Embedded Controller Programming II

I/O Device Programming in C

Part 2: Display Device Drivers in C

Ken Arnold

1

Overview




- Displays:
  - LED
    - Light Emitting Diode
  - LCD
    - Liquid Crystal Display
  - VFD
    - Vacuum Flourescent Display
- Display Multiplexing
- Interrupt Driven I/O

2

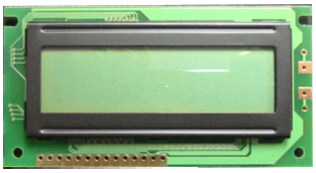
Displays

- Basic Displays
  - Simple On/Off Indicators LEDs
  - Simple Liquid Crystal Displays
  - Multiplexed LED Numeric Displays
- Intelligent Displays
  - Parallel Alphanumeric LCD
  - Serial LCD

3

Liquid Crystal Display

- Simple Text Displays
- Optional Back-lighting
- Graphic Displays
- Mono/Color, Touch-Screen Input Overlay and Controller



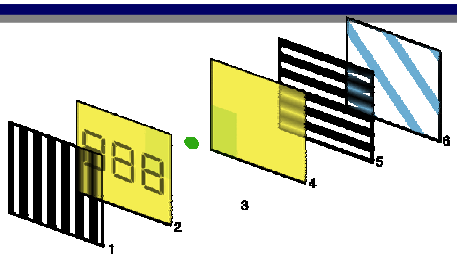
- For more info see:
  - http://en.wikipedia.org/wiki/LCD
  - http://www.howstuffworks.com/lcd.htm
  - http://www.geocities.com/dinceraydin/lcd/index.html

4

Passive LCD Operation

LCD display:

Reflective twisted nematic liquid crystal display



1. Vertical filter film allows vertically polarized light to enter.

2. Glass with thin metal film (transparent) electrodes on inner surface in contact with liquid crystal. The shapes of these electrodes determine the dark images that will appear when the LCD is on. Vertical ridges etched on the surface align the liquid crystals with the polarized light.

3. Twisted nematic liquid crystals. Rotate light polarization 90 degrees when there's no electric field – when an electric field is present, they randomize the polarization.

4. Glass substrate with thin metal film electrode on upper side (in contact with liquid crystal) has horizontal ridges to line liquid crystal molecules up with the horizontal filter.

5. Horizontal filter film to allow through only horizontally polarized light.

6. Reflective surface sends light back to viewer.

Source: http://en.wikipedia.org/wiki/LCD

5

LCD Interface

- **LCD Interfacing can be done 2 ways:**
  - Port I/O Mapped
    - 4 data bits and 3 control lines or
    - 8 data bits and 3 control lines
  - Memory Mapped (use MOVX instructions)
    - *Timing is Problematic for Faster Processors*

6

Demo: LCD Program in C

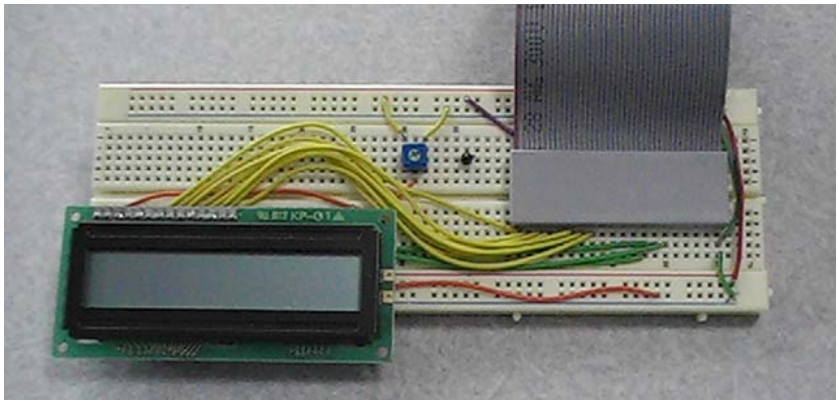
- Hello Program:
- Outputs to LCD
- and Serial Port
  - Character Set
  - 0x00 to 0xFF
  - Random Numbers

Example code from Section 11: Display Devices
program LCD1: http://hte.com/uconline/ecp/lcd1.exe

