Using GPIO on the Vortex86

Summary
Application Note
AP0100 (v1)

How to configure and drive the GPIO pins on the Vortex86SX and Vortex86DX processors.

These DOS code examples were written and compile using Borland Turbo C++. There is a download link for this compiler at http://cc.embarcadero.com/item/26014

All the GPIO pins on the Vortex86 are independent and can be configured as inputs or outputs.

Not all Vortex86 modules provide access to all of the GPIO ports. Table 1 includes addresses for all ports for completeness.

The ports are all controlled by reading and writing from register pairs in the I/O space of the Vortex86. The register locations are shown in Table 1.

	Port 0	Port 1	Port 2	Port 3	Port4
Data Register Address	78h	79h	7Ah	7Bh	7Ch
Direction Register Address	98h	99h	9Ah	9Bh	9Ch

Table 1

For each of the registers, the 8 bits are mapped to the respective GPIO pins. The direction register determines the direction of each GPIO pin:

- 0 = input mode
- 1=output mode

Example values to write into register for port 0:

- Writing 00h to register 98h makes all pins input
- Writing FFh makes all pins output
- Writing F0h makes pins [7-4] 0utputs and pins [3-0] inputs

The data register is a read/write register; again the 8 GPIO pins are mapped to the 8 bits of the register. A logic 1 denotes the pin is high and a logic 0 denotes the pin low.

Only pins set to output are affected by a write operation, the status of all the pins can be read from the register, regardless of direction.

Example DOS program

```
#include <dos.h>

#define PORT_0_DATA 0x78
#define PORT_0_DIR 0x98

void main (void)
{
unsigned char ReadValue;

// set all port 0 pins to input
       outportb(PORT_0_DIR,0x00);
// read in the value
       ReadValue = inportb(PORT_0_DATA);

// Now drive pins on port 0

// First set all port 0 pins to output
       outportb(PORT_0_DIR,0xff);

// drive all pins high
       outportb(PORT_0_DATA,0xff);

// drive all pins low
       outportb(PORT_0_DATA,0x00);

// put the pattern 01010101 out to GPIO 0
       outportb(PORT_0_DATA,0x55);

// example of reading the port first to enable just a single pin to be changed
       ReadValue=inportb(PORT_0_DATA);  //read in current status of all pins
       ReadValue = (ReadValue | 0x02);  // set bit we wish to change
       outportb(PORT_0_DATA,ReadValue);
       // bit pattern will now be 01010111

} // end of main
```