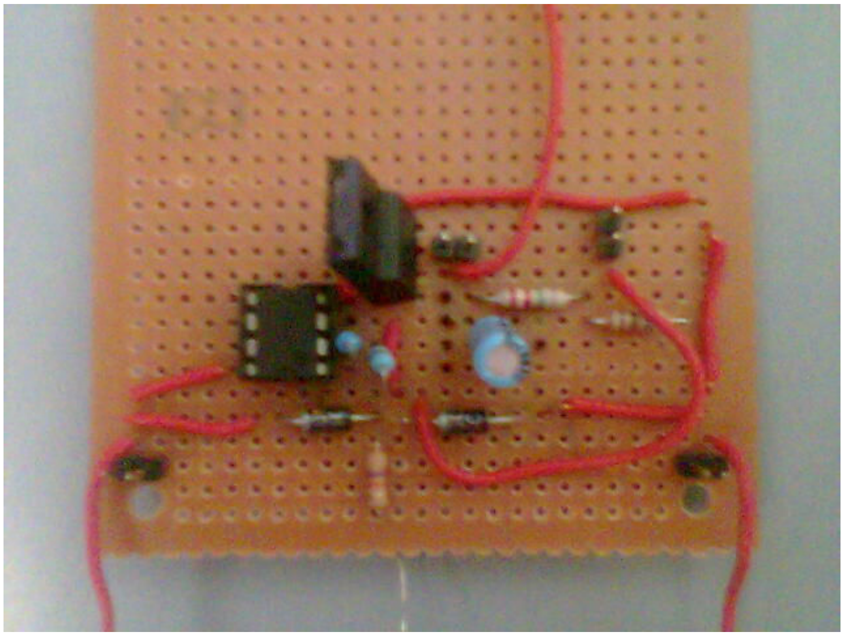
- Pemabagi Frekuensi Tampak Atas



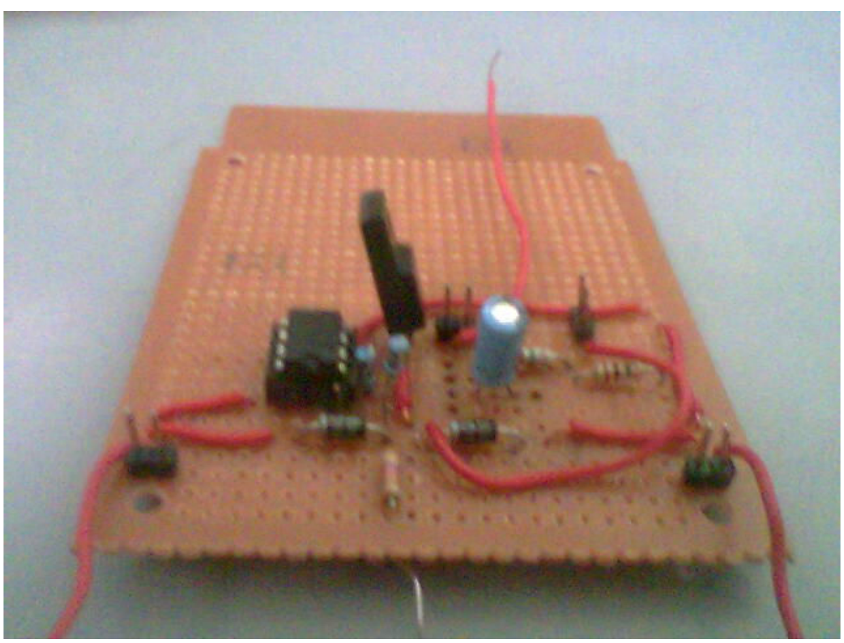
- Pembagi Frekuensi Tampak Samping



- Penyearah Tampak Atas



- Penyearah Tampak Samping



LAMPIRAN C

• Datasheet IC 74390

FAIRCHILD SEMICONDUCTOR™

August 1986
Revised March 2000

DM74LS390
Dual 4-Bit Decade Counter

General Description

Each of these monolithic circuits contains eight master-slave flip-flops and additional gating to implement two individual four-bit counters in a single package. The DM74LS390 incorporates dual divide-by-two and divide-by-five counters, which can be used to implement cycle lengths equal to any whole and/or cumulative multiples of 2 and/or 5 up to divide-by-100. When connected as a bi-quinary counter, the separate divide-by-two circuit can be used to provide symmetry (a square wave) at the final output stage. The DM74LS390 has parallel outputs from each counter stage so that any submultiple of the input count frequency is available for system-timing signals.

Features

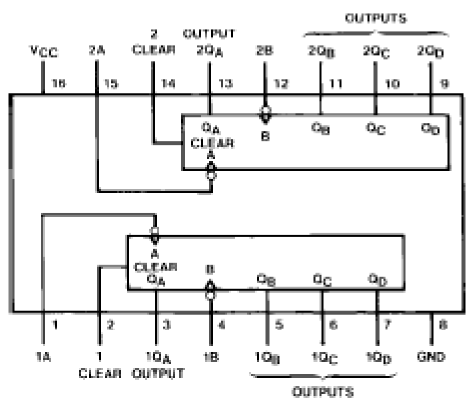
- Dual version of the popular DM74LS90
- DM74LS390…individual clocks for A and B flip-flops provide dual ÷ 2 and ÷ 5 counters
- Direct clear for each 4-bit counter
- Dual 4-bit version can significantly improve system densities by reducing counter package count by 50%
- Typical maximum count frequency…35 MHz
- Buffered outputs reduce possibility of collector commutation

Ordering Code:

Order Number	Package Number	Package Description
DM74LS390M	M16A	16-Lead Small Outline Integrated Circuit (SOIC), JEDEC MS-012, 0.150 Narrow
DM74LS390N	N16E	16-Lead Plastic Dual-In-Line Package (PDIP), JEDEC MS-001, 0.300 Wide

Devices also available in Tape and Reel. Specify by appending the suffix letter "X" to the ordering code.

