

Int 15 emulator driver for Windows.

1. Int15 Hardware

Resources:

1. EEPROM - 2K size
000h-3FFh - reserved
400h-7FFh - free for user data
2. Temperature sensor
3. Watchdog hardware.

Access to these resources under DOS can be provided by INT 15h function, see user manual.

Access under Windows 98, ME , 2000, XP can be provided over "Int15dl"-WDM driver, under Windows-NT "Int15dl"-NT driver.

For the moment this driver support all Digital-Logic boards with PIIX4 and ICH4 chipsets.

2. Driver Installation (Windows-2000)

"Int15dl" is not plug and play driver, it must be installed manually.

1. Open "Control Panel".
2. Doubleclick on "Add/Remove Hardware".
3. To continue click "Next>" button.
4. On the page "Choose a Hardware Task" check "Add/Troubleshoot a device" and click "Next>" button.
5. After "New hardware detection" windows automatic procedure, choose "Add a new device" item and click "Next>" button.
6. In the "Find New Hardware" page choose "No, I want to select the hardware from a list" and click "Next>" button.
7. Choose "Other devices" in the "Hardware Type" list and click "Next>" button.
8. On the page "Select a Device Driver" press "Have Disk..." button and find the driver location (Int15dl.inf - WDM). After opening the "inf" file, installation program will show a Models list and "Digital-Logic INT15 functions emulator" string. Then press "Next>" button.
9. Then press "Finish" button, it's not necessary to restart a computer after installation.
10. After installation, please, be sure, that "Digital-Logic INT15 functions emulator" has been installed properly. Open "Control Panel", then doubleclick on "System" icon. Choose tabsheet "Hardware" and click on "Device Manager" button. Then expand "System Devices" and doubleclick on "Digital-Logic INT15 functions emulator". Be sure, that device is working properly.

Driver Installation (Windows-NT)

1. Boot with administrative privileges.
2. Copy NT-driver "Int15dl.sys" into WINNT/System32/drivers folder.
3. Register the driver – "doubleclick" on "int15dl.reg" file.
4. Reboot a computer.

3. Programming Int15dl interface under Windows:

Programming of the Int15dl Interface is very similar to DOS programming, based on DeviceIoControl function, which operate with predefined structure named "Registers".

Files:

Int15srv.h contains definitions for Registers structure.

Int15dlioctl.h contains definition for IO control code constants.

Test_Int15dl.cpp - example of subroutines, which provide access to hardware functions over Int15dl driver.

Functions (Test Int15dl.cpp):

bool Int15(Registers *Regs) - the main function, which send user request to driver.

Returns **true**, if request finished successfully, otherwise **false**.

Regs - address of Registers structure, which contains specific data of request (defined in Int15srv.h).

For example, the following code will initiate temperature measuring:

```
Registers Regs;
Regs.ah = 0xEC;
if(!Int15(&Regs)) //error in driver request
{
    printf("Error reading temperature\n");
    return;
}
//success - temperature value is in Regs.al
if(Regs.bl == 0)printf("\tTemperature = %d C\n",Regs.al);
//error - not valid value
else printf("\tError reading Temperature\n");
```

Note: Input and output arguments of Int15 function for different chipsets and BIOSes are different, please, read the user manual about registers definition.

For example: To get temperature value on the board with PIIX4 chipset you have to use "Regs.ah = 0xEC;", but on the board with ICH4 chipset, please use "Regs.ax = 0x78EC;".

bool Open_Int15dl(void) - the first function, which must be called to create a link between "Digital-Logic INT15 functions emulator" driver and user software. It returns **"true"**, if device was successfully opened, otherwise - **"false"**.

void Close_Int15dl(void) - the last function, which breaks a link between driver and user software.

int GetChipID(void) - additional service function, returns the type of chipset - for PIIX4 = 4, for ICH4 = 5.

Registers structure:

It's used for exchanging information between user program and "Int15dl" driver.

```
typedef struct Registers {  
    union {  
        struct {  
            unsigned short ax;  
            unsigned short bx;  
            unsigned short cx;  
            unsigned short dx;  
            unsigned short bp;  
            unsigned short si;  
            unsigned short di;  
            unsigned short ds;  
            unsigned short es;  
            unsigned short flags;  
        };  
        struct {  
            unsigned char al;  
            unsigned char ah;  
            unsigned char bl;  
            unsigned char bh;  
            unsigned char cl;  
            unsigned char ch;  
            unsigned char dl;  
            unsigned char dh;  
        };  
    };  
} TRegisters;
```

4. Information for advanced users:

1. At first call of the function **Open_Int15dl()**, Int15dl driver try to detect the type of chipset. To disable this procedure user have to define the next parameters in "Int15dl.inf" file before installation of the driver:

for PIIX4 chipset:

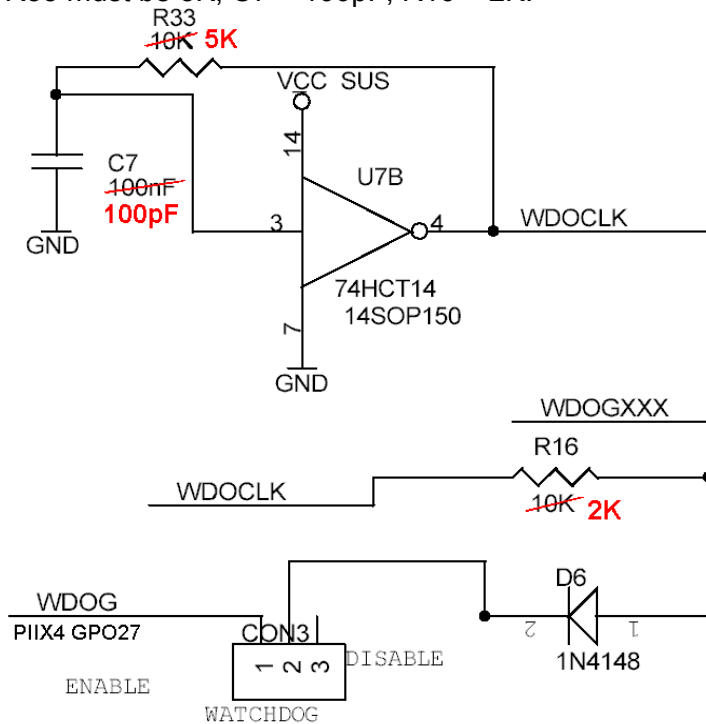
```
HKR, "Parameters", "chipID", 0x00010001, 0x4
HKR, "Parameters", "pmBase", 0x00010001, 0x1000
HKR, "Parameters", "smbBase", 0x00010001, 0x1040
HKR, "Parameters", "tsaddr", 0x00010001, 0x9E - LM75 sensor address
```

for ICH4 chipset:

```
HKR, "Parameters", "chipID", 0x00010001, 0x5
HKR, "Parameters", "pmBase", 0x00010001, 0x1000
HKR, "Parameters", "smbBase", 0x00010001, 0x1880
HKR, "Parameters", "tsaddr", 0x00010001, 0x9C - ADM1023 sensor address
```

2. WatchDOG "toggle" function may not work properly on the PIIX4 boards with old values of the components: R33, C7, R16 (regarding MPC board).

R33 must be 5K, C7 = 100pF, R16 = 2K.



WatchDOG reset timeout for the boards with PIIX4 chipset is defined by hardware and has a value about 600mS.

In this case, please get in contact with the Digital-Logic support:

support@digitallogic.ch