Munc5X

Library

With

COMPILER MICRO-C V2.4

By **MIBot** team [e-mail : raek@se-ed.net , satanraek@yahoo.com]

Supachai Budsaratij (È Ø ª Ñ ° È Ø 0 · Ô)

Panida Intarayaem (¾ ¹ Ô ì Í Ô · Ã á Ë Â Á)

Hardware support

CP-SB31 V1 / V2

MTool-7

V3155

ET-BOARD V6 [MCS-51] - New update 2 -

Update 1 - March 27,2000 | Update 2 - August 8,2000

Copyright 2000, By Supachai Budsaratij

ÊÒÃºÑ

Readme.TXT

Library and Header files

Tark5x.H	- Main definition and MARCO processor.
LCD.H	- LCD library.
i8255.H	- 8255 PPI library.
Misc.H	- Utilitis library.
SB31.H	- Header file for ETT CP-SB31 V1.0/V2.0
V3155.H	- Header file for SILA V3155
MTOOL7.H	- Header file for SILA MTOOL-7 V2.0
Etv6.H	**- Header file for ETT ET-BOARD V6 [MCS-51]** - New update 2 -

Samples files

HelloSB.C	- Hello world for ETT CP-SB31 V1.0/V2.0
HelloMT.C	- Hello world for SILA MTOOL-7 V2.0
GetADC.C	- ·´ÊÍº Í®¹¤®¨Ò¡ LTC1298
MT7SegX.C	- Test the Multiplex 7-Segment with 8255 port A/B.
MT7Step.C	- Test stepping motor with Mtool-7
MT7Track.C	- Test read logic from P1.7
TarkOne.C	- Test moving mobile robot with CP-SB31 V1/V2
HelloV6.C	**- Hello world for ETT ET-BOARD V6 [MCS-51]** - New update 2 -

```
/*
 * Filename    : sb31.h
 * Author      : Supachai Budsaratij
 *                     School of Computer and Advanced Technologies
 *                     Rajabhat Institute Phetchaburi
 * e-mail      : raek@se-ed.net
 * Date        : March 27,2000
 * Compiler    : Micro-C V2.4
 * Hardware    : ETT CP-SB31 V1.0/V2.0
 *
 */

#define ETT_SB31
#define USE_LCD
#define USE_8255

/*
#define USE_RS232
#define USE_WATCHDOG
*/


#define       i8255PA       0xE0E0
#define       i8255PB       0xE0E1
#define       i8255PC       0xE0E2
#define       i8255PX       0xE0E3

asm {
i8255PAA            EQU   $E0E0
i8255PBA            EQU   $E0E1
i8255PCA            EQU   $E0E2
i8255PXA            EQU   $E0E3

WR_INSLCDA          EQU   $E0C0
RD_INSLCDA                EQU   $E0C1
WR_DATLCDA          EQU   $E0C2
RD_DATLCDA          EQU   $E0C3
}

#define       WR_INSLCD     0xE0C0
#define       RD_INSLCD     0xE0C1
#define       WR_DATLCD     0xE0C2
#define       RD_DATLCD     0xE0C3


#include "Tark5x.h"


#ifdef USE_WATCHDOG

/* ====================================================================

  WatchDogACK( )
```

```
  Reset watch dog timer.

  Parameter : none
  Return    : none
  Notice    : If you setting WatchDOG Timer ON. You must call this.

  ---------------------------------------------------------------------- */

WatchDogACK( ) asm {
   mov    DPTR,#$E0A0
   movx   [DPTR],#$00
}

#endif
```

```
/*
 * Filename    : MTool7.h
 * Author      : Supachai Budsaratij
 *                    School of Computer and Advanced Technologies
 *                    Rajabhat Institute Phetchaburi
 * e-mail      : raek@se-ed.net
 * Date        : March 27,2000
 * Compiler    : Micro-C V2.4
 * Hardware    : SILA MTOOL-7 V2.0
 *
 */

#define SILA_MTOOL7

#define USE_LCD
#define USE_8255
#define USE_RS232
#define USE_WATCHDOG

#define i8255PA             0xFC00
#define i8255PB             0xFC01
#define i8255PC             0xFC02