Connection Diagram



 DS006433 www.fairchildsemi.com

Function Tables

BCD Count Sequence

(Each Counter) (Note 1)

Count	Outputs			
	Q_D	Q_C	Q_B	Q_A
0	L	L	L	L
1	L	L	L	H
2	L	L	H	L
3	L	L	H	H
4	L	H	L	L
5	L	H	L	H
6	L	H	H	L
7	L	H	H	H
8	H	L	L	L
9	H	L	L	H

Bi-Quinary (5-2)

(Each Counter) (Note 2)

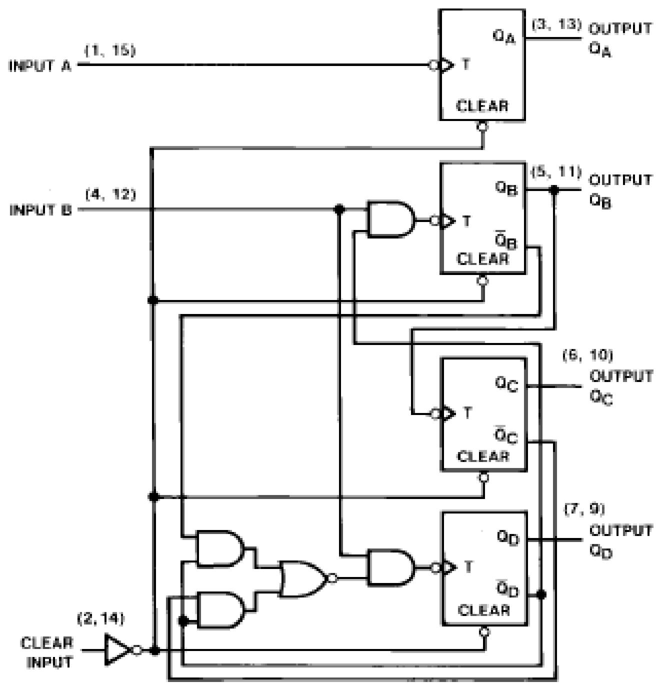
Count	Outputs			
	Q_A	Q_D	Q_C	Q_B
0	L	L	L	L
1	L	L	L	H
2	L	L	H	L
3	L	L	H	H
4	L	H	L	L
5	H	L	L	L
6	H	L	L	H
7	H	L	H	L
8	H	L	H	H
9	H	H	L	L

H = HIGH Level
L = LOW Level

Note 1: Output Q_A is connected to Input B for BCD count.

Note 2: Output Q_D is connected to Input A for Bi-quinary count.

Logic Diagram



Absolute Maximum Ratings (Note 3)

Supply Voltage	7V
Input Voltage	
Clear	7V
A or B	5.5V
Operating Free Air Temperature Range	0°C to +70°C
Storage Temperature Range	−65°C to +150°C

Note 3: The "Absolute Maximum Ratings" are those values beyond which the safety of the device cannot be guaranteed. The device should not be operated at these limits. The parametric values defined in the Electrical Characteristics tables are not guaranteed at the absolute maximum ratings. The "Recommended Operating Conditions" table will define the conditions for actual device operation.

Recommended Operating Conditions

Symbol	Parameter		Min	Nom	Max	Units
V_{CC}	Supply Voltage		4.75	5	5.25	V
V_{IH}	HIGH Level Input Voltage		2			V
V_{IL}	LOW Level Input Voltage				0.8	V
I_{OH}	HIGH Level Output Current				−0.4	mA
I_{OL}	LOW Level Output Current				8	mA
f_{CLK}	Clock Frequency (Note 4)	A to Q_A	0		25	MHz
		B to Q_B	0		20	
f_{CLK}	Clock Frequency (Note 5)	A to Q_A	0		20	MHz
		B to Q_B	0		15	
t_W	Pulse Width (Note 4)	A	20			ns
		B	25			
		Clear HIGH	20			
t_{REL}	Clear Release Time (Note 6)(Note 7)		25↓			ns
T_A	Free Air Operating Temperature		0		70	°C

Note 4: $C_L = 15$ pF, $R_L = 2$ kΩ, $T_A = 25°C$ and $V_{CC} = 5V$.

Note 5: $C_L = 50$ pF, $R_L = 2$ kΩ, $T_A = 25°C$ and $V_{CC} = 5V$.

Note 6: The symbol (↓) indicates the falling edge of the clear pulse is used for reference.

Note 7: $T_A = 25°C$ and $V_{CC} = 5V$.

Electrical Characteristics

over recommended operating free air temperature range (unless otherwise noted)

Symbol	Parameter	Conditions		Min	Typ (Note 8)	Max	Units
V_I	Input Clamp Voltage	V_{CC} = Min, I_I = −18 mA				−1.5	V
V_{OH}	HIGH Level Output Voltage	V_{CC} = Min, I_{OH} = Max V_{IL} = Max, V_{IH} = Min		2.7	3.4		V
V_{OL}	LOW Level Output Voltage	V_{CC} = Min, I_{OL} = Max V_{IL} = Max, V_{IH} = Min			0.35	0.5	V
		I_{OL} = 4 mA, V_{CC} = Min			0.25	0.4	
I_I	Input Current @ Max Input Voltage	V_{CC} = Max, V_I = 7V	Clear			0.1	mA
		V_{CC} = Max V_I = 5.5V	A			0.2	
			B			0.4	
I_{IH}	HIGH Level Input Current	V_{CC} = Max V_I = 2.7V	Clear			20	μA
			A			40	
			B			80	
I_{IL}	LOW Level Input Current	V_{CC} = Max, V_I = 0.4V	Clear			−0.4	mA
			A			−1.6	
			B			−2.4	
I_{OS}	Short Circuit Output Current	V_{CC} = Max (Note 9)		−20		−100	mA
I_{CC}	Supply Current	V_{CC} = Max (Note 10)			15	26	mA

Note 8: All typicals are at $V_{CC} = 5V$, $T_A = 25°C$.

Note 9: Not more than one output should be shorted at a time, and the duration should not exceed one second.

