

7 Examples of GPIO Programming

7.1 SuperIO Model: Winbond W83627UHG

- GPIO OUT use GP 60~63
- GPIO IN use GP 20~23

7.2 W83627UHG Access index port: 4Eh/4Fh

- Index Address Port: 4Eh
- Index Data Port: 4Fh

7.3 Configure GPIO register sequence

1. Enter the extended function mode
2. Select logic device number 8
3. Activate the logic device GPIO Port 6
4. Configure GPIO Port 6 register
5. Select logic device number 9
6. Activate the logic device GPIO Port 2
7. Configure GPIO Port 2 register
8. Exit the extended function mode

7.4 Read/write GPIO sequence

1. Enter the extended function mode
2. Select logic device number
3. Read/write GPIO register value
4. Exit the extended function mode

7.5 Software programming example

- **Enter the extended function mode**

Writing 87h to index address port twice will enter the extended function mode.

Example x86 assembly code:

```
mov    dx, 4Eh
mov    al, 87h
out    dx, al
out    dx, al
```

Example C code:

```
outportb(0x4E, 0x87);
outportb(0x4E, 0x87);
```

- **Exit the extended function mode**

Writing AAh to index address port will exit the extended function mode.

Example x86 assembly code:

```
mov    dx, 4Eh
mov    al, 0AAh
out    dx, al
```

Example C code:

```
outportb(0x4E, 0xAA);
```

- **Select logic device number**

Example x86 assembly code:

```
mov    dx, 4Eh
mov    al, 007h    ;LDN selection register
out    dx, al
mov    dx, 4Fh
mov    al, 008h    ;Select LDN=8, GPIO Port6
                        ;or Select LDN9, GPIO Port2
out    dx, al
```

Example C code:

```
outportb(0x4E, 0x07);    //LDN selection register
outportb(0x4F, 0x08);    //Select LDN=8, GPIO Port6
or
outportb(0x4E, 0x07);    //LDN selection register
outportb(0x4F, 0x09);    //Select LDN=9, GPIO Port2
```

- **Activate the logic device**

Example x86 assembly code:

```
mov    dx, 4Eh
mov    al, 030h    ;Logic device activation control reg.
out    dx, al
mov    dx, 4Fh
in     al, dx
or     al, 004h    ;Set bit2 to enable GPIO Port6 if LDN=8
or     al, 002h    ;Set bit1 to enable GPIO Port2 if LDN=9
out    dx, al
```

Example C code:

```
outportb(0x4E, 0x30);    //Logic device activation control
outportb(0x4F, (inportb(0x4F)|0x2));
```

//Set bit[1] to enable GPIO Port2 if LDN=9

Or

```
outportb(0x4E, 0x30); //Logic device activation control
outportb(0x4F, (inportb(0x4F)|0x4));
```

//Set bit[2] to enable GPIO Port6 if LDN=8

- **Configure GPIO register**

Example x86 assembly code:

```
mov    dx, 4Eh
mov    al, 0E6h    ;GPIO inversion reg.

out    dx, al
mov    dx, 4Fh
mov    al, 000h    ;0 – normal, 1 - inverted
out    dx, al

mov    dx, 4Eh
mov    al, 0E4h    ;GPIO I/O selection reg.
out    dx, al
mov    dx, 4Fh
mov    al, 0FFh    ;0 – Output, 1 – Input
                    ;or mov al, 0F0h to set output
out    dx, al
```

Example C code:

```
outportb(0x4E, 0xE6); //GPIO I/O selection reg.
outportb(0x4F, 0x0); //0 – normal, 1 - inverted
```

```
outportb(0x4E, 0xE4); //GPIO inversion reg.
outportb(0x4F, 0xFF); //0 – Output, 1 - Input
```

or

```
outportb(0x4E, 0xE4); //GPIO inversion reg.
outportb(0x4F, 0xF0); //0 – Output, 1 - Input
```

- **Read GPIO value**

Example x86 assembly code:

```
mov    dx, 4Eh
mov    al, 0E5h    ;GPIO data reg.
out    dx, al
mov    dx, 4Fh
in     al, dx      ;Bit[3::0] = GPI[3::0] value
```