7

Liquid Crystal Display

- LCD Module on Solderless Prototyping Board

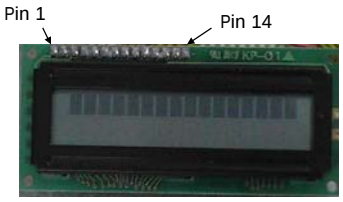


8

LCD Connector Pinout

Pin#	Symbol	Level	Function
1	Vss	GND	Ground
2	Vcc	+V	Module power
3	Vee		Liquid crystal drive (bias)
4	RS_	H/L	Register select, H=data, L=instruction
5	R/W	H/L	Read/Write, H=read (LCD->CPU) L=write (CPU->LCD)
6	E		Edge-sensitive Enable

Pin#	Symbol	Level	Function
7	DB0		Data bit 0 (LSB)
8	DB1		"
9	DB2		"
10	DB3		"
11	DB4		"
12	DB5		"
13	DB6		"
14	DB7		"



9

LCD Connections to CPU:

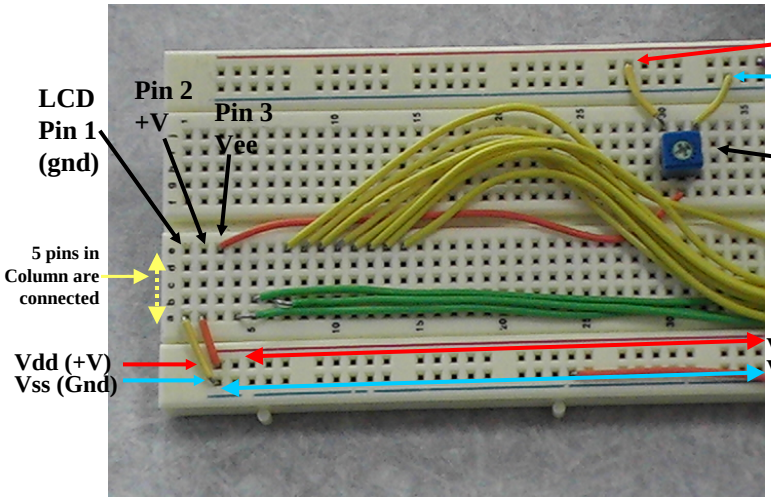
CPU Signal	LCD Signal	CPU Pin#	LCD Pin#	
P1.0	D0	1	7	Data
P1.1	D1	2	8	Data
P1.2	D2	3	9	Data
P1.3	D3	4	10	Data
P1.4	D4	5	11	Data
P1.5	D5	6	12	Data
P1.6	D6	7	13	Data
P1.7	D7	8	14	Data

CPU Signal	LCD Signal	CPU Pin#	LCD Pin#	Signal Function
P3.3	RS	13	4	Register Select
P3.4	R/W	14	5	Read/Write RD:1, WR:0
P3.5	EN	15	6	Enable (1)
	GND		1	Ground
	+V		2	+Supply V
	Vee		3	LCD Bias (Contrast)

Connections are shown for the sample LCD1 program, which uses a software driven, bit-banged interface.