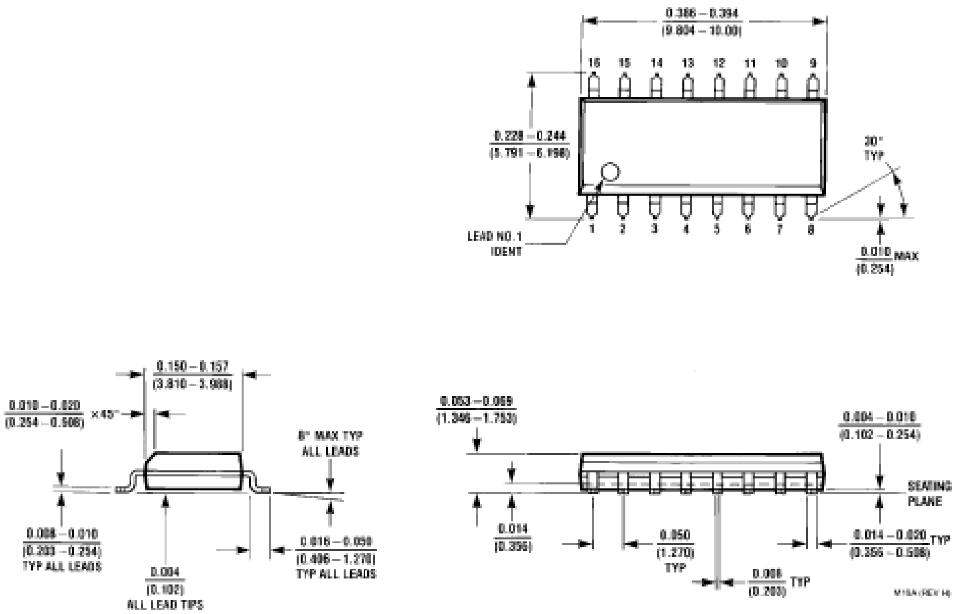
Note 10: I_{CC} is measured with all outputs OPEN, both CLEAR inputs grounded following momentary connection to 4.5 and all other inputs grounded.

Switching Characteristics

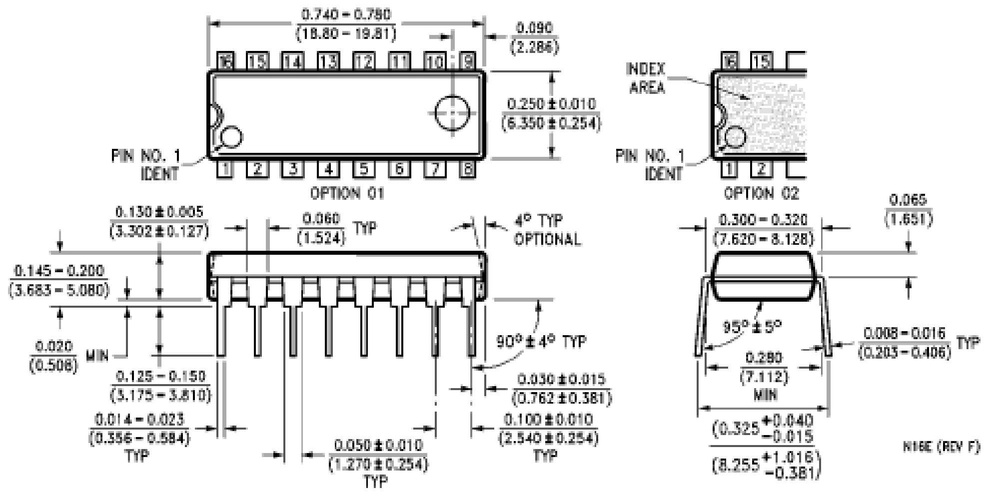
at V_{CC} = 5V and T_A = 25°C

Symbol	Parameter	From (Input) To (Output)	R_L = 2 kΩ				Units
			C_L = 15 pF		C_L = 50 pF		
			Min	Max	Min	Max	
f_{MAX}	Maximum Clock Frequency	A to Q_A	25		20		MHz
		B to Q_B	20		15		
t_{PLH}	Propagation Delay Time LOW-to-HIGH Level Output	A to Q_A		20		24	ns
t_{PHL}	Propagation Delay Time HIGH-to-LOW Level Output	A to Q_A		20		30	ns
t_{PLH}	Propagation Delay Time LOW-to-HIGH Level Output	A to Q_C		60		81	ns
t_{PHL}	Propagation Delay Time HIGH-to-LOW Level Output	A to Q_C		60		81	ns
t_{PLH}	Propagation Delay Time LOW-to-HIGH Level Output	B to Q_B		21		27	ns
t_{PHL}	Propagation Delay Time HIGH-to-LOW Level Output	B to Q_B		21		33	ns
t_{PLH}	Propagation Delay Time LOW-to-HIGH Level Output	B to Q_C		39		51	ns
t_{PHL}	Propagation Delay Time HIGH-to-LOW Level Output	B to Q_C		39		54	ns
t_{PLH}	Propagation Delay Time LOW-to-HIGH Level Output	B to Q_D		21		27	ns
t_{PHL}	Propagation Delay Time HIGH-to-LOW Level Output	B to Q_D		21		33	ns
t_{PHL}	Propagation Delay Time HIGH-to-LOW Level Output	Clear to Any Q		39		45	ns

Physical Dimensions inches (millimeters) unless otherwise noted



16-Lead Small Outline Integrated Circuit (SOIC), JEDEC MS-012, 0.150 Narrow
Package Number M16A



16-Lead Plastic Dual-In-Line Package (PDIP), JEDEC MS-001, 0.300 Wide
Package Number N16E

• Datasheet IC LM 358

National Semiconductor

October 2005

LM158/LM258/LM358/LM2904
Low Power Dual Operational Amplifiers

General Description

The LM158 series consists of two independent, high gain, internally frequency compensated operational amplifiers which were designed specifically to operate from a single power supply over a wide range of voltages. Operation from split power supplies is also possible and the low power supply current drain is independent of the magnitude of the power supply voltage.

