

**Выполнила: Белоусова Е., ИП-911**

## **Задача**

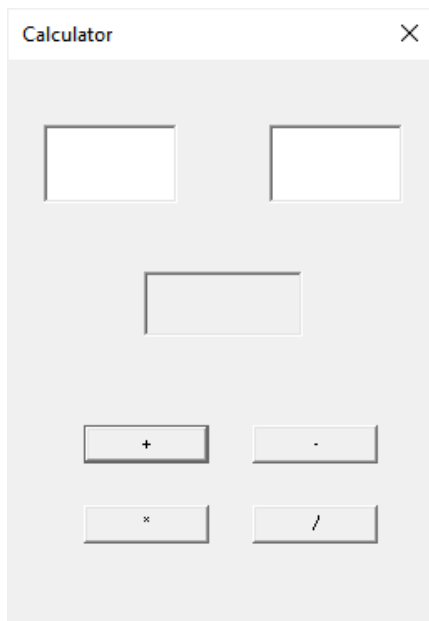
**Цель:** приобрести навыки разработки Windows приложений с использованием низкоуровневого интерфейса прикладного программирования — Windows API; освоиться с некоторыми механизмами взаимодействия ОС MS Windows с выполняемыми пользовательскими приложениями.

**Задание 2:** написать примитивный калькулятор с интерфейсом, представленным диалоговым окном.

**Задание 3:** организовать передачу данных на основе WM\_COPYDATA между двумя приложениями с диалоговыми окнами.

## **Описание работы программы**

Интерфейс примитивного калькулятора представлен на рисунке:



*Рисунок 1 Калькулятор*

Имеется два поля для ввода операндов, одно поле со свойством read only для вывода результата и четыре кнопки для разных операций. Результат появляется после нажатия на одну из кнопок с операцией.

Если не был введен один из операндов, то по умолчанию переменным присвоены нулевые значения.

Калькулятор поддерживает расчеты с вещественными числами.

Выбранная операция остается выделенной, до выбора другой операции.



*Рисунок 2 Операция*

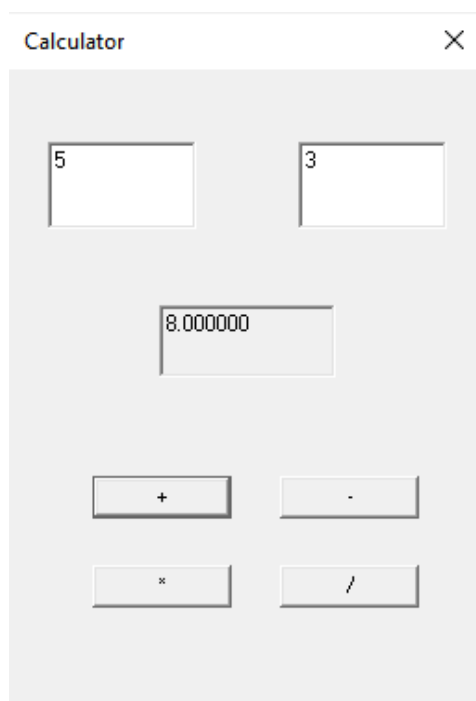


Рисунок 3 Пример вычисления

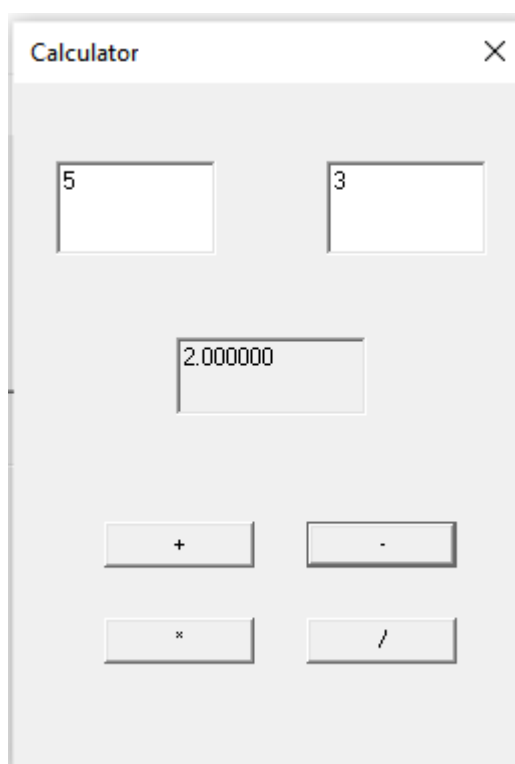


Рисунок 4 Пример вычисления

Программа с помощью функции `atof` преобразует содержимое полей в числовые значения.

В следующем задании нужно организовать передачу данных на основе `WM_COPYDATA` между двумя приложениями с диалоговыми окнами.

Пусть клиентом будет калькулятор, а для хоста сделаем отдельное диалоговое окно с полем, в которое будет передаваться результат и кнопкой, чтобы можно было отправить сообщение в клиент.