#define i8255PX             0xFC03

#define WR_INSLCD           0xFA00
#define RD_INSLCD           0xFA01
#define WR_DATLCD           0xFA02
#define RD_DATLCD           0xFA03

asm {
i8255PAA    EQU   $FC00
i8255PBA    EQU   $FC01
i8255PCA    EQU   $FC02
i8255PXA    EQU   $FC03

WR_INSLCDA   EQU   $FA00
RD_INSLCDA   EQU   $FA01
WR_DATLCDA   EQU   $FA02
RD_DATLCDA   EQU   $FA03
}

#include "Tark5x.h"


/* ======================================================================

  WatchDogACK( )

  Reset watch dog timer.

  Parameter : none
  Return    : none
  Notice    : Must call this function before 1.2 sec because WatchDog
              Timer must reset MTOOL-7.
```

```
  ---------------------------------------------------------------------- */

WatchDogACK( ) asm {
   clr   P3.2
   setb  P3.2
}

/*
WatchDogACK( )
{
   clrbit(P3.2);
   setbit(P3.2);
}
*/
```

```
/*
 * Filename    : v3155.h
 * Author      : Supachai Budsaratij
 *                    School of Computer and Advanced Technologies
 *                    Rajabhat Institute Phetchaburi
 * e-mail      : raek@se-ed.org
 * Date        : March 27,2000
 * Compiler    : Micro-C V2.4
 * Hardware    : SILA  V3155
 *
 */

#define SILA_V3155
#define USE_8255

#define USE_RS232

#define i8255PA   0x8000
#define i8255PB   0x8001
#define i8255PC   0x8002
#define i8255PX   0x8003

asm {
i8255PAA     EQU    $8000
i8255PBA     EQU    $8001
i8255PCA     EQU    $8002
i8255PXA     EQU    $8003
}

#include "Tark5x.h"
```

```
/*
 * Filename    : Tark5x.h
 * Author      : Supachai Budsaratij
 *                     School of Computer and Advanced Technologies
 *                     Rajabhat Institute Phetchaburi
 * e-mail      : raek@se-ed.org
 * Date        : March 27,2000
 * Compiler    : Micro-C V2.4
 * Hardware    :       ETT  CP-SB31 V1.0/V2.0
 *                     SILA MTOOL-7 V2.0
 *
 */
#include <8051reg.h>
#include <8051bit.h>

#ifdef USE_8255
#include "i8255.h"
#endif

#ifdef USE_LCD
#include "lcd.h"
#endif

#include "misc.h"
```

```
/*
 * Filename    : i8255.h
 * Author      : Supachai Budsaratij
 *                       School of Computer and Advanced Technologies
 *                       Rajabhat Institute Phetchaburi
 * e-mail      : raek@se-ed.net
 * Date        : March 27,2000
 * Compiler    : Micro-C V2.4
 * Hardware    :       ETT  CP-SB31 V1.0/V2.0
 *                       SILA MTOOL-7 V2.0
 *
 */

#ifndef i8255_H

#define i8255_H

#define i8255waitF          30 /* Fast */
#define i8255waitM           50 /* Middle */
#define i8255waitS          70 /* Slow */



/* ======================================================================

  i8255Delay( )

  Waiting for 8255 working.

  Parameter : none
  Return    : none
  Notice    : none

  ---------------------------------------------------------------------- */

i8255Delay( )
{
   unsigned char i;

#ifdef i8255_FAST
       for (i=0;i<i8255waitF;i++);
#else ifdef i8255_SLOW
       for (i=0;i<i8255waitS;i++);
#else
       for (i=0;i<i8255waitM;i++);
#endif

#ifdef USE_WATCHDOG
   WatchDogACK();
#endif

}
```

```
