Министерство образования Республики Беларусь

Учреждение образования

«Белорусский государственный университет информатики и радиоэлектроники»

Кафедра электронных вычислительных машин

Дисциплина: Интерфейсы и устройства вычислительных машин

Лабораторная работа №3

«Дисковые накопители»

Выполнил: Проверил:

Ст. гр. 150504 Селезнев А. И.

Горбачевский К. В.

Минск, 2023

1. Постановка задачи

Вывести данные о дисках, подключённых к компьютеру: модель, изготовитель, серийный номер, версия прошивки, сведения о памяти (свободно/занято/всего), тип интерфейса, список поддерживаемых режимов. Подключение к шине реализовать через порты ввода-вывода.

Вывод информации о подключенных USB-устройствах не делать. Подключение к шине производить с помощью готовых библиотек нельзя. Подключение к шине реализовать с применением портов ввода-вывода.

1. Краткие теоретические сведения

Жесткий диск (HDD, Hard Disk Drive) - это запоминающее устройство, основанное на принципе магнитной записи.

Принцип работы жёстких дисков похож на работу магнитофонов. Рабочая поверхность диска движется относительно считывающей головки (например, в виде катушки индуктивности с зазором в магнитопроводе). При подаче переменного электрического тока (при записи) на катушку головки возникающее переменное магнитное поле из зазора головки воздействует на ферромагнетик поверхности диска и изменяет направление вектора намагниченности доменов в зависимости от величины сигнала. При считывании перемещение доменов у зазора головки приводит к изменению магнитного потока в магнитопроводе головки, что приводит к возникновению переменного электрического сигнала в катушке за счёт электромагнитной индукции.

SSD (твердотельные накопители) - это запоминающее устройство, которое использует микросхемы памяти NAND Flash или DRAM вместо магнитных пластин и других механических компонентов, которые применяются в стандартных жестких дисках (HDD).

Технология объединения дисков в массив – RAID. Суть идеи: дорогостоящие серверные диски большого объема можно заменить набором дешевых и не столь надежных винчестеров настольного класса за счет усложнения логики доступа к ним со стороны контроллера.

ATA также называемый IDE - стандарт на интерфейс между компьютером (контроллером) и накопителем на жестких магнитных дисках (дисководом, HDD), включая:

• физический уровень (разъёмы, кабели);

• электрические и логические характеристики сигналов;

• регистры устройства;

• команды и протоколы.

1. Код программы

#include "stdafx.h"

#include "hexioctrl.h"

#include <stdio.h>

#include <conio.h>

#include <iostream>

#include <string.h>

#define IDENTIFY\_DEVICE 0xEC

#define IDENTIFY\_PACKET\_DEVICE 0xA1

const int dataRegister[2] = {0x1F0, 0x170};

const int DH\_register[2] = {0x1F6, 0x176};

const int StateCommandRegister[2] = {0x1F7, 0x177};

const int altStateRegister[2] = {0x3F6, 0x376};

unsigned short data[256];

void waitDeviceBusy(int channelNum);

bool getDeviceInfo(int devNum, int channelNum);

void showTable();

bool waitReady(int channelNum);

int main()

{

ALLOW\_IO\_OPERATIONS;

std::cout << " Model | Serial Number |Type | Size | PIO | Multiword DMA | Ultra DMA |" << std::endl;

std::cout << "-------------------------------------------------------------------------------------------------------------------------------" << std::endl;

for (int channel = 0; channel <= 1; channel++ )

for (int device = 0; device <= 1; device++ )

{

if(getDeviceInfo(device, channel))

{

showTable();

}

}

system("pause");

return 0;

}

bool waitReady(int channelNum)

{

for (int i = 0; i < 1000; i++)

{

unsigned char state = \_inp(altStateRegister[channelNum]);

if(state & (1 << 6)) return true;

}

return false;

}

void waitDeviceBusy(int channelNum)

{

unsigned char state;

do state = \_inp(altStateRegister[channelNum]);

while (state & (1 << 7));

}

bool getDeviceInfo(int devNum, int channelNum)

{

const int commands[2] = {IDENTIFY\_PACKET\_DEVICE, IDENTIFY\_DEVICE};

for (int i = 0; i < 2; i++)

{

waitDeviceBusy(channelNum);

unsigned char regData = (devNum << 4) + (7 << 5); //111X0000

\_outp(DH\_register[channelNum], regData); //device in DH

if(!waitReady(channelNum)) return false;

\_outp(StateCommandRegister[channelNum], commands[i]);

waitDeviceBusy(channelNum);

}

for( int i = 0; i < 256; i++ )

data[i] = \_inpw(dataRegister[channelNum]);

return true;

}

void showTable()

{

//model

for(int i = 27; i <= 46; i++)

printf("%c%c", data[i] >> 8, data[i] & 0x00FF );

std::cout<<"|";