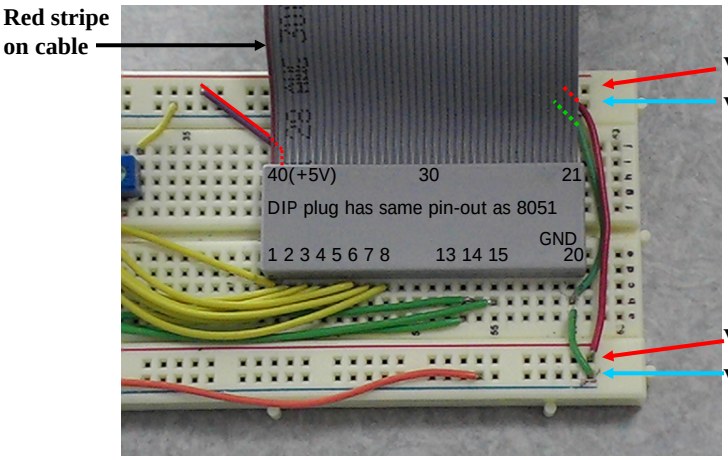
10

Connections to LCD



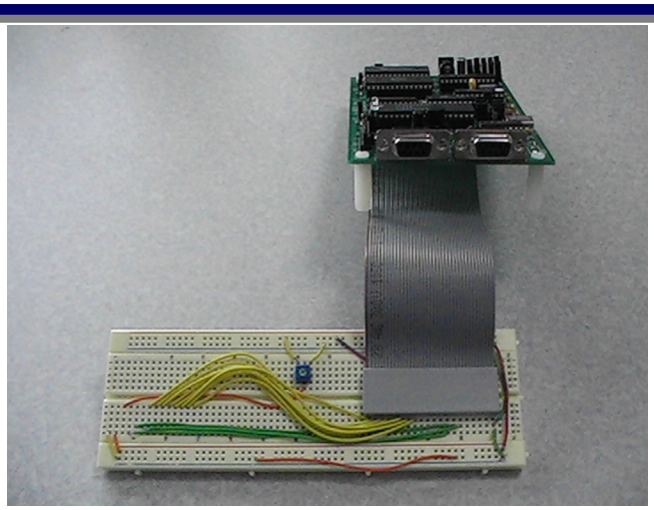
11

Connections to SDK Cable



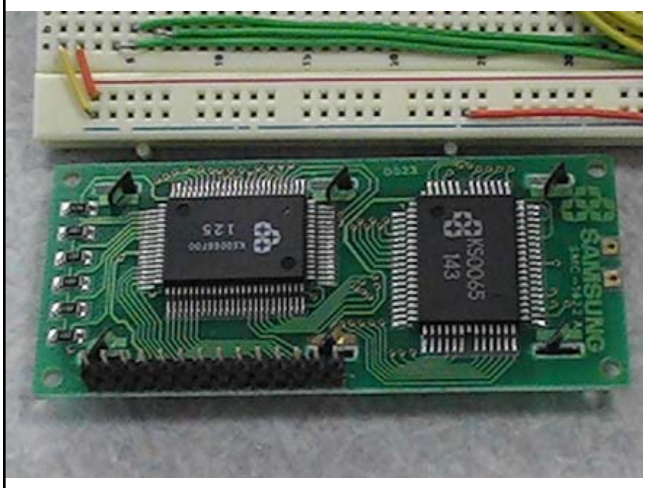
12

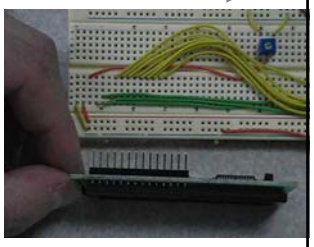
SDK Cable to Proto Board

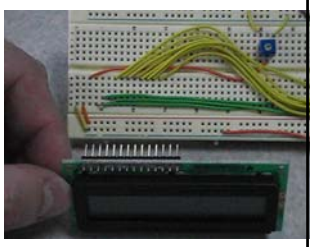


13

LCD Back with Connector







14

Screen Capture: LCD FAQ

- The LCD FAQ tour goes here.

15

Some Basic LCD Routines

```
void data_write (unsigned char value) {
  DISPDATA = value;           // Latch Data in Port
  RS = 1;                     // Select Data Register
  RW = 0;                     // Select Write Mode
  EN = 1;                     // Strobe data in
  EN = 0;
 }

Void cmd_write (unsigned char value) {
  DISPDATA = value;           // Latch Data in Port
  RS = 0;                    // Select Command Register
  RW = 0;                     // Select Write Mode
  EN = 1;                     // Strobe data in
  EN = 0;
 }
```

16

Basic LCD Routines (cont.)

```
char lcd_write_data(char LCDdata)
{
  LCDDataLine = LCDdata;    // Write output byte to Port
  RS = 1;          // Turn on  (RS)
  RW = 0;          // Turn off (R/W)
  EN = 1;          // Turn on  (EN)
  EN = 0;          // Turn off (EN)
  return (LCDdata); // Return Byte Written
}
```

17

Basic LCD Routines (cont.)

```
char data_read (void)
{
  unsigned char value;
  DISPDATA = 0xFF;         // Setup port for Inputs
  RS = 1;                  // select the data reg
  RW = 1;                  // select read mode
  EN = 1;                  // enable the LCD output
  value = DISPDATA;        // read in the data
  EN = 0;                  // disable the LCD output
  return (value);
}
```

18

LCD Interface Timing

- LCD Timing requires the following:
  - Configure LCD Display at Initialization
  - Write Operation Timing Constraints
  - Read Operation Timing Constraints
  - Handling Display Not Ready:
    - Polling Ready Status, Requires Reading LCD
    - Simple: Use Maximum Delay for All Operations

19

Text LCD Parallel Interface

- Memory Mapped Interface
- See LCD Application Note 51LCD.PDF
- FAQ File lcdfaq.txt
- CPU Bus Signals:
  - Addr, /RD, /WR signals vs. Enable, R/W, RS
  - Data bus connects directly: D0..7
  - Still Need +V, GND, Bias (contrast)

20

Serial LCD Modules

- Convert Async Serial Data to Parallel
- Logic Level (5V) or RS-232 Level Versions
- Reduced I/O Pins (as little as One pin)
- No “Bit-Banging” Required
- Much Simpler Interface, Much Higher Cost
- Proprietary Serial Data Formats
- Some are Like a Text Terminal Emulator

21

LCD/LED/VFD Sources

Sources of Displays

Timeline Inc. – www.timeline-inc.com

All Electronics – www.allelectronics.com

Digi-Key – www.digikey.com

Jameco – www.jameco.com

JDR – www.jdr.com

Serial LCDs, VFDs – www.seetron.com

Lots of other surplus sources:

Gateway Electronics, Hosfelt Electronics...