/* ===========================================================================

  i8255Write( portx, pdata )

  Send pdata to portx of 8255.

  Parameter : portx = Address of 8255 port.
              pdata = Data for send to 8255's portx.
  Return    : none
  Notice    : none

  --------------------------------------------------------------------- */

i8255Write(portx,pdata)
int portx;
unsigned char pdata;
{
   poke(portx, pdata);
   i8255Delay();

#ifdef USE_WATCHDOG
   WatchDogACK();
#endif

}


/* ===========================================================================

  i8255Read( portx )

  Read data from portx of 8255.

  Parameter : portx = Address of 8255's port.
  Return    : value from portx.
  Notice    : none

  --------------------------------------------------------------------- */

unsigned char i8255Read( ) asm {
   mov     R0,#-5
   lcall   ?auto0

#ifdef USE_WATCHDOG
   lcall   WatchDogACK
#endif

   mov     A,[R0]     Move LO-BYTE into A
   mov     DPL,A

   inc     R0         Get next byte (HI-BYTE)
   mov     A,[R0]     Move HI-BYTE into A
```

```
    mov     DPH,A

    movx    A,[DPTR]
    mov     B,#$00
}

/*
unsigned char i8255Read(portx)
int portx;
{
    WatchDogACK();
    return(peek(portx));
}
 */


#endif
```

```
/*
 * Filename    : LCD.h
 * Author      : Supachai Budsaratij
 *                   School of Computer and Advanced Technologies
 *                   Rajabhat Institute Phetchaburi
 * e-mail      : raek@se-ed.net
 * Date        : March 27,2000
 * Compiler    : Micro-C V2.4
 * Hardware    :       ETT  CP-SB31 V1.0/V2.0
 *                   SILA MTOOL-7 V2.0
 *
 */

#ifndef LCD_H

#define LCD_H

#define     LCD_CLS             0x01
#define     LCD_HOME            0x02

/*
 * LCD set up mode
 */

#define     LCD_SET_8B2L5x10    0x3C /* 8bit, 2Line, 5x10 */
#define     LCD_SET_8B2L5x7     0x38 /* 8bit, 2Line, 5x7  */
#define     LCD_SET_8B1L5x10    0x34 /* 8bit, 1Line, 5x10 */
#define     LCD_SET_8B1L5x7     0x30 /* 8bit, 1Line, 5x7  */
#define     LCD_SET_4B2L5x10    0x2C /* 4bit, 2Line, 5x10 */
#define     LCD_SET_4B2L5x7     0x28 /* 4bit, 2Line, 5x7  */
#define     LCD_SET_4B1L5x10    0x24 /* 4bit, 1Line, 5x10 */
#define     LCD_SET_4B1L5x7     0x20 /* 4bit, 1Line, 5x7  */

/*
 * LCD Shift cursor or display
 */

#define     LCD_SHL_CUR         0x10
#define     LCD_SHL_DSP         0x18
#define     LCD_SHR_CUR         0x14
#define     LCD_SHR_DSP         0x1C

/*
 * LCD Entry mode
 */

#define     LCD_ENT_DTINC       0x06 /* Data not move, Cursor move */
#define     LCD_ENT_CUINC       0x07 /* Data move, Cursor not move */
#define     LCD_ENT_DTDEC       0x04 /* Data not move, Cursor move */
#define     LCD_ENT_CUDEC       0x05 /* Data move, Cursor not move */

/*
 * LCD Display mode
 */
```

```
#define         LCD_DSP_OFF         0x08 /* Display OFF */
#define         LCD_DSP_ON                  0x0C /* Display ON */
#define         LCD_DSP_CURSOR              0x0E /* Display ON, CURSOR ON */
#define         LCD_DSP_CBLINK              0x0F /* Display ON, CURSOR BLINK */


/* ========================================================================