Application areas include transducer amplifiers, dc gain blocks and all the conventional op amp circuits which now can be more easily implemented in single power supply systems. For example, the LM158 series can be directly operated off of the standard +5V power supply voltage which is used in digital systems and will easily provide the required interface electronics without requiring the additional ±15V power supplies.

The LM358 and LM2904 are available in a chip sized package (8-Bump micro SMD) using National's micro SMD package technology.

Unique Characteristics

- In the linear mode the input common-mode voltage range includes ground and the output voltage can also swing to ground, even though operated from only a single power supply voltage.
- The unity gain cross frequency is temperature compensated.
- The input bias current is also temperature compensated.

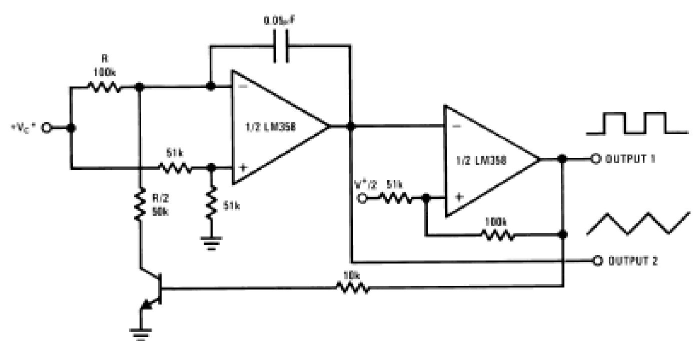
Advantages

- Two internally compensated op amps
- Eliminates need for dual supplies
- Allows direct sensing near GND and V_{OUT} also goes to GND
- Compatible with all forms of logic
- Power drain suitable for battery operation

Features

- Available in 8-Bump micro SMD chip sized package, (See AN-1112)
- Internally frequency compensated for unity gain
- Large dc voltage gain: 100 dB
- Wide bandwidth (unity gain): 1 MHz (temperature compensated)
- Wide power supply range:
  — Single supply: 3V to 32V
  — or dual supplies: ±1.5V to ±16V
- Very low supply current drain (500 μA)—essentially independent of supply voltage
- Low input offset voltage: 2 mV
- Input common-mode voltage range includes ground
- Differential input voltage range equal to the power supply voltage
- Large output voltage swing

Voltage Controlled Oscillator (VCO)



00778723

 DS007787 www.national.com

Absolute Maximum Ratings (Note 9)

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/ Distributors for availability and specifications.

	LM158/LM258/LM358 LM158A/LM258A/LM358A	LM2904
Supply Voltage, V^+	32V	26V
Differential Input Voltage	32V	26V
Input Voltage	−0.3V to +32V	−0.3V to +26V
Power Dissipation (Note 1)		
Molded DIP	830 mW	830 mW
Metal Can	550 mW	
Small Outline Package (M)	530 mW	530 mW
micro SMD	435mW	
Output Short-Circuit to GND		
(One Amplifier) (Note 2)		
$V^+ \leq 15V$ and $T_A = 25°C$	Continuous	Continuous
Input Current ($V_{IN} < -0.3V$) (Note 3)	50 mA	50 mA
Operating Temperature Range		
LM358	0˚C to +70˚C	−40˚C to +85˚C
LM258	−25˚C to +85˚C	
LM158	−55˚C to +125˚C	
Storage Temperature Range	−65˚C to +150˚C	−65˚C to +150˚C
Lead Temperature, DIP		
(Soldering, 10 seconds)	260˚C	260˚C
Lead Temperature, Metal Can		
(Soldering, 10 seconds)	300˚C	300˚C
Soldering Information		
Dual-In-Line Package		
Soldering (10 seconds)	260˚C	260˚C
Small Outline Package		
Vapor Phase (60 seconds)	215˚C	215˚C
Infrared (15 seconds)	220˚C	220˚C

See AN-450 “Surface Mounting Methods and Their Effect on Product Reliability” for other methods of soldering surface mount devices.

ESD Tolerance (Note 10)	250V	250V

Electrical Characteristics

$V^+ = +5.0V$, unless otherwise stated

Parameter	Conditions	LM158A			LM358A			LM158/LM258			Units
		Min	Typ	Max	Min	Typ	Max	Min	Typ	Max	
Input Offset Voltage	(Note 5), $T_A = 25°C$		1	2		2	3		2	5	mV
Input Bias Current	$I_{IN(+)}$ or $I_{IN(-)}$, $T_A = 25°C$, $V_{CM} = 0V$, (Note 6)		20	50		45	100		45	150	nA
Input Offset Current	$I_{IN(+)} - I_{IN(-)}$, $V_{CM} = 0V$, $T_A = 25°C$		2	10		5	30		3	30	nA
Input Common-Mode Voltage Range	$V^+ = 30V$, (Note 7) (LM2904, $V^+ = 26V$), $T_A = 25°C$	0		$V^+–1.5$	0		$V^+–1.5$	0		$V^+–1.5$	V
Supply Current	Over Full Temperature Range $R_L = \infty$ on All Op Amps										
	$V^+ = 30V$ (LM2904 $V^+ = 26V$)		1	2		1	2		1	2	mA
	$V^+ = 5V$		0.5	1.2		0.5	1.2		0.5	1.2	mA

Electrical Characteristics

$V^+ = +5.0V$, unless otherwise stated

Parameter	Conditions	LM358			LM2904			Units
		Min	Typ	Max	Min	Typ	Max	
Input Offset Voltage	(Note 5) , $T_A = 25°C$		2	7		2	7	mV
Input Bias Current	$I_{IN(+)}$ or $I_{IN(-)}$, $T_A = 25°C$, $V_{CM} = 0V$, (Note 6)		45	250		45	250	nA
Input Offset Current	$I_{IN(+)} - I_{IN(-)}$, $V_{CM} = 0V$, $T_A = 25°C$		5	50		5	50	nA
Input Common-Mode Voltage Range	$V^+ = 30V$, (Note 7) (LM2904, $V^+ = 26V$), $T_A = 25°C$	0		$V^+-1.5$	0		$V^+-1.5$	V
Supply Current	Over Full Temperature Range $R_L = \infty$ on All Op Amps							
	$V^+ = 30V$ (LM2904 $V^+ = 26V$)		1	2		1	2	mA
	$V^+ = 5V$		0.5	1.2		0.5	1.2	mA

