MITSUBISHI
ELECTRONIC DEVICE GROUP

Application Notes for M37702

How to Control a LED Display with User Interface

Introduction

This application note presents a program in C that displays a moving message across the LED display once pushbutton SW3 has been pressed.

Background

Controllers are often used in devices that require displays to allow users to interface with the software and need a visual interface with the user. Normally, a LED or LCD display is used and this would normally require a microcontroller with a built-in display section. But in this example the control of the display is done in software. This example shows the power of the microcontroller to be able to process the instructions quickly enough to perform this action without a built-in display section.

How It Works

The program uses a series of variables to store the 4 digit display message. But there are 11 variables to store the blank spaces as well as the actual message of "”—AbC—””. This is done to simulate movement of the message” across the display. On other displays with actual memory allocated for each digit you can normally advance the value to the next address and that will cause the movement of the digit across the display. But for this controller there are not any special memory for the display and therefore the controller must use the RAM area to hold this information and manipulate it. The program displays only 4 out of the 11 variables at one time. The first 4 "variables hold blanks, the remaining 7 holds the message "”—AbC—””. "The first 4 variables are displayed and then the data is rotated among the 11 variables for 10 times until the message and blank spaces appears. Port 4 is used to control the LED digit to be displayed. Port 5 is used to store the data for the LED digit being displayed. The controller will need to display each digit successively for 20 times then the variables values are rotated to move the digit across the display and the first 4 values are displayed 20 times more. This will continue until the first loop of 10 times finishes to display the complete message.

Program Listings

```
#pragma language=extended       /* enable use of extended keywords */
“#include “”m37702.h”””
“#include “”stdio.h”””
“#include “”intrins.h”””
#define ld0data 0x08    /* A */
#define ld1data 0x03    /* b */
#define ld2data 0x46    /* C */
#define ld3data 0xff    /*   */

/*    0     1    2    3    4    5    6    7    8
      —
 5  /    / 1    bit layout for 7 segment LEDs on board
 6   —   6    set bit to 0 to turn led on
 4 /    / 2
      —
 3 */
/* programmer: ray leung */
/* file name: abc.c */
/* move message ‘AbC’ across the LED display
   of the MTK7706 board with push of SW3 button */
/* ********************************************** */
/*    mitsubishi electronics america inc.                     */
/* ********************************************** */
/* Bit map for timer x & y */
bit cpumde = PMR.2;
bit prt61  = P6.1;
short   i;      /* variable used to display individual led */
short   m;      /* variable used to rotate all data fields */
short   n;      /* variable used to display entire led field */
short   p0data; /* variable for data field 0 */
short   p1data; /* variable for data field 1 */
short   p2data; /* variable for data field 2 */
short   p3data; /* variable for data field 3 */
short   p4data; /* variable for data field 4 */
short   p5data; /* variable for data field 5 */
short   p6data; /* variable for data field 6 */
```

Program Listings

```
short   p7data; /* variable for data field 7 */
short   p8data; /* variable for data field 8 */
short   p9data; /* variable for data field 9 */
short   padata; /* variable for data field 10 */
void dsply0(void)
{
    for (m=9; m>=0; m—) {  /* loop to rotate all 10 data fields */
        p0data=p1data;      /* rotate from data 1 to data 0 */
        p1data=p2data;      /* rotate from data 2 to data 1 */
        p2data=p3data;      /* rotate from data 3 to data 2 */
        p3data=p4data;      /* rotate from data 4 to data 3 */
        p4data=p5data;      /* rotate from data 5 to data 4 */
        p5data=p6data;      /* rotate from data 6 to data 5 */
        p6data=p7data;      /* rotate from data 7 to data 6 */
        p7data=p8data;      /* rotate from data 8 to data 7 */
        p8data=p9data;      /* rotate from data 9 to data 8 */
        p9data=padata;      /* rotate from data 10 to data 9 */
        padata=p0data;      /* rotate from data 0 to data 10 */
        for (n=20; n>=0; n—) { /* loop to display all leds */
            P4=0x7b;        /* turn on led0 only */
            P5=p0data;     /* load led0 data to port 5 */
            for(i = 1000000-82000; i >= 0; i—);   /* Sleep */
            P4=0xbb;        /* turn on led1 only */
            P5=p1data;     /* load led1 data to port 5 */
            for(i = 1000000-82000; i >= 0; i—);   /* Sleep */
            P4=0xdb;        /* turn on led2 only */
            P5=p2data;     /* load led2 data to port 5 */
            for(i = 1000000-82000; i >= 0; i—);   /* Sleep */
            P4=0xeb;        /* turn on led3 only */
            P5=p3data;     /* load led3 data to port 5 */
            for(i = 1000000-82000; i >= 0; i—);   /* Sleep */
        }
    }
}
void main(void)
{
    disable_interrupt();
    cpumde=1;
    P5=0x88;          /* turn off leds */
    P5D=0xff;         /* set port latch direction */
    P4=0xff;            /* turn off all leds */
    P4D=0xf8;           /* set port latch direction */
    TA0MR=0x88;          /* initialize timerxy */
    enable_interrupt();       /* enable interrupts */
    while (1)
        {
        if (prt61==0)
            {
            if (P5==0xff)
                {
                p0data=0xff;
                p1data=0xff;
                p2data=0xff;
                p3data=0xff;
                p4data=0xbf;
                p5data=0xbf;
                p6data=ld0data;
                p7data=ld1data;
                p8data=ld2data;
                p9data=0xbf;
                padata=0xbf;
                dsply0();
                P5=0x00;     /* turn on leds */
                }
            else
                P5=0xff;
            }
        else
        P5=0xff;
        }
}
```