  LcdBusy( )

  Wait if LCD is busy.

  Parameter : none
  Return    : none
  Notice    : Use with 8bit mode only.

  ------------------------------------------------------------------------ */

LcdBusy( ) asm {
        mov     DPTR,#RD_INSLCDA
?LcdBusy   movx    A,[DPTR]
        jb      A.7,?LcdBusy

#ifdef USE_WATCHDOG
        lcall   WatchDogACK
#endif
}

/*
LcdBusy( )
{
   while ((peek(RD_INSLCD) & 0x80) == 0x80) {
#ifdef USE_WATCHDOG
      WatchDogACK();
#endif
   }
}
 */

/* ========================================================================

  LcdCommand( command )

  Send command to LCD.

  Parameter : command = command's byte for control LCD.
  Return    : none
  Notice    : none

  ------------------------------------------------------------------------ */

LcdCommand() asm {
   mov      R0,#-5
```

```
   lcall     ?auto0

   mov       A,[R0]
   mov       DPTR,#WR_INSLCDA
   movx      [DPTR],A

   lcall     LcdBusy
}

/*
LcdCommand(command)
unsigned char   command;
{
   poke(WR_INSLCD,command);
   LcdBusy();
}
 */

/* ============================================================================

  LcdPutC( ascii )

  Show a character on LCD.

  Parameter : ascii = ASCII number of character.
  Return    : none
  Notice    : none

  ---------------------------------------------------------------------- */

LcdPutC( ) asm {
   mov       R0,#-5
   lcall     ?auto0

   mov       A,[R0]
   mov       DPTR,#WR_DATLCDA
   movx      [DPTR],A

   lcall     LcdBusy
}

/*
LcdPutC(ascii)
unsigned char   ascii;
{
   LcdBusy();
   poke(WR_DATLCD,ascii);
}

 */

/* ============================================================================

  LcdPutS( string )
```

```
  Show more characters on LCD.

  Parameter : string = Data of null string.
  Return    : none
  Notice    : Your string length must less than or equal to maximum
              LCD's column.

  ------------------------------------------------------------------------ */

LcdPutS(string)
unsigned char  *string;
{
   while (*string != 0x00) {
      LcdPutC(*string);
      string++;
   }
}


/* ========================================================================

  LcdGoto(x,y)

  Move DDRAM Address to X,Y.
  Parameter : x = column number. (0-15)
              y = row number. (0-3)
  Return    : none
  Notice    : This module use for 16x4 only, other size must change
              value of row address.

  ------------------------------------------------------------------------ */

LcdGoto(x,y)
unsigned char x;
unsigned char y;
{
   switch (y) {
    case 0: y = 0x80; break;
    case 1: y = 0xC0; break;
    case 2: y = 0x90; break;
    case 3: y = 0xD0; break;
    }
    y += x;
    LcdBusy();
    poke(WR_INSLCD,y);
}


/* ========================================================================

  LcdInit( )

  Initialize LCD before used.
```

```
   Parameter : none
   Return    : none
   Notice    : none

   ------------------------------------------------------------------------ */

LcdInit()
{
   LcdCommand(LCD_SET_8B2L5x10);
   LcdCommand(LCD_DSP_ON);
   LcdCommand(LCD_ENT_DTINC);
   LcdCommand(LCD_CLS);
