

```

#include "1_ones_processor_header.h"
#include "1_ones_processor_header_2.h"

#define zero 0
#define one 1
#define two 2
#define three 3
#define four 4
#define five 5
#define six 6
#define seven 7

char reg_bkp[8];
char update_reg;
char bit_num, keypress;
char digits[8];

int main(void){
char mode;

setup_HW;

String_to_PC_Basic("\r\n\r\nManipulating one of the PORT registers\r\n");

User_prompt_Basic;
if(User_response == 'r')

{String_to_PC_Basic\
("\r\nStep 1 Set up CPU register\r\n\
Type bit names\r\n\
z, o, tw, th, fo, fi, si or se ('x' when done)");

do{
if(select_bits())
{cpu_reg_2 = 1;Display_registers;_delay_ms(500);
cpu_reg_2 = (cpu_reg_2 << bit_num); Display_registers;_delay_ms(500);

cpu_reg_1 |= cpu_reg_2;
cpu_reg_2 = 0;
Display_registers;}
}while (keypress != 'x');

String_to_PC_Basic("\r\n\r\nStep 2 Copy to PORT register ak\r\n");
waitforkeypress_Basic();
port_reg = cpu_reg_1;
cpu_reg_1 = 0;
Display_registers;}

if(User_response == 'R'){String_to_PC_Basic("\r\nport_reg |= (1 << one) | (1 << two) | (1 << five) | (1 << six);\r\n");
port_reg |= (1 << one) | (1 << two) | (1 << five) | (1 << six);Display_registers;}

String_to_PC_Basic("\r\nTo manipulate bits\r\n\t\
send 'r' to clear one\r\n\t\
's' to set one,\r\n\t\
'c' to change one and\r\n\t\
't' to test one");

while(1){

mode = waitforkeypress_Basic();

System_response;

if (mode == 'x')continue;

select_bit_to_process();

switch(mode){
case 'r': clear_bit_1_text; cpu_reg_1 = ~cpu_reg_1;
clear_bit_2_text;port_reg = port_reg & cpu_reg_1; break;

case 's': set_bit_text; port_reg = port_reg | cpu_reg_1;break;
case 'c': change_bit_text; port_reg = port_reg ^ cpu_reg_1;break;
case 't': cpu_reg_2 = port_reg & cpu_reg_1;test_bit_text; break;}

update_display;}}

/*****/
void select_bit_to_process(void){
while(!(select_bits()));
{cpu_reg_1 |= (1 << bit_num);
Display_registers;}}

/*****/

```

```

/*****/
char set_update(char bit_num)
{if(!(reg_bkp[bit_num])){update_reg = 0;reg_bkp[bit_num] = 1;}
else update_reg = 1;return update_reg;}

/*****/
char select_bits(void){
    update_reg = 0;
    keypress = waitforkeypress_Basic();
    switch(keypress){
        case 'z': bit_num = zero; set_update(0); break;
        case 'o': bit_num = one; set_update(1); break;
        case 't': switch(waitforkeypress_Basic()){
            case 'w': bit_num = two; set_update(2); break;
            case 'h': bit_num = three; set_update(3); break;
            default: update_reg = 1; break;}break;
        case 'f': switch(waitforkeypress_Basic()){
            case 'o': bit_num = four; set_update(4); break;
            case 'i': bit_num = five; set_update(5); break;
            default: update_reg = 1; break;}break;
        case 's': switch(waitforkeypress_Basic()){
            case 'i': bit_num = six; set_update(6); break;
            case 'e': bit_num = seven; set_update(7);break;
            default: update_reg = 1; break;} break;
        case 'x': for(int m = 0; m <=7; m++)reg_bkp[m] = 0;
        default: update_reg = 1; break;}
        if(!(update_reg))return 1; else return 0; }

/*****/
```