ARYABHATTA 8051 DEVELOPMENT BOARD WITH ON-BOARD USB PROGRAMMER
1. Descriptions:

With this board you can develop and prototype 40 pin 89S52 and 89s51 microcontrollers. The on board programmer allows easy connection with PC using USB type B cable for Programming. The Operating Voltage is 9V to 15V DC.

2. Features:

1. Quartz crystal 11.0592 MHz
2. On board programmer.
3. Reset button.
5. GND bus.
6. VCC bus.
7. On board 5V voltage regulator.
8. Power Indicating LED.
9. On board Regulated Power Supply 5V, 12V, GND.
11. External pull-up resistors for Port 0.
12. Port extensions for all ports.
3. Specifications:

1. Size: 85 x 62 mm.

4. Hardware Details:

PORT 0(P0.0-P0.7)- 8 bit Bidirectional I/O port with external pullup and having multiplexed low-order address/data bus
PORT 1(P1.0-P1.7)- 8-bit bidirectional I/O port with internal pull-ups.
PORT 2(P2.0-P2.7)- 8 bit Bidirectional I/O port with internal pull-ups and having multiplexed Higher-order address/data bus.
5. Examples:

1. LED blinking
2. Serial Communication
3. 7 Segment Display
4. LCD interfacing

1) **LED blinking:**

Schematic:
Code:

```c
#include<reg52.h>  // special function register declarations

sbit LED = P2^0;  // LED pin

void Delay(void);  // Function prototype declaration

void main(void)
{
    while(1)  // infinite loop
    {
        LED = 0;  // LED ON
        Delay();
        LED = 1;  // LED OFF
        Delay();
    }
}

void Delay(void)
{
    int j;
    int i;
    for(i=0;i<10;i++)
    {
        for(j=0;j<10000;j++)
        {
        }
    }
}
```
2) **Serial Communication:**

**Schematic:**

![Serial Communication Schematic](image)

**Description:**

Serial data transmitted by 8051 is printed on serial terminal of PC by using PL2303 or FTDI232 which are usb to ttl convretre.

**Code:**

```c
#include<reg51.h>

void main(void)
{
    TMOD=0x20; // Timer1 Mode2 8 bit auto reload
}
```

- +5V of PL2303 to +5V of 8051
- GND of PL2303 to GND of 8051
- TXD of PL2303 to RX of 8051
- RxD of PL2303 to TX of 8051

*PL2303 is USB to TTL converter. FTDI chip also work.*
TH1=0xFD; // 9600 bps
SCON=0x50; // 8 Data bit, 1 start bit, 1 stop bit
TR1=1; // Timer1 ON

while(1==1)
{
    SBUF='R';
    while(TI==0); // Pole TI flag for complete transmission
    TI=0;
    SBUF='O';
    while(TI==0);
    TI=0;
    SBUF='B';
    while(TI==0);
    TI=0;
    SBUF='U';
    while(TI==0);
    TI=0;
    SBUF='!';
    while(TI==0);
    TI=0;
    SBUF='T';
    while(TI==0);
    TI=0;
    SBUF='N';
    while(TI==0);
    TI=0;
}
}
3. 7 Segment Display:

Schematic:
Code:

```c
#include<reg51.h>

void msdelay(unsigned int time) // Function for creating delay in milliseconds.
{
    unsigned i,j;
    for(i=0;i<time;i++)
        for(j=0;j<1275;j++);
}

void main()
{
    unsigned char
    no_code[]={0xC0,0xF9,0xA4,0xB0,0x90,0x92,0x82,0xF8,0x80,0x90}; // Array for hex values 0-9 for common anode 7 segment
    int k;
    while(1)
    {
        for(k=0;k<10;k++)
        {
            P1=no_code[k];
            msdelay(100);
        }
    }
}
```
4. LCD 16x2

SCHEMATIC:
**Code:**

// Program for LCD Interfacing with 8051 Microcontroller (AT89S52)
#include<reg51.h>
#define display_port P2      //Data pins connected to port 2 on microcontroller
sbit rs = P3^2;     //RS pin connected to pin 2 of port 3
sbit rw = P3^3;     //RW pin connected to pin 3 of port 3
sbit e = P3^4;      //E pin connected to pin 4 of port 3

void msdelay(unsigned int time) // Function for creating delay in milliseconds.
{
    unsigned i,j;
    for(i=0;i<time;i++)
        for(j=0;j<1275;j++);
}

void lcd_cmd(unsigned char command) //Function to send command instruction to LCD
{
    display_port = command;
    rs= 0;
    rw=0;
    e=1;
    msdelay(1);
    e=0;
}

void lcd_data(unsigned char disp_data) //Function to send display data to LCD
{
    display_port = disp_data;
    rs= 1;
    rw=0;
    e=1;
    msdelay(1);
    e=0;
}

void lcd_init() //Function to prepare the LCD and get it ready
{
    lcd_cmd(0x38); // for using 2 lines and 5X7 matrix of LCD
    msdelay(10);
    lcd_cmd(0x0f); // turn display ON, cursor blinking
    msdelay(10);
    lcd_cmd(0x01); //clear screen
    msdelay(10);
    lcd_cmd(0x81); // bring cursor to position 1 of line 1
    msdelay(10);
}

void main()
{
    int l=0;
    lcd_init();
while(a[l] != '\0') // searching the null terminator in the sentence
{
    lcd_data(a[l]);
    l++;
    msdelay(50);
}
}