}


#endif
```

```
/*
 * Filename    : Misc.h
 * Author      : Supachai Budsaratij
 *                     School of Computer and Advanced Technologies
 *                     Rajabhat Institute Phetchaburi
 * e-mail      : raek@se-ed.net
 * Date        : March 27,2000
 * Compiler    : Micro-C V2.4
 * Hardware    :       ETT  CP-SB31 V1.0/V2.0
 *                     SILA MTOOL-7 V2.0
 *
 */

#ifndef MISC_H

#define MISC_H

PowerOnReset()
{
#ifdef USE_8255
   unsigned char inner, outer;

   for (outer = 0x00; outer < 0x40; outer++) {
      for (inner = 0x00; inner < 0xFF; inner++);

#ifdef USE_WATCHDOG
      WatchDogACK();
#endif
   }
   poke(i8255PX, 0x80);
#endif

#ifdef USE_LCD
   LcdInit();
#endif

#ifdef USE_RS232
   serinit(9600);
#endif
}

unsigned char IsDEC(ch)
unsigned char ch;
{
   if ((ch>='0')&&(ch<='9'))
      return(1);
   return(0);
}

unsigned char IsHEX(ch)
unsigned char ch;
{
   if (((ch>='0')&&(ch<='9'))||
      ((ch>='a')&&(ch<='f'))||
```

```
        ((ch>='A')&&(ch<='F')))
        return(1);
    return(0);
}

#endif
```

```
/*
 * Filename    : HelloSB.c
 * Author      : Supachai Budsaratij
 *                      School of Computer and Advanced Technologies
 *                      Rajabhat Institute Phetchaburi
 * e-mail      : raek@se-ed.net
 * Date        : March 27,2000
 * compile     : cc51 helloSB -pio m=s
 * Hardware    : ETT  CP-SB31 V1.0/V2.0
 */
#include <8051io.h>
#include "sb31.h"

main()
{
   int count;

   PowerOnReset();
   count = 0;
   LcdCommand(LCD_CLS);
   LcdGoto(0,0); LcdPutS("- drOT startup -");
   LcdGoto(1,1); LcdPutS("Line 2.");
   LcdGoto(2,2); LcdPutS("Line 3.");
   LcdGoto(3,3); LcdPutS("Line 4.");

   while(1) {
      printf("Hello no.%d\n", count);
      if (count++ == 10000) {
         count = 0;
      }
   }
}
```

```
/*
 * Filename    : HelloMT7.c
 * Author      : Supachai Budsaratij
 *                      School of Computer and Advanced Technologies
 *                      Rajabhat Institute Phetchaburi
 * e-mail      : raek@se-ed.net
 * Date        : March 27,2000
 * compile     : cc51 helloMT7 -pio m=s
 * Hardware    : SILA MTOOL7 V2.0
 */
#include <8051io.h>
#include "mtool7.h"

main()
{
   int count;

   PowerOnReset();
   count = 0;
   LcdCommand(LCD_CLS);
   LcdGoto(0,0); LcdPutS("- drOT startup -");
   LcdGoto(1,1); LcdPutS("Line 2.");
   LcdGoto(2,2); LcdPutS("Line 3.");
   LcdGoto(3,3); LcdPutS("Line 4.");

   while(1) {
      printf("Hello no.%d\n", count);
      if (count++ == 10000) {
         count = 0;
      }
      WatchDogACK();
   }
}
```

```
/*
 * Filename    : GetADC.c
 * Author      : Supachai Budsaratij
 *                  School of Computer and Advanced Technologies
 *                  Rajabhat Institute Phetchaburi
 * e-mail      : raek@se-ed.net
 * Date        : March 30,2000
 * compile     : cc51 getADC -pio m=s
 * Hardware    : ETT  CP-SB31 V1.0/V2.0 with LTC1298 A/D Convertor.
 * Note        : ·´ÊÍºÍÒ¹¤Ò¨Ò¡äÍ«ÕÍÃ•LTC1298«Öè»š A/D Convertor º¹ºÍÃ• áÅÇáÊ´§¼Å¼Ò¹
RS232
* ¶ÒŒÍ§¡ÒÃãªŒ ã´ãËŒ #define USE_xxxx
       #define USE_LCD
       #define USE_8255
       #define USE_RS232