22

7 Segment LED Displays

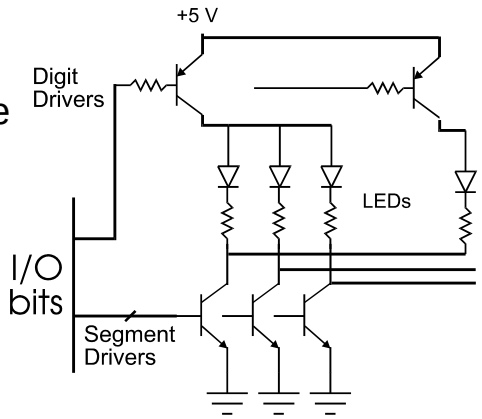
7 Segment Displays can be configured for a wide variety of applications:

- Single 7 Segment Display (One 8-bit I/O port)
- 7 Segment Displays with BCD/7 Segment Decoder Driver
- Multiple 7 Segment Displays with Software Multiplexing (see example LED1)
- Multiple 7 Segment Displays with external Controller (Maxim, others make chips)

23

Driving an LED Matrix

- For 7-Segment LEDs
- Use external transistors for source and sink current
- One transistor pulls digit anodes high
- One transistor pulls cathode segments low



24

Multiplexing LED Displays

```
/* THIS IS FROM PROGRAM LED1 */

  void main (void)
  {
    unsigned char dig1,dig2;
    unsigned char i;
    unsigned char array1[] =
   {0x3F,0x06,0x5B,0x4F,0x66,0x6D,0x7D,0x07,
   0x7F,0x67,0x77,0x7C,0x39,0x5E,0x79,0x71};

    T1 = 0;
    T0 = 1;
    dig1 = 0;
    dig2 = 0;
    delaysec(1);
```

25

Multiplexing LED Displays (cont.)

```
  while (1)
    {
         for (i=0;i<50;++i)
         {
          P1 = array1[dig1];
          T0 = 1; T1 = 0;      /* Toggle Output Line */
          delaymsec(10);
          P1 = array1[dig2];
          T0 = 0; T1 = 1;      /* Toggle Output Line */
          delaymsec(10);
         }
         ++dig1;          /* increment ones digit */
         if (dig1 == 10)  /* if 10 reset to 0 */
          {
           dig1 = 0;
           ++dig2;        /* increment tens digit */
          }

         if (dig2 == 10) /* if 10 reset to 0 */
           dig2 = 0;
       }
    }
```

26

Interrupts for Multiplex Delay

```
FILE LED2.C

#include <stdio.h>
#include <reg51.h>
#include <intrins.h>
#include <time.h>

#define RELOADH 0xdc /* Setup to Interrupt every 10 mS */
#define RELOADL 0x00

  unsigned char array1[] =  /* digits 0 - 9 */
    {0x3F,0x06,0x5B,0x4F,0x66,0x6D,0x7D,0x07,
     0x7F,0x67,0x77,0x7C,0x39,0x5E,0x79,0x71};
  unsigned char dig1,dig2;
```

27

Using Interrupts for Multiplexing (cont.)

```
void multiplex_tick (void) interrupt 1 {
/* Take care of timer - note TF0 is automatically cleared */
    static bit tick_flag = 0;
    TR0 = 0;                  /* Stop Timer    */
    TH0 = RELOADH;            /* Reload Timer  */
    TL0 = RELOADL;
    TR0 = 1;                  /* Restart Timer */
    if (tick_flag)
     {
       P1 = array1[dig1];
       T1 = 0; T0 = 1;      /* Toggle Output Line */
     }
    else
     {
       P1 = array1[dig2];
       T0 = 0; T1 = 1;      /* Toggle Output Line */
     }
    tick_flag ^= 1;       /* XOR bitflag */
  }
```

