**Write an 8051 C program to send values 00 – FF to port P1.**

Solution:

#include <reg51.h>

void main(void)

{ unsigned char z;

for (z=0;z<=255;z++)

P1=z; }

**Write an 8051 C program to send hex values for ASCII characters of**

**0, 1, 2, 3, 4, 5, A, B, C, and D to port P1.**

Solution:

#include <reg51.h>

void main(void)

{

unsigned char mynum[]=“012345ABCD”;

unsigned char z;

for (z=0;z<=10;z++)

P1=mynum[z];

}

**Write an 8051 C program to toggle all the bits of P1 continuously.**

Solution:

//Toggle P1 forever

#include <reg51.h>

void main(void)

{

for (;;)

{

p1=0x55;

p1=0xAA;

}}

**Write an 8051 C program to send values of –4 to +4 to port P1.**

Solution:

//Singed numbers

#include <reg51.h>

void main(void)

{

char mynum[]={+1,-1,+2,-2,+3,-3,+4,-4};

unsigned char z;

for (z=0;z<=8;z++)

P1=mynum[z];

}

**Write an 8051 C program to toggle bit D0 of the port P1 (P1.0)**

**50,000 times.**

Solution:

#include <reg51.h>

sbit MYBIT=P1^0;

void main(void)

{

unsigned int z;

for (z=0;z<=50000;z++)

{

MYBIT=0;

MYBIT=1;

}

}

**Write an 8051 C program to toggle bits of P1 continuously forever**

**with some delay.**

Solution:

//Toggle P1 forever with some delay in between

//“on” and “off”

#include <reg51.h>

void main(void)

{

unsigned int x;

for (;;) //repeat forever

{

p1=0x55;

for (x=0;x<40000;x++); //delay size

//unknown

p1=0xAA;

for (x=0;x<40000;x++);

}

**Write an 8051 C program to toggle bits of P1 ports continuously with**

**a 250 ms.**

Solution:

#include <reg51.h>

void MSDelay(unsigned int);

void main(void)

{

while (1) //repeat forever

{

p1=0x55;

MSDelay(250);

p1=0xAA;

MSDelay(250);

}

}

void MSDelay(unsigned int itime)

{

unsigned int i,j;

for (i=0;i<itime;i++)

for (j=0;j<1275;j++);

}

**Write an 8051 C program to get a byte of data form P1, wait 1/2**

**second, and then send it to P2.**

Solution:

#include <reg51.h>

void MSDelay(unsigned int);

void main(void)

{

unsigned char mybyte;

P1=0xFF; //make P1 input port

while (1)

{

mybyte=P1; //get a byte from P1

MSDelay(500);

P2=mybyte; //send it to P2

}

**Write an 8051 C program to toggle only bit P2.4 continuously without**

**disturbing the rest of the bits of P2.**

Solution:

//Toggling an individual bit

#include <reg51.h>

sbit mybit=P2^4;

void main(void)

{

while (1)

{

mybit=1; //turn on P2.4

mybit=0; //turn off P2.4

}

}

**Write an 8051 C program to get a byte of data form P0. If it is less**

**than 100, send it to P1; otherwise, send it to P2.**

Solution:

#include <reg51.h>

void main(void)

{

unsigned char mybyte;

P0=0xFF; //make P0 input port

while (1)

{

mybyte=P0; //get a byte from P0

if (mybyte<100)

P1=mybyte; //send it to P1

else

P2=mybyte; //send it to P2

}

}

**Write an 8051 C program to monitor bit P1.5. If it is high, send 55H**

**to P0; otherwise, send AAH to P2.**

Solution:

#include <reg51.h>

sbit mybit=P1^5;

void main(void)

{

mybit=1; //make mybit an input

while (1)

{

if (mybit==1)

P0=0x55;

else

P2=0xAA;

}

}

**A door sensor is connected to the P1.1 pin, and a buzzer is connected**

**to P1.7. Write an 8051 C program to monitor the door sensor, and**

**when it opens, sound the buzzer. You can sound the buzzer by**

**sending a square wave of a few hundred Hz.**

Solution:

#include <reg51.h>

void MSDelay(unsigned int);

sbit Dsensor=P1^1;

sbit Buzzer=P1^7;

void main(void)

{

Dsensor=1; //make P1.1 an input

while (1)

{

while (Dsensor==1)//while it opens

{

Buzzer=0;

MSDelay(200);

Buzzer=1;