       #define USE_WATCHDOG
       #define USE_ADC
*/

#define USE_LCD
#define USE_ADC

/*
  The value return from LDR-Sensor:
       White zone = 0,        Dark zone  = 1
 */

#include <8051io.h>
#include "sb31.h"

main()
{
   int left, right;

   PowerOnReset();
   LcdCommand(LCD_CLS);
   LcdGoto(0,0); LcdPutS("> Tark51 ROBOT <");
   LcdGoto(0,1); LcdPutS("----------------");
   LcdGoto(0,2); LcdPutS("Left sensor  = 0");
   LcdGoto(0,3); LcdPutS("Right sensor = 0");

   while(1) {
      left  = ReadLTC1298(0);
      right = ReadLTC1298(1);

      left >>= 8;
      right >>= 8;

      LcdGoto(0,2);
      (left>0)?LcdPutS("Left sensor  = 1"):LcdPutS("Left sensor  = 0");

      LcdGoto(0,3);
```

```
        (right>0)?LcdPutS("Right sensor = 1"):LcdPutS("Right sensor = 0");
    }
}
```

```
/*
 * Filename    : MT7segX.c
 * Author      : Supachai Budsaratij
 *                    School of Computer and Advanced Technologies
 *                    Rajabhat Institute Phetchaburi
 * e-mail      : raek@se-ed.net
 * Date        : March 27,2000
 * compile     : cc51 MT7segX -pio m=L
 * Hardware    : SILA MTOOL7 V2.0
 */
#define USE_LCD
#define USE_RS232
#define USE_8255

#include <8051io.h>
#include "mtool7.h"

unsigned char enable[6] = {0x01, 0x02, 0x04, 0x08, 0x10, 0x20};

main()
{
   unsigned char count;
   int loops;

   PowerOnReset();
   count = 0;
   LcdCommand(LCD_CLS);
   LcdGoto(0,0); LcdPutS("- Munc5x startup -");

   while(1) {
      i8255Write(i8255PA,0x3F);
      i8255Write(i8255PB,~enable[count]);
      count++;
      if (count == 6) count = 0;
      for (loops = 0; loops < 250; loops++) {
         WatchDogACK();
         i8255Delay();
      }
   }
}
```

```
/*
 * Filename    : MT7Step.c
 * Author      : Supachai Budsaratij
 *                     School of Computer and Advanced Technologies
 *                     Rajabhat Institute Phetchaburi
 * e-mail      : raek@se-ed.net
 * Date        : March 27,2000
 * compile     : cc51 MT7Step -pio m=L
 * Hardware    : SILA MTOOL7 V2.0
 */
#define USE_LCD
#define USE_RS232
#define USE_8255

#include <8051io.h>
#include "mtool7.h"


main()
{
   int loops;

   PowerOnReset();
   LcdCommand(LCD_CLS);
   LcdGoto(0,0); LcdPutS("- STEP motor -");

   while(1) {
      i8255Write(i8255PA,0x11);
      for (loops = 0; loops < 40; loops++) {
         WatchDogACK();
      }
      i8255Write(i8255PA,0x22);
      for (loops = 0; loops < 40; loops++) {
         WatchDogACK();
      }
      i8255Write(i8255PA,0x44);
      for (loops = 0; loops < 40; loops++) {
         WatchDogACK();
      }
      i8255Write(i8255PA,0x88);
      for (loops = 0; loops < 40; loops++) {
         WatchDogACK();
      }
   }
}
```

```
/*
 * Filename    : MT7Track.c
 * Author      : Supachai Budsaratij
 *                      School of Computer and Advanced Technologies
 *                      Rajabhat Institute Phetchaburi
 * e-mail      : raek@se-ed.net
 * Date        : March 27,2000
 * compile     : cc51 MT7Track -pio m=L
 * Hardware    : SILA MTOOL7 V2.0
 */
#define USE_LCD
#define USE_RS232
#define USE_8255

#include <8051io.h>
