

How to Use **GPIO** Souce Code

1. Enter GPIO Information in GpioTab.txt File: (format as shown in the red box in the diagram)

```
##### Note #####
#1. Please write in the format PINx = ICH_xx or PINx = SIO_xx.
# "PIN1 = SIO_1" indicates that GPIO pin 1 is SIO's GPIO1.
# "#" indicates a single line comment.
# "ICH" denotes GPIO from the southbridge chip.
# "SIO" denotes GPIO from the Super I/O.

# J_GPIO1
PIN1 = ICH_20 # Southbridge GPIO
PIN2 = ICH_21
PIN3 = ICH_22
PIN4 = ICH_23
PIN6 = ICH_24
PIN7 = ICH_25
PIN8 = ICH_26
PIN9 = ICH_27
```

2. Select Operating System: Modify TEST.cpp as shown in the diagram.

<pre>#define OS_DOS 1 #define OS_WINDOWS 0 #define OS_LINUX 0</pre>	DOS
<pre>#define OS_DOS 0 #define OS_WINDOWS 1 #define OS_LINUX 0</pre>	WINDOWS
<pre>#define OS_DOS 0 #define OS_WINDOWS 0 #define OS_LINUX 1</pre>	LINUX

3. Compile:

For DOS: Install UESTudio 06 and Borland C 3.1, set the compile path, double-click " TEST.prj" and compile directly.

For Windows: Install VC6.0, double-click "TEST.dsw" to open the project, and compile directly.

For Linux: Install g++ 4.4 and make, go to the source directory, and enter make.

Appendix: Windows Dynamic Link Library

Dynamic Link Library:

●long _stdcall IMCLoadDriver(long);

// Load the GPIO driver

●long _stdcall IMCRemoveDriver(long);

// Unload the GPIO driver

●long _stdcall IoWrite8(UINT16 addr,UINT8 value);

// Write value to the address in the IO space.

●UINT8 _stdcall IoRead8(UINT16 addr);

// Return the value from the address in the IO space.

●UINT32 _stdcall PciCfgRead32(UINT16 Bus, UINT16 Dev, UINT16 Func, UINT16 Reg);

// Read the register Reg of the PCI device at the address (bus, dev, func).