//serial number

for( int i = 10; i <= 19; i++ )

printf("%c%c", data[i] >> 8, data[i] & 0x00FF );

std::cout<<"|";

int isAtapi = data[0] & (1 << 15);

if(isAtapi)

printf("ATAPI");

else

printf("ATA ");

std::cout<<"|";

if(!isAtapi)

{

printf("%lu", ((unsigned long \*)data)[30] >> 1);

}

else

printf(" ");

std::cout <<"|";

printf("[%s%s", (data[64] & 1) ? "+" : "-" ,"] PIO 3 |" );

std::cout << std::endl;

printf("\t\t\t\t\t\t\t\t\t |[%s%s", (data[64] & 2) ? "+" : "-" ,"] PIO 4 |" );

std::cout << std::endl;

printf("\t\t\t\t\t\t\t\t\t\t\t |[%s%s", (data[63] & 1) ? "+" : "-" ,"] MWDMA 0 |" );

std::cout << std::endl;

printf("\t\t\t\t\t\t\t\t\t\t\t |[%s%s", (data[63] & 2) ? "+" : "-" ,"] MWDMA 1 |" );

std::cout << std::endl;

printf("\t\t\t\t\t\t\t\t\t\t\t |[%s%s", (data[63] & 4) ? "+" : "-" ,"] MWDMA 2 |" );

std::cout << std::endl;

printf("\t\t\t\t\t\t\t\t\t\t\t\t\t\t |[%s%s", (data[88] & 1) ? "+" : "-" ,"] UDMA Mode 0 |" );

std::cout << std::endl;

printf("\t\t\t\t\t\t\t\t\t\t\t\t\t\t |[%s%s", (data[88] & (1 << 1)) ? "+" : "-" ,"] UDMA Mode 1 |" );

std::cout << std::endl;

printf("\t\t\t\t\t\t\t\t\t\t\t\t\t\t |[%s%s", (data[88] & (1 << 2)) ? "+" : "-" ,"] UDMA Mode 2 |" );

std::cout << std::endl;

printf("\t\t\t\t\t\t\t\t\t\t\t\t\t\t |[%s%s", (data[88] & (1 << 3)) ? "+" : "-" ,"] UDMA Mode 3 |" );

std::cout << std::endl;

printf("\t\t\t\t\t\t\t\t\t\t\t\t\t\t |[%s%s", (data[88] & (1 << 4)) ? "+" : "-" ,"] UDMA Mode 4 |" );

std::cout << std::endl;

printf("\t\t\t\t\t\t\t\t\t\t\t\t\t\t |[%s%s", (data[88] & (1 << 5)) ? "+" : "-" ,"] UDMA Mode 5 |" );

std::cout << std::endl;

}

1. **Тестирование программы**

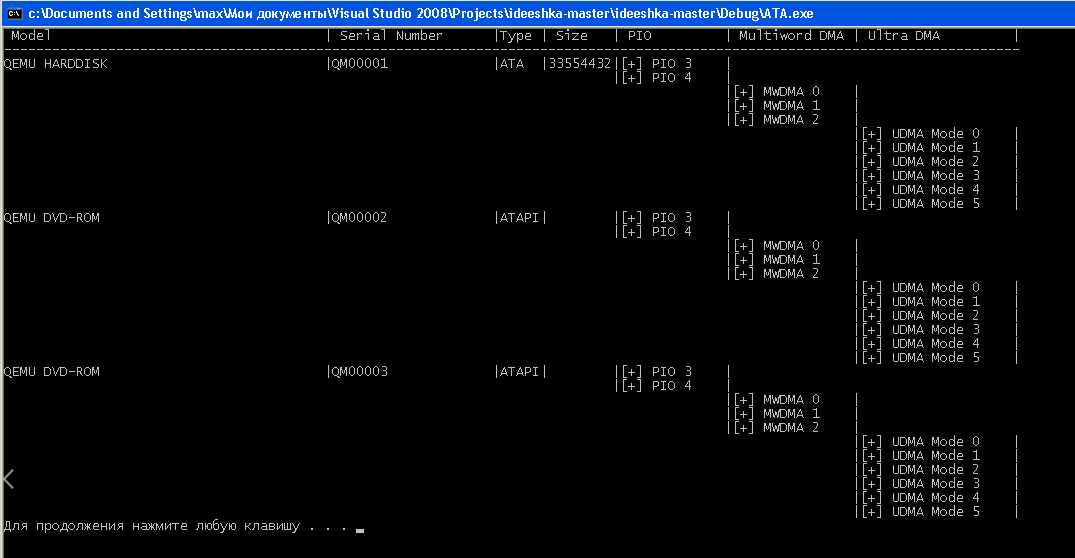
****

Рисунок 4.1 – Результат работы программы

1. **Вывод**

В ходе выполнения данной лабораторной работы были изучены теоретические основы дисковых накопителей, возможности работы с ними в операционной системе Windows XP и способы доступа ко всем подключённым к компьютеру дискам.

Программа написана для операционной системы Windows ХР, использованные инструменты – язык программирования C++, Microsoft Visual Studio 2008, предоставленные драйверы.