

```
// Program to operate a Vacuum Fluorescent Display
// using an Arduino Uno, Atmega328P microcontroller.
```

```
/*code: [modified eg 12.5.4-Mazidi, 8 pin 2 ports DON]
```

```
#define F_CPU 16000000UL
```

```
#include <avr/io.h>
```

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

```
#define LCD_DPRT PORTD // configuring PortD for data
```

```
#define LCD_DDDR DDRD
```

```
#define LCD_DPIN PIN_D
```

```
#define LCD_CPRT PORTB // utilizing PortB pins for the control.
```

```
#define LCD_CDDR DDRB
```

```
#define LCD_CPIN PINB
```

```
#define LCD_RS 0 // control pin assignments.
```

```
#define LCD_RW 1
```

```
#define LCD_EN 2
```

```
//*****
```

```
void delay_us(unsigned int d)
```

```
{
    _delay_us(d);
}
```

```
//*****
```

```
void lcdCommand( unsigned char cmnd )
```

```
{
    LCD_DPRT = cmnd;
    LCD_CPRT &= ~(1<<LCD_RS); // LCD_RS/LCD_RW pins of PortB are cleared...
    LCD_CPRT &= ~(1<<LCD_RW); // ...as required to send commands to LCD.
    LCD_CPRT |= (1<<LCD_EN); // LCD_EN pin of PortB is set (pulsed)for internal latching to send a command.
    delay_us(.02); // delay 20 ns for LCD module to run a command.
    LCD_CPRT &= ~(1<<LCD_EN); // LCD_EN pin of PortB is cleared after sending a command.
    delay_us(.26); // delay 260ns between commands sent.
}
```

```
//*****
```

```
void lcdData( unsigned char data )
```

```
{
    LCD_DPRT = data;
    LCD_CPRT |= (1<<LCD_RS); // LCD_RS pin of PortB is set!!!!as required to send data.
    LCD_CPRT &= ~(1<<LCD_RW); // LCD_RW pin of PortB is cleared as required to send data.
    LCD_CPRT |= (1<<LCD_EN); // LCD_EN pin of PortB is set for internal latching of the LCD to send data.
    delay_us(100); // delay 100us for LCD module to write data to screen.
    LCD_CPRT &= ~(1<<LCD_EN); // LCD_EN pin of PortB is cleared after sending data byte.
    delay_us(100); // delay of 100us between data bytes sent.
}
```

```
//*****
```

```
void lcd_init() // Equivalent to void InitializeComputerBoard (void).
```

```
{
    // ...exception: the delay btwn clear display and entry mode.
    LCD_DDDR = 0xFF; // Port D is configured as output.
}
```

```

LCD_CDDR = 0xFF;      // Port B is configured as output.

LCD_CPRT &=~(1<<LCD_EN); // LCD_EN pin of PortB is cleared in preparation for sending commands.
delay_us(2000);        // delay of 2ms before sending commands.
lcdCommand(0x38);      // Function set command.
lcdCommand(0x0E);      // Display on with blinking cursor.
lcdCommand(0x01);      // Display cleared.
delay_us(2000);
lcdCommand(0x06);      // Entry mode set
}

//*****
void lcd_gotoxy(unsigned char x, unsigned char y)
{
    unsigned char firstCharAdr[]={0x80,0xC0,0x94,0xD4}; //table 12-5 of text.
    lcdCommand(firstCharAdr[y-1] + x - 1); // calculating the address location to the first and subsequent characters.
    delay_us(100); // delay of 100us btwn each character being displayed
}

//*****
void lcd_print( char * str )
{
    unsigned char i = 0 ;
    while(str[i]!=0)
    {
        lcdData(str[i]); // string array of characters to be displayed
        i++; // each member of array called up incrementally.
    }
}

//*****
int main(void)
{
    lcd_init(); // init routine.
    lcd_gotoxy(1,1);
    lcd_print("The world is but");
    lcd_gotoxy(1,2);
    lcd_print("one country");
    while(1); // do forever.
    return 0;
}

*/code

```