#include "mtool7.h"


main()
{
   PowerOnReset();

   LcdCommand(LCD_CLS);
   LcdGoto(0,0); LcdPutS("> Munc startup <");

   while(1) {
      if (P1 & 0x80) {
         LcdGoto(0,1);
         LcdPutS("P1.7 = Hi");
      }
      else {
         LcdGoto(0,1);
         LcdPutS("P1.7 = Lo");
      }
      WatchDogACK();
   }
}
```

```
/*
 * Filename     : TarkOne.c
 * Author       : Supachai Budsaratij
 *                    School of Computer and Advanced Technologies
 *                    Rajabhat Institute Phetchaburi
 * e-mail       : raek@se-ed.net
 * Date         : March 31,2000
 * compile      : cc51 TarkOne -pio m=s
 * Hardware     : ETT  CP-SB31 V1.0/V2.0 with LTC1298 A/D Convertor.
 * Note         : ·´ÊÍºÍèÒ¹¤èÒ¨Ò¡äÍ«ÕÍÃ LTC1298 «Öè§à»ç¹ A/D Convertor
*                 º¹ºÍÃì´ áÅÐáÊ´§¼Å¼èÒ¹ RS232
 *                ¶éÒµéÍ§¡ÒÃãªé§Ò¹ã´ãËéãÊè #define USE_xxxx
        #define USE_LCD
        #define USE_8255
        #define USE_RS232
        #define USE_WATCHDOG
        #define USE_ADC
*/

#define USE_8255
#define USE_RS232

#define USE_LCD
#define USE_ADC

#include <8051io.h>
#include "sb31.h"

/* ---- STEPPER Routine ---- */

MotorDelay()
{    unsigned char  loops;
     for (loops = 0; loops < 220; loops++);
}

MotorStop()
{    i8255Write(i8255PA,0x00);     }

MotorForward()
{   i8255Write(i8255PA,0x99); MotorDelay();
    i8255Write(i8255PA,0x88); MotorDelay();
    i8255Write(i8255PA,0xCC); MotorDelay();
    i8255Write(i8255PA,0x44); MotorDelay();
    i8255Write(i8255PA,0x66); MotorDelay();
    i8255Write(i8255PA,0x22); MotorDelay();
    i8255Write(i8255PA,0x33); MotorDelay();
    i8255Write(i8255PA,0x11); MotorDelay();
}

MotorBackward()
{   i8255Write(i8255PA,0x11); MotorDelay();
    i8255Write(i8255PA,0x33); MotorDelay();
```

```
    i8255Write(i8255PA,0x22); MotorDelay();
    i8255Write(i8255PA,0x66); MotorDelay();
    i8255Write(i8255PA,0x44); MotorDelay();
    i8255Write(i8255PA,0xCC); MotorDelay();
    i8255Write(i8255PA,0x88); MotorDelay();
    i8255Write(i8255PA,0x99); MotorDelay();
}

MotorLeft()
{   i8255Write(i8255PA,0x09); MotorDelay();
    i8255Write(i8255PA,0x08); MotorDelay();
    i8255Write(i8255PA,0x0C); MotorDelay();
    i8255Write(i8255PA,0x04); MotorDelay();
    i8255Write(i8255PA,0x06); MotorDelay();
    i8255Write(i8255PA,0x02); MotorDelay();
    i8255Write(i8255PA,0x03); MotorDelay();
    i8255Write(i8255PA,0x01); MotorDelay();
}

MotorRight()
{   i8255Write(i8255PA,0x90); MotorDelay();
    i8255Write(i8255PA,0x80); MotorDelay();
    i8255Write(i8255PA,0xC0); MotorDelay();
    i8255Write(i8255PA,0x40); MotorDelay();
    i8255Write(i8255PA,0x60); MotorDelay();
    i8255Write(i8255PA,0x20); MotorDelay();
    i8255Write(i8255PA,0x30); MotorDelay();
    i8255Write(i8255PA,0x10); MotorDelay();
}

ShowSensor()
{    int left, right;

    left  = ReadLTC1298(0);
    right = ReadLTC1298(1);

    left >>= 8;
    right >>= 8;
    LcdCommand(LCD_CLS);
    LcdGoto(0,0); LcdPutS("> Tark51 ROBOT <");
    LcdGoto(0,1); LcdPutS("----------------");
    LcdGoto(0,2);
    (left>0)?LcdPutS("Left sensor  = 1"):LcdPutS("Left sensor  = 0");
    LcdGoto(0,3);