Electrical Characteristics

$V^+ = +5.0V$, (Note 4), unless otherwise stated

Parameter		Conditions	LM158A			LM358A			LM158/LM258			Units
			Min	Typ	Max	Min	Typ	Max	Min	Typ	Max	
Large Signal Voltage Gain		$V^+ = 15V$, $T_A = 25°C$, $R_L \geq 2\ k\Omega$, (For $V_O = 1V$ to 11V)	50	100		25	100		50	100		V/mV
Common-Mode Rejection Ratio		$T_A = 25°C$, $V_{CM} = 0V$ to $V^+-1.5V$	70	85		65	85		70	85		dB
Power Supply Rejection Ratio		$V^+ = 5V$ to 30V (LM2904, $V^+ = 5V$ to 26V), $T_A = 25°C$	65	100		65	100		65	100		dB
Amplifier-to-Amplifier Coupling		f = 1 kHz to 20 kHz, $T_A = 25°C$ (Input Referred), (Note 8)		−120			−120			−120		dB
Output Current	Source	$V_{IN}^+ = 1V$, $V_{IN}^- = 0V$, $V^+ = 15V$, $V_O = 2V$, $T_A = 25°C$	20	40		20	40		20	40		mA
	Sink	$V_{IN}^- = 1V$, $V_{IN}^+ = 0V$ $V^+ = 15V$, $T_A = 25°C$, $V_O = 2V$	10	20		10	20		10	20		mA
		$V_{IN}^- = 1V$, $V_{IN}^+ = 0V$ $T_A = 25°C$, $V_O = 200$ mV, $V^+ = 15V$	12	50		12	50		12	50		µA
Short Circuit to Ground		$T_A = 25°C$, (Note 2), $V^+ = 15V$		40	60		40	60		40	60	mA
Input Offset Voltage		(Note 5)			4			5			7	mV
Input Offset Voltage Drift		$R_S = 0\Omega$		7	15		7	20		7		µV/°C
Input Offset Current		$I_{IN(+)} - I_{IN(-)}$			30			75			100	nA
Input Offset Current Drift		$R_S = 0\Omega$		10	200		10	300		10		pA/°C
Input Bias Current		$I_{IN(+)}$ or $I_{IN(-)}$		40	100		40	200		40	300	nA
Input Common-Mode Voltage Range		$V^+ = 30$ V, (Note 7) (LM2904, $V^+ = 26V$)	0		V^+-2	0		V^+-2	0		V^+-2	V

Electrical Characteristics (Continued)

V^+ = +5.0V, (Note 4), unless otherwise stated

Parameter		Conditions		LM158A			LM358A			LM158/LM258			Units
				Min	Typ	Max	Min	Typ	Max	Min	Typ	Max	
Large Signal Voltage Gain		V^+ = +15V (V_O = 1V to 11V) $R_L \geq$ 2 kΩ		25			15			25			V/mV
Output Voltage Swing	V_{OH}	V^+ = +30V (LM2904, V^+ = 26V)	R_L = 2 kΩ	26			26			26			V
			R_L = 10 kΩ	27	28		27	28		27	28		V
	V_{OL}	V^+ = 5V, R_L = 10 kΩ			5	20		5	20		5	20	mV
Output Current	Source	V_{IN}^+ = +1V, V_{IN}^- = 0V, V^+ = 15V, V_O = 2V		10	20		10	20		10	20		mA
	Sink	V_{IN}^- = +1V, V_{IN}^+ = 0V, V^+ = 15V, V_O = 2V		10	15		5	8		5	8		mA

Electrical Characteristics

V^+ = +5.0V, (Note 4), unless otherwise stated

Parameter		Conditions	LM358			LM2904			Units
			Min	Typ	Max	Min	Typ	Max	
Large Signal Voltage Gain		V^+ = 15V, T_A = 25°C, $R_L \geq$ 2 kΩ, (For V_O = 1V to 11V)	25	100		25	100		V/mV
Common-Mode Rejection Ratio		T_A = 25°C, V_{CM} = 0V to V^+–1.5V	65	85		50	70		dB
Power Supply Rejection Ratio		V^+ = 5V to 30V (LM2904, V^+ = 5V to 26V), T_A = 25°C	65	100		50	100		dB
Amplifier-to-Amplifier Coupling		f = 1 kHz to 20 kHz, T_A = 25°C (Input Referred), (Note 8)		–120			–120		dB
Output Current	Source	V_{IN}^+ = 1V, V_{IN}^- = 0V, V^+ = 15V, V_O = 2V, T_A = 25°C	20	40		20	40		mA
	Sink	V_{IN}^- = 1V, V_{IN}^+ = 0V V^+ = 15V, T_A = 25°C, V_O = 2V	10	20		10	20		mA
		V_{IN}^- = 1V, V_{IN}^+ = 0V T_A = 25°C, V_O = 200 mV, V^+ = 15V	12	50		12	50		µA
Short Circuit to Ground		T_A = 25°C, (Note 2), V^+ = 15V		40	60		40	60	mA
Input Offset Voltage		(Note 5)			9			10	mV
Input Offset Voltage Drift		R_S = 0Ω		7			7		µV/°C
Input Offset Current		$I_{IN(+)} - I_{IN(-)}$			150		45	200	nA
Input Offset Current Drift		R_S = 0Ω		10			10		pA/°C
Input Bias Current		$I_{IN(+)}$ or $I_{IN(-)}$		40	500		40	500	nA
Input Common-Mode Voltage Range		V^+ = 30 V, (Note 7) (LM2904, V^+ = 26V)	0		V^+–2	0		V^+ –2	V