Example C code:

```
outportb(0x4E, 0xE5); //GPIO data reg.
GP = inportb(0x4F);    //Bit[3::0] = GPI[3::0]
```

- **Write GPIO value**

Example x86 assembly code:

```

;Set GPO62
    mov    dx, 4Eh
    mov    al, 0E5h    ;GPIO data reg.
    out    dx, al
    mov    dx, 4Fh
    in     al, dx
or     al, 00000100b    ;Bit2 = GPO62
    out    dx, al

```

```

;Clear GPO62
    mov    dx, 4Eh
    mov    al, 0E5h    ;GPIO data reg.
    out    dx, al
    mov    dx, 4Fh
    in     al, dx
    and    al, not 00000100b
    out    dx, al

```

Example C code:

```

//Set GPO62
    outportb(0x4E, 0xE5);                //GPIO data reg.
    Outportb(0x4F, (inportb(0x4F)|0x4)); //Set Bit[2]

//Clear GPO62
    outportb(0x4E, 0xE5);                //GPIO data reg.
    Outportb(0x4F, (inportb(0x4F)&0xFB)); //Clear Bit[2]

```

The followings are C language source code:

```

#include "stdio.h"
#include "conio.h"

//Super I/O index access port
#define INDEXP    0x4E
#define DATAP    0x4F

//Enter super I/O programming mode
#define ENTERPRG { \
    outportb(INDEXP, 0x87); \
    outportb(INDEXP, 0x87);}

//Super I/O index write

```

* Specifications subject to change without notice. All brand names and trademark are their registered properties. *

```

#define WRITEREG(reg,val) { \
    outportb(INDEXP, reg); \
    outportb(DATAP, val);}

//Exit super I/O programming mode
#define EXITPRG { \
    outportb(INDEXP, 0xAA);}

//Select logic device number
#define SELETDEV(ldn) { \
    outportb(INDEXP, 7); \
    outportb(DATAP, ldn); }

//Initialize the GPIO port2
int InitGP2() {

    //Start the super I/O chip programming
    ENTERPRG

    //Select the logical device 9, GP2
    SELETDEV(9)

    //Activate GP1
    WRITEREG(0x30, (inportb(0x30)|0x2))

    WRITEREG(0xE6, 0x0)
    WRITEREG(0xE4, 0xFF)

    //Exit the super I/O chip programming
    EXITPRG

    return 0;
}

//Initialize the GPIO port6
int InitGP6() {

    //Start the super I/O chip programming
    ENTERPRG

    //Select the logical device 8, GP6
    SELETDEV(8)

    //Activate GP1

```

* Specifications subject to change without notice. All brand names and trademark are their registered properties. *

```

WRITEREG(0x30, (inportb(0x30)|0x4))

WRITEREG(0xE6, 0x0)
WRITEREG(0xE4, 0xF0)

//Exit the super I/O chip programming
EXITPRG

return 0;
}

```

```

//Read GPIO Port2
unsigned char ReadGP2() {
    unsigned char cGP2;

    //Start the super I/O chip programming
    ENTERPRG

    //Select the logical device 9, GP2
    SELETDEV(9)

    //Read GPIO Value
    outportb(INDEXP, 0xE5);
    cGP2 = inportb(DATAP);
    cGP2 = cGP2 & 0xF;

    //Exit the super I/O chip programming
    EXITPRG

    return cGP2;
}

```

```

//Write GPIO Port6
int WriteGP6(unsigned char cGP6) {

    //Start the super I/O chip programming
    ENTERPRG

    //Select the logical device 8, GP6
    SELETDEV(8)

    //Write GP1 value
    WRITEREG(0xE5, cGP6)
}

```

```
//Exit the super I/O chip programming
EXITPRG

return 0;
}

int main() {
    unsigned char cGP;

    //Initialize the GPIO port
    InitGP2();
    InitGP6();

    //Read GPIO Port 2
    cGP = ReadGP2();
    printf("\nRead GPIO Port 2 Status: %X", cGP);

    //Write GPIO Port 6
    WriteGP6(cGP);
    printf("\nSet GPIO Port 6 Status: %X", cGP);

    return 0;
}
```