    (right>0)?LcdPutS("Right sensor = 1"):LcdPutS("Right sensor = 0");
}

/* ---- MAIN FUNCTION ----*/

main()
{
    register char menu;

    PowerOnReset();
```

```
printf("Hello\n");

while (1) {
   MotorStop();
   ShowSensor();

   printf("\n--INTbot MENU--\n");
   printf("\n1. Forward");
   printf("\n2. Backward");
   printf("\n3. Turn Left");
   printf("\n4. Turn Right");
   printf("\n5. Sensor status");
   printf("\n--------------");
   printf("\nX. Exit");
   printf("\n--------------");
   printf("\nChoose > ");

   menu = getch();
   putch(menu);

   LcdCommand(LCD_CLS);
   switch (menu) {
   case '1':
      while (chkchr()==-1) {
         MotorForward();
         ShowSensor();
      }
      break;
   case '2':
      while (chkchr()==-1) {
         MotorBackward();
         ShowSensor();
      }
      break;
   case '3':
      while (chkchr()==-1) {
         MotorLeft();
         ShowSensor();
      }
      break;
   case '4':
      while (chkchr()==-1) {
         MotorRight();
         ShowSensor();
      }
      break;
   case '5':
      while (chkchr()==-1) {
         ShowSensor();
      }
      break;
   case 'x':
   case 'X':
      exit();
```

```
      }
    }
  }
```

```
/*
 * Filename : etv6.h
 * Author   : Supachai Budsaratij
 *          School of Computer and Advanced Technologies
 *          Rajabhat Institute Phetchaburi
 * e-mail   : raek@se-ed.org
 * Date     : August 8, 2000
 * Compiler : Micro-C V2.4
 * Hardware : ETT ET-BOARD V6 [MCS-51]
 *
 */

#define ETT_V6

/* 8255 User port
 */
#define i8255PA   0xE020
#define i8255PB   0xE021
#define i8255PC   0xE022
#define i8255PX   0xE023

/* 8255 System port
 */
#define s8255PA   0xE000     /* PA0-3 = Selector 7Seg/LED, Other I2C bus */
#define s8255PB   0xE001     /* 7-Segment/LEDx8 */
#define s8255PC   0xE002     /* Hi = Row's Key, Lo = I2C clock/control */
#define s8255PX   0xE003

asm {
i8255PAA      EQU    $E020
i8255PBA      EQU    $E021
i8255PCA      EQU    $E022
i8255PXA      EQU    $E023

s8255PAA      EQU    $E000
s8255PBA      EQU    $E001
s8255PCA      EQU    $E002
s8255PXA      EQU    $E003

WR_INSLCDA    EQU    $E060
RD_INSLCDA    EQU    $E061
WR_DATLCDA    EQU    $E062
RD_DATLCDA    EQU    $E063
}

#define WR_INSLCD  0xE060
#define RD_INSLCD  0xE061
#define WR_DATLCD  0xE062
#define RD_DATLCD  0xE063


#include "Tark5x.h"
```

```
#endif
```

```
/*
 * Filename : HelloV6.c
 * Author   : Supachai Budsaratij
 *          School of Computer and Advanced Technologies
 *          Rajabhat Institute Phetchaburi
 * e-mail   : raek@se-ed.org
 * Date     : 08-08-2000 [dd-mm-yyyy]
 * compile  : cc51 helloV6 -pio m=L
 * Hardware : ETT  ET-BOARD V6 [MCS-51]
 */

#define USE_LCD
#define USE_RS232

#include <8051io.h>
#include "etv6.h"

main()
{
   int count;

   PowerOnReset();
   count = 0;
   LcdCommand(LCD_CLS);
   LcdGoto(0,0); LcdPutS("- drOT startup -");
   LcdGoto(1,1); LcdPutS("Line 2.");
   LcdGoto(2,2); LcdPutS("Line 3.");
   LcdGoto(3,3); LcdPutS("Line 4.");

   while(1) {
      printf("Hello no.%d\n", count);
      if (count++ == 10000) {
         count = 0;
      }
   }
}
```