Electrical Characteristics (Continued)

$V^+ = +5.0V$, (Note 4), unless otherwise stated

Parameter		Conditions		LM358			LM2904			Units
				Min	Typ	Max	Min	Typ	Max	
Large Signal Voltage Gain		$V^+ = +15V$ (V_O = 1V to 11V) $R_L \geq 2\ k\Omega$		15			15			V/mV
Output Voltage Swing	V_{OH}	$V^+ = +30V$	$R_L = 2\ k\Omega$	26			22			V
		(LM2904, $V^+ = 26V$)	$R_L = 10\ k\Omega$	27	28		23	24		V
	V_{OL}	$V^+ = 5V$, $R_L = 10\ k\Omega$			5	20		5	100	mV
Output Current	Source	$V_{IN}^+ = +1V$, $V_{IN}^- = 0V$, $V^+ = 15V$, $V_O = 2V$		10	20		10	20		mA
	Sink	$V_{IN}^- = +1V$, $V_{IN}^+ = 0V$, $V^+ = 15V$, $V_O = 2V$		5	8		5	8		mA

Note 1: For operating at high temperatures, the LM358/LM358A, LM2904 must be derated based on a +125˚C maximum junction temperature and a thermal resistance of 120˚C/W for MDIP, 182˚C/W for Metal Can, 189˚C/W for Small Outline package, and 230˚C/W for micro SMD, which applies for the device soldered in a printed circuit board, operating in a still air ambient. The LM258/LM258A and LM158/LM158A can be derated based on a +150˚C maximum junction temperature. The dissipation is the total of both amplifiers — use external resistors, where possible, to allow the amplifier to saturate or to reduce the power which is dissipated in the integrated circuit.

Note 2: Short circuits from the output to V^+ can cause excessive heating and eventual destruction. When considering short cirucits to ground, the maximum output current is approximately 40 mA independent of the magnitude of V^+. At values of supply voltage in excess of +15V, continuous short-circuits can exceed the power dissipation ratings and cause eventual destruction. Destructive dissipation can result from simultaneous shorts on all amplifiers.

Note 3: This input current will only exist when the voltage at any of the input leads is driven negative. It is due to the collector-base junction of the input PNP transistors becoming forward biased and thereby acting as input diode clamps. In addition to this diode action, there is also lateral NPN parasitic transistor action on the IC chip. This transistor action can cause the output voltages of the op amps to go to the V^+voltage level (or to ground for a large overdrive) for the time duration that an input is driven negative. This is not destructive and normal output states will re-establish when the input voltage, which was negative, again returns to a value greater than −0.3V (at 25˚C).

Note 4: These specifications are limited to $-55˚C \leq T_A \leq +125˚C$ for the LM158/LM158A. With the LM258/LM258A, all temperature specifications are limited to $-25˚C \leq T_A \leq +85˚C$, the LM358/LM358A temperature specifications are limited to $0˚C \leq T_A \leq +70˚C$, and the LM2904 specifications are limited to $-40˚C \leq T_A \leq +85˚C$.

Note 5: $V_O = 1.4V$, $R_S = 0\Omega$ with V^+ from 5V to 30V; and over the full input common-mode range (0V to V^+ −1.5V) at 25˚C. For LM2904, V^+ from 5V to 26V.

Note 6: The direction of the input current is out of the IC due to the PNP input stage. This current is essentially constant, independent of the state of the output so no loading change exists on the input lines.

Note 7: The input common-mode voltage of either input signal voltage should not be allowed to go negative by more than 0.3V (at 25˚C). The upper end of the common-mode voltage range is V^+ −1.5V (at 25˚C), but either or both inputs can go to +32V without damage (+26V for LM2904), independent of the magnitude of V^+.

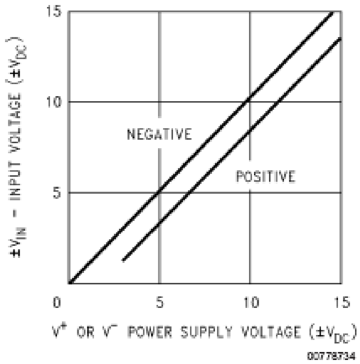
Note 8: Due to proximity of external components, insure that coupling is not originating via stray capacitance between these external parts. This typically can be detected as this type of capacitance increases at higher frequencies.

Note 9: Refer to RETS158AX for LM158A military specifications and to RETS158X for LM158 military specifications.

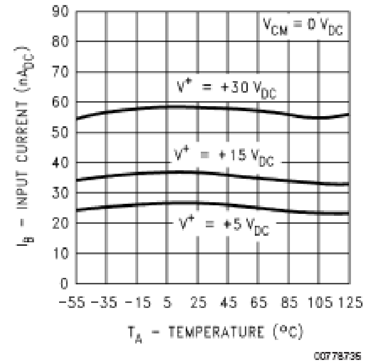
Note 10: Human body model, 1.5 kΩ in series with 100 pF.

