БГУИР

Кафедра ЭВМ

Лабораторная работа №1

Асинхронная двунаправленная побайтная передача данных

Выполнил студент группы 950503 Проверил ассистент кафедры ЭВМ

Сякачёв П.В. Марцинкевич В.А.

Минск 2021

Цель: разработать модуль асинхронной побайтной передачи данных, соответствующий физическому уровню модели OSI, на основе последовательных интерфейсов RS-232 и RS-485.

Модуль должен быть оформлен в виде библиотеки функций или класса. Разработанный модуль будет использоваться в следующих лабораторных работах в качестве приемопередатчика.

Для физического подключения по стандарту RS-232 используются девятиконтактные разъемы.

Традиционное назначение цифровых цепей RS-232:

-SOUT – выход передатчика

-SIN – вход приемника

-RTS – сигнал-запрос от UART к модему о передачи байта

-CTS – сигнал-подтверждение от модема к UART о готовности принять байт

-DTR – сигнал от UART к модему о готовности к взаимодействию

-DSR – сигнал от модема к UART о готовности к взаимодействию

-DCD - сигнал от модема к UART об обнаружении данных

-RI – сигнал об обнаружении входящего телефонного звонка

Использование пары RTS/CTS позволяет передавать в одном направлении, а использование DSR/DTR в обратном.

Поскольку передатчик и приемник, не имеют общего источника времени, в канале используется синхросигнал. Исходя из этого выделяются два режима обмена:

1. Асинхронный – синхронизируется посылка каждого информационного байта.
2. Синхронный – синхронизируется весь информационный поток.

По своей сути, передатчик и приемник COM-порта работают как сдвиговые регистры. Cначала данные записываются параллельно в регистр, а затем при помощи сдвигов последовательно передаются при помощи синхроимпульса. Заполнение приемника так же происходит автоматически. Таким образом, если байты записываются слишком быстро, а считываются медленно, то возникает переполнение очереди передатчика. Если наоборот, то после переполнения очереди приемника происходит потеря данных.

Тактирование происходит при помощи встроенного бод-генератора. (Один бод равен одному сигналу в секунду).

Адаптер Rs-485 является продолжением стандарта RS-232. Главными различиями являются:

1. Способ передачи данных – дифференциальная пара.
2. Максимальное количество передатчика, приемников увеличено до 32.
3. Так же увеличены максимальные пропускная способность и расстояние.

int Server()

{

HANDLE readEnd = CreateEvent(NULL, FALSE, FALSE, L"readEnd");

HANDLE writeEnd = CreateEvent(NULL, FALSE, FALSE, L"writeEnd");

HANDLE hExit = CreateEvent(NULL, FALSE, FALSE, L"exit");

HANDLE hWork = CreateEvent(NULL, FALSE, FALSE, L"work");

HANDLE hFile = CreateFile(L"COM1", GENERIC\_WRITE | GENERIC\_READ, 0, NULL, OPEN\_EXISTING, 0, NULL);

if (hFile == NULL)

{

printf("Error\n");

exit(GetLastError());

}

SetCommMask(hFile, EV\_RXCHAR);

SetupComm(hFile, 1500, 1500);

COMMTIMEOUTS CommTimeOuts;

CommTimeOuts.ReadIntervalTimeout = 0xFFFFFFFF;

CommTimeOuts.ReadTotalTimeoutMultiplier = 0;

CommTimeOuts.ReadTotalTimeoutConstant = 1200;

CommTimeOuts.WriteTotalTimeoutMultiplier = 0;

CommTimeOuts.WriteTotalTimeoutConstant = 1200;

if (!SetCommTimeouts(hFile, &CommTimeOuts))

{

CloseHandle(hFile);

hFile = INVALID\_HANDLE\_VALUE;

return 1;

}

DCB ComDCM;

memset(&ComDCM, 0, sizeof(ComDCM));

ComDCM.DCBlength = sizeof(DCB);

GetCommState(hFile, &ComDCM);

ComDCM.BaudRate = setBaudrate();

ComDCM.ByteSize = 8;

ComDCM.Parity = NOPARITY;

ComDCM.StopBits = ONESTOPBIT;

ComDCM.fAbortOnError = TRUE;

ComDCM.fDtrControl = DTR\_CONTROL\_DISABLE;

ComDCM.fRtsControl = RTS\_CONTROL\_DISABLE;

ComDCM.fBinary = TRUE;

ComDCM.fParity = FALSE;

ComDCM.fInX = FALSE;

ComDCM.fOutX = FALSE;

ComDCM.XonChar = 0;

ComDCM.XoffChar = (unsigned char)0xFF;

ComDCM.fErrorChar = FALSE;

ComDCM.fNull = FALSE;

ComDCM.fOutxCtsFlow = FALSE;

ComDCM.fOutxDsrFlow = FALSE;

ComDCM.XonLim = 128;

ComDCM.XoffLim = 128;

if (!SetCommState(hFile, &ComDCM))

{

CloseHandle(hFile);

hFile = INVALID\_HANDLE\_VALUE;

return 2;

}

PORT1(readEnd, writeEnd, hFile, hExit, hWork);

CloseHandle(hFile);

return 0;

}

Данный метод предназначен для инициализации COM порта при помощи структуры DCB.

В начале метода открывается порт для чтения/записи и инициализируются структуры для асинхронного чтения/записи. Далее происходит начальная настройка порта.

string read(HANDLE hFile)

{

char buf = '\0';

string msg;

DWORD bytesReaded;

do

{

if (!ReadFile(hFile, &buf, 1, &bytesReaded, NULL))

cout << GetLastError() << endl;

if (buf == '\n') break;

msg += buf;

} while (1);

return msg;

}

bool write(HANDLE hFile, string buffer)

{

int i = 0;

cout << endl;

while (i < buffer.size())

{

WriteFile(hFile, &(buffer[i++]), 1, NULL, NULL);

}

if (buffer == "exit\n")

return false;

else

return true;

}

Данные методы отвечают за асинхронные чтения/запись данных.

Чтение/запись происходит при помощи функций ReadFile/WriteFile.

Вывод: в ходе выполнения данной лабораторной работы был получен опыт в разработке программного модуля(библиотеки функций) для асинхронной побайтной передачи данных.