MSDelay(200);

}

}

}

**Write an 8051 C program to turn bit P1.5 on and off 50,000 times.**

Solution:

sbit MYBIT=0x95;

void main(void)

{

unsigned int z;

for (z=0;z<50000;z++)

{

MYBIT=1;

MYBIT=0;

}

**Write an 8051 C program to get the status of bit P1.0, save it, and**

**send it to P2.7 continuously.**

Solution:

#include <reg51.h>

sbit inbit=P1^0;

sbit outbit=P2^7;

bit membit; //use bit to declare

//bit- addressable memory

void main(void)

{

while (1)

{

membit=inbit; //get a bit from P1.0

outbit=membit; //send it to P2.7

}

}

**Write an 8051 C program to toggle all the bits of P0 and P2**

**continuously with a 250 ms delay. Using the inverting and Ex-OR**

**operators, respectively**.

Solution:

#include <reg51.h>

void MSDelay(unsigned int);

void main(void)

{

P0=0x55;

P2=0x55;

while (1)

{

P0=~P0;

P2=P2^0xFF;

MSDelay(250);

}

}

**Write an 8051 C program to get bit P1.0 and send it to P2.7 after**

**inverting it.**

Solution:

#include <reg51.h>

sbit inbit=P1^0;

sbit outbit=P2^7;

bit membit;

void main(void)

{

while (1)

{

membit=inbit; //get a bit from P1.0

outbit=~membit; //invert it and send

//it to P2.7

}

}

**Write an 8051 C program to read the P1.0 and P1.1 bits and issue an**

**ASCII character to P0 according to the following table.**

**P1.1 P1.0**

**0 0 send ‘0’ to P0**

**0 1 send ‘1’ to P0**

**1 0 send ‘2’ to P0**

**1 1 send ‘3’ to P0**

Solution:

#include <reg51.h>

void main(void)

{

unsignbed char z;

z=P1;

z=z&0x3;

switch (z)

{

case(0):

{

P0=‘0’;

break;

}

case(1):

{

P0=‘1’;

break;

}

case(2):

{

P0=‘2’;

break;

}

case(3):

{

P0=‘3’;

break;

}

}

}

**Write an 8051 C program to convert packed BCD 0x29 to ASCII and**

**display the bytes on P1 and P2.**

Solution:

#include <reg51.h>

void main(void)

{

unsigned char x,y,z;

unsigned char mybyte=0x29;

x=mybyte&0x0F;

P1=x|0x30;

y=mybyte&0xF0;

y=y>>4;

P2=y|0x30;

}

**Write an 8051 C program to convert ASCII digits of ‘4’ and ‘7’ to**

**packed BCD and display them on P1.**

Solution:

#include <reg51.h>

void main(void)

{

unsigned char bcdbyte;

unsigned char w=‘4’;

unsigned char z=‘7’;

w=w&0x0F;

w=w<<4;

z=z&0x0F;

bcdbyte=w|z;

P1=bcdbyte;

}

**Write an 8051 C program to calculate the checksum byte for the data**

**25H, 62H, 3FH, and 52H.**

Solution:

#include <reg51.h>

void main(void)

{

unsigned char mydata[]={0x25,0x62,0x3F,0x52};

unsigned char sum=0;

unsigned char x;

unsigned char chksumbyte;

for (x=0;x<4;x++)

{

P2=mydata[x];

sum=sum+mydata[x];

P1=sum;

}

chksumbyte=~sum+1;

P1=chksumbyte;

}

**Write an 8051 C program to perform the checksum operation to**

**ensure data integrity. If data is good, send ASCII character ‘G’ to P0.**

**Otherwise send ‘B’ to P0.**

Solution:

#include <reg51.h>

void main(void)

{

unsigned char mydata[]

={0x25,0x62,0x3F,0x52,0xE8};

unsigned char shksum=0;

unsigned char x;

for (x=0;x<5;x++)

chksum=chksum+mydata[x];

if (chksum==0)

P0=‘G’;

else

P0=‘B’;

}

**Write an 8051 C program to convert 11111101 (FD hex) to decimal**

**and display the digits on P0, P1 and P2**.

Solution:

#include <reg51.h>

void main(void)

{

unsigned char x,binbyte,d1,d2,d3;

binbyte=0xFD;

x=binbyte/10;

d1=binbyte%10;

d2=x%10;

d3=x/10;

P0=d1;

P1=d2;

P2=d3;

}