Typical Performance Characteristics

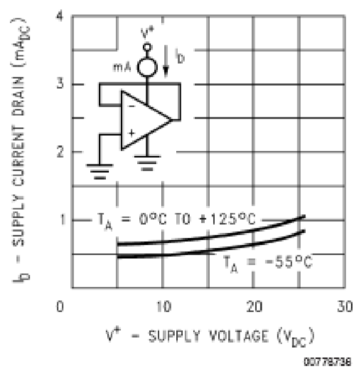
Input Voltage Range



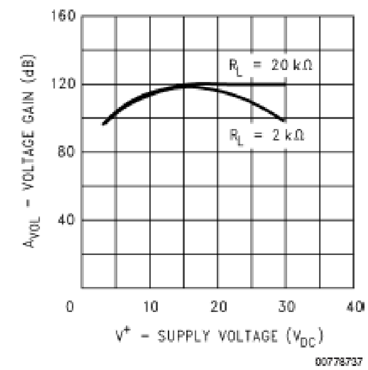
Input Current



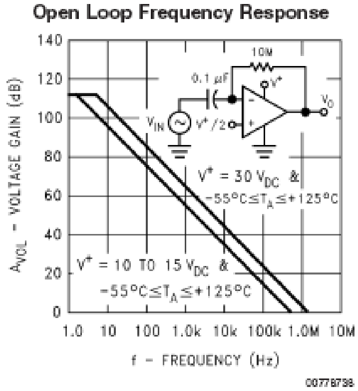
Supply Current



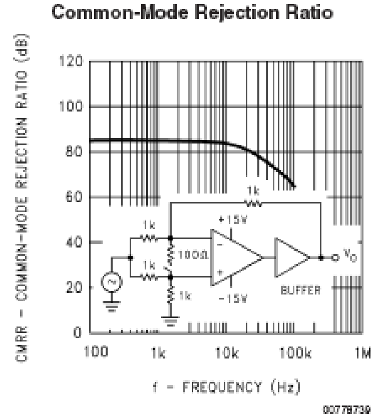
Voltage Gain



Open Loop Frequency Response

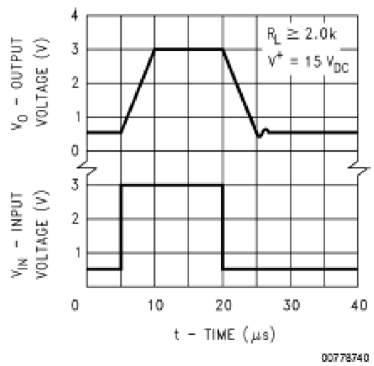


Common-Mode Rejection Ratio

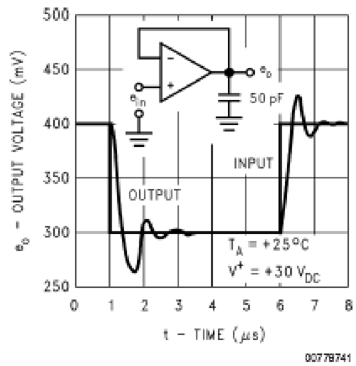


Typical Performance Characteristics (Continued)

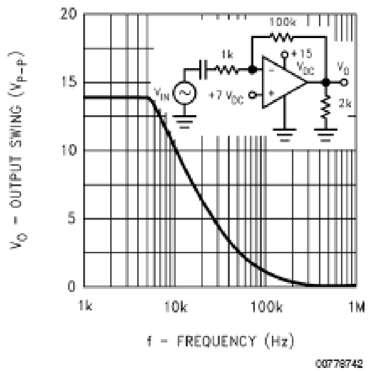
Voltage Follower Pulse Response



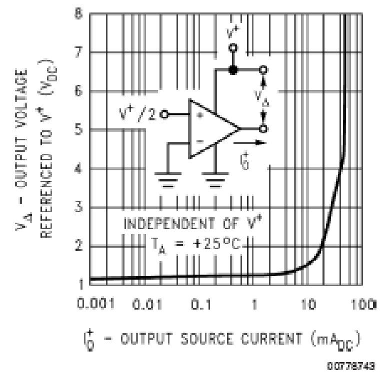
Voltage Follower Pulse Response (Small Signal)



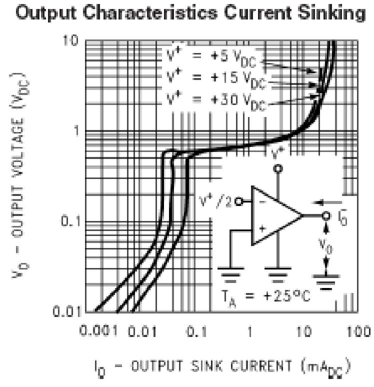
Large Signal Frequency Response



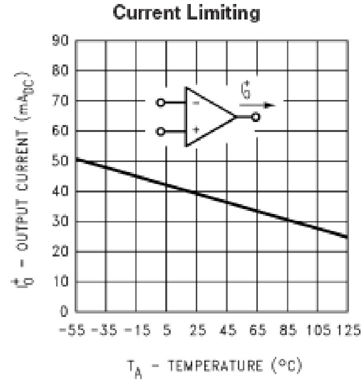
Output Characteristics Current Sourcing



Output Characteristics Current Sinking

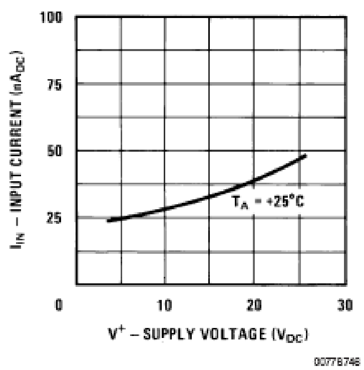


Current Limiting

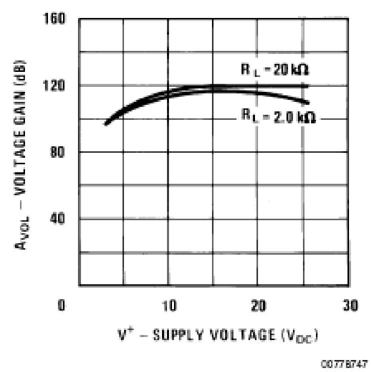


Typical Performance Characteristics (Continued)

Input Current (LM2902 only)



Voltage Gain (LM2902 only)



Application Hints

The LM158 series are op amps which operate with only a single power supply voltage, have true-differential inputs, and remain in the linear mode with an input common-mode voltage of 0 V_{DC}. These amplifiers operate over a wide range of power supply voltage with little change in performance characteristics. At 25°C amplifier operation is possible down to a minimum supply voltage of 2.3 V_{DC}.

Precautions should be taken to insure that the power supply for the integrated circuit never becomes reversed in polarity or that the unit is not inadvertently installed backwards in a test socket as an unlimited current surge through the resulting forward diode within the IC could cause fusing of the internal conductors and result in a destroyed unit.