28

Using Interrupts for Multiplexing (cont.)

```
void main (void) {
  T1 = 0; T0 = 1; /* T0 & T1 output pins: P3.4 and P3.5 */
  dig1 = 0; dig2 = 0;  /* Initialize the two counting digits */
  TMOD = 0x11;      /* Setup Both Timers in Mode 1 */
  TH0 = RELOADH; TL0 = RELOADL; /* Reload Timer  */
  TF0 = 0; TR0 = 1; /* Clear Overflow Flag, Start Timer 0 */
  EA = 1;           /* Enable Global Interrupts */
  ET0 = 1;          /* Enable Timer 0 Interrupt */
  while (1) {
    ET0 = 0;       /* Disable Timer 0 Interrupt */
    if (dig1 == 10) {
      dig1 = 0;
      ++dig2;
     }
      if (dig2 == 10)
      dig2 = 0;
    ET0 = 1;       /* Reenable Timer 0 Interrupt */
    delaysec(1);
    ++dig1;
    }
}
```

29

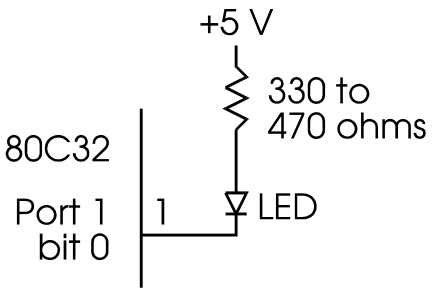
Output Current for 80C32

- I_{OL} (sink current) max = ~15 mA
  - This is an ABSOLUTE MAXIMUM spec!
  - *Greater I_{OL} may damage the device!*
  - Total I_{OL} for 8 bits may not exceed ~26 mA
- I_{OH} (source current) max = 50 uA=.05 mA
  - Non-destructive - self limited
  - Static when output is shorted to ground
  - Maximum: -650 uA During 1->0 Transition

30

Driving an LED directly

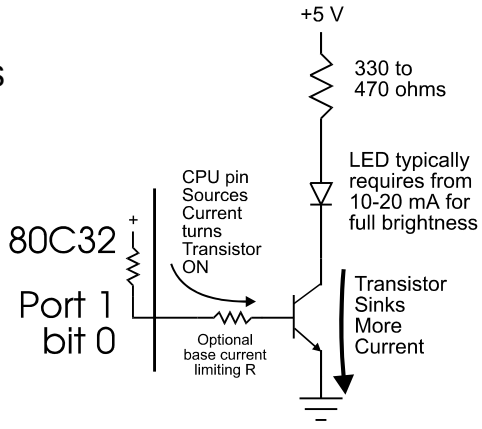
- 8051 Port I_{OL} >> I_{OH}
- Drive LED from 5V
- LED OFF when bit= 1
- LED ON when bit= 0
- For LED with Vf=~2V
  - voltage across R is: +5V - 2V = 3V
  - current in R and LED: 3V/330 ohms = 9 mA



31

Increasing Output Drive

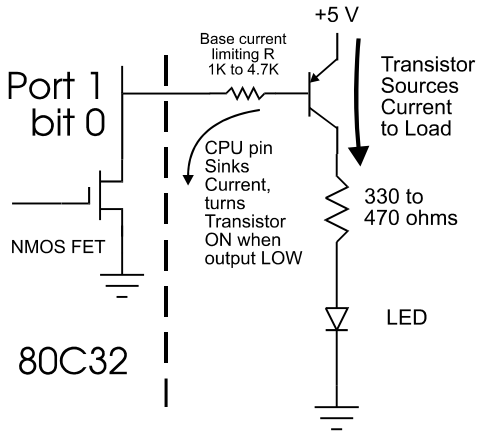
- Use NPN as a switch
- Transistor gain limits load sink current
- High Beta (1,000 or more) Darlington
- Higher output voltage, limit > 5V
- Port 1 min. source current is 50 uA



32

PNP Transistor Output

- CPU I_{OL} > I_{OH}
  - I_{OL} = 1.6 mA
  - I_{OH} = 50 uA
- Transistor can have lower gain than NPN
- I/O pin=1 on Reset, so transistor is OFF
- Works with load connected to Ground



33

Standard Port Output

Data Output via a Port Pin (buzzer, single LED, scope probe, etc.) ;

```
void foo (int intval)
  {
   P1.0 = 0; /* signal function start */
   .   /* function code */
   .
   P1.0 = 1; /* signal function end */
  }
```

34

Summary

- Basic Output
- Displays:
  - LED, LCD
- Multiplexing I/O

35

Homework

- Your Class Project is Due Next Week!

36