Large differential input voltages can be easily accomodated and, as input differential voltage protection diodes are not needed, no large input currents result from large differential input voltages. The differential input voltage may be larger than V^+ without damaging the device. Protection should be provided to prevent the input voltages from going negative more than −0.3 V_{DC} (at 25°C). An input clamp diode with a resistor to the IC input terminal can be used.

To reduce the power supply current drain, the amplifiers have a class A output stage for small signal levels which converts to class B in a large signal mode. This allows the amplifiers to both source and sink large output currents. Therefore both NPN and PNP external current boost transistors can be used to extend the power capability of the basic amplifiers. The output voltage needs to raise approximately 1 diode drop above ground to bias the on-chip vertical PNP transistor for output current sinking applications.

For ac applications, where the load is capacitively coupled to the output of the amplifier, a resistor should be used, from the output of the amplifier to ground to increase the class A bias current and prevent crossover distortion. Where the load is directly coupled, as in dc applications, there is no crossover distortion.

Capacitive loads which are applied directly to the output of the amplifier reduce the loop stability margin. Values of 50 pF can be accomodated using the worst-case non-inverting unity gain connection. Large closed loop gains or resistive isolation should be used if larger load capacitance must be driven by the amplifier.

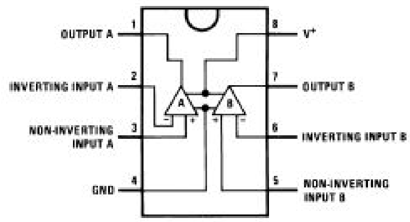
The bias network of the LM158 establishes a drain current which is independent of the magnitude of the power supply voltage over the range of 3 V_{DC} to 30 V_{DC}.

Output short circuits either to ground or to the positive power supply should be of short time duration. Units can be destroyed, not as a result of the short circuit current causing metal fusing, but rather due to the large increase in IC chip dissipation which will cause eventual failure due to excessive function temperatures. Putting direct short-circuits on more than one amplifier at a time will increase the total IC power dissipation to destructive levels, if not properly protected with external dissipation limiting resistors in series with the output leads of the amplifiers. The larger value of output source current which is available at 25°C provides a larger output current capability at elevated temperatures (see typical performance characteristics) than a standard IC op amp.

The circuits presented in the section on typical applications emphasize operation on only a single power supply voltage. If complementary power supplies are available, all of the standard op amp circuits can be used. In general, introducing a pseudo-ground (a bias voltage reference of $V^+/2$) will allow operation above and below this value in single power supply systems. Many application circuits are shown which take advantage of the wide input common-mode voltage range which includes ground. In most cases, input biasing is not required and input voltages which range to ground can easily be accommodated.

Connection Diagrams

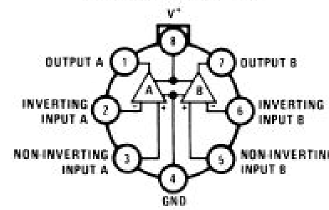
DIP/SO Package



00778702

Top View

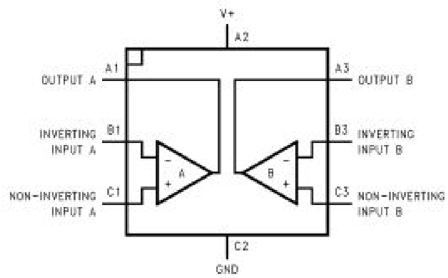
Metal Can Package



00778701

Top View

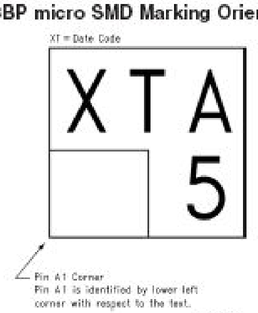
8-Bump micro SMD



00778755

Top View
(Bump Side Down)

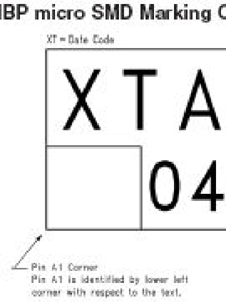
LM358BP micro SMD Marking Orientation



00778756

Top View

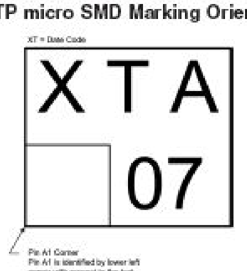
LM2904IBP micro SMD Marking Orientation



00778757

Top View

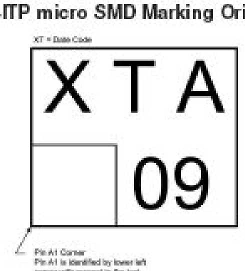
LM358TP micro SMD Marking Orientation



00778758

Top View

LM2904ITP micro SMD Marking Orientation



00778759

Top View

Ordering Information

Package	Temperature Range				NSC Drawing
	−55˚C to 125˚C	−25˚C to 85˚C	0˚C to 70˚C	−40˚C to 85˚C	
SO-8			LM358AM LM358AMX LM358M LM358MX	LM2904M LM2904MX	M08A
8-Pin Molded DIP			LM358AN LM358N	LM2904N	N08E
8-Pin Ceramic DIP	LM158AJ/883(Note 11) LM158J/883(Note 11) LM158J LM158AJLQML(Note 12) LM158AJQMLV(Note 12)				J08A
TO-5, 8-Pin Metal Can	LM158AH/883(Note 11) LM158H/883(Note 11) LM158AH LM158H LM158AHLQML(Note 12) LM158AHLQMLV(Note 12)	LM258H	LM358H		H08C
8-Bump micro SMD			LM358BP LM358BPX	LM2904IBP LM2904IBPX	BPA08AAB 0.85 mm Thick
8-Bump micro SMD Lead Free			LM358TP LM358TPX	LM2904ITP LM2904ITPX	TPA08AAA 0.50 mm Thick
14-Pin Ceramic SOIC	LM158AWG/883				WG10A

Note 11: LM158 is available per SMD #5962-8771001

LM158A is available per SMD #5962-8771002

Note 12: See STD Mil DWG 5962L87710 for Radiation Tolerant Devices