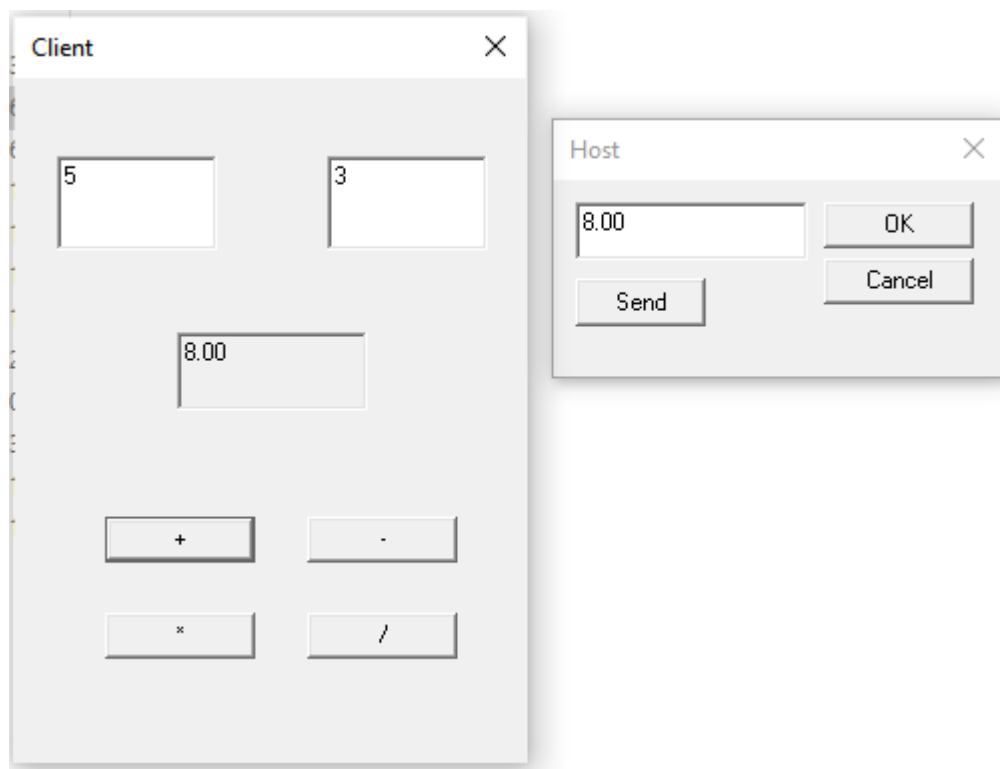


Рисунок 5 Обмен между приложениями

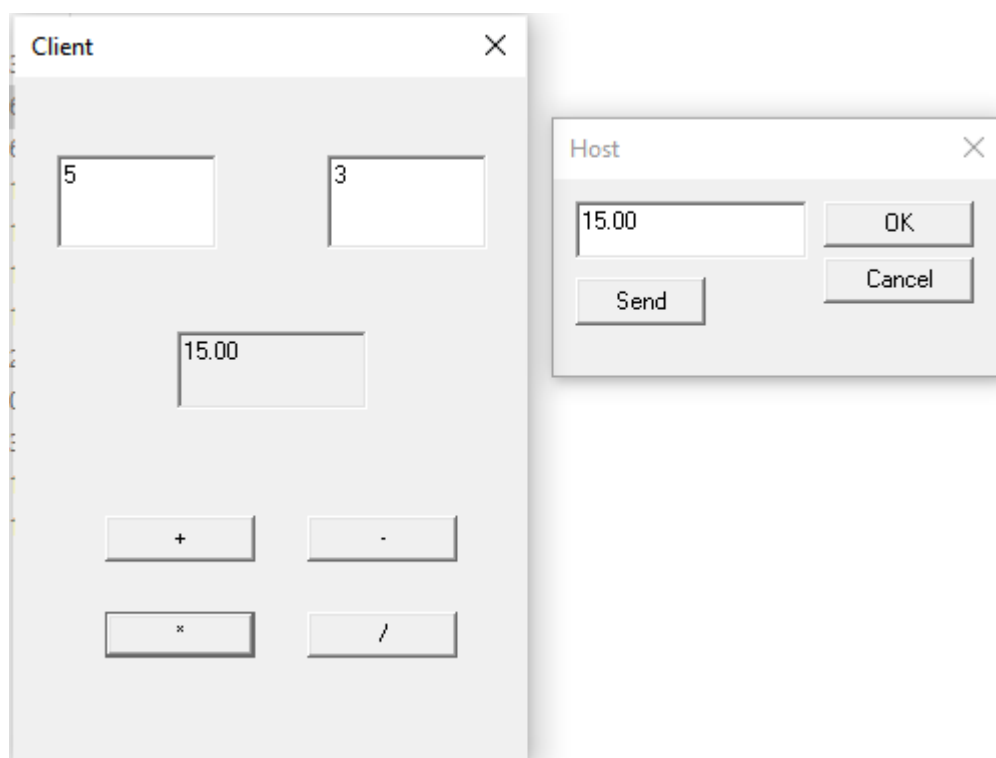


Рисунок 6 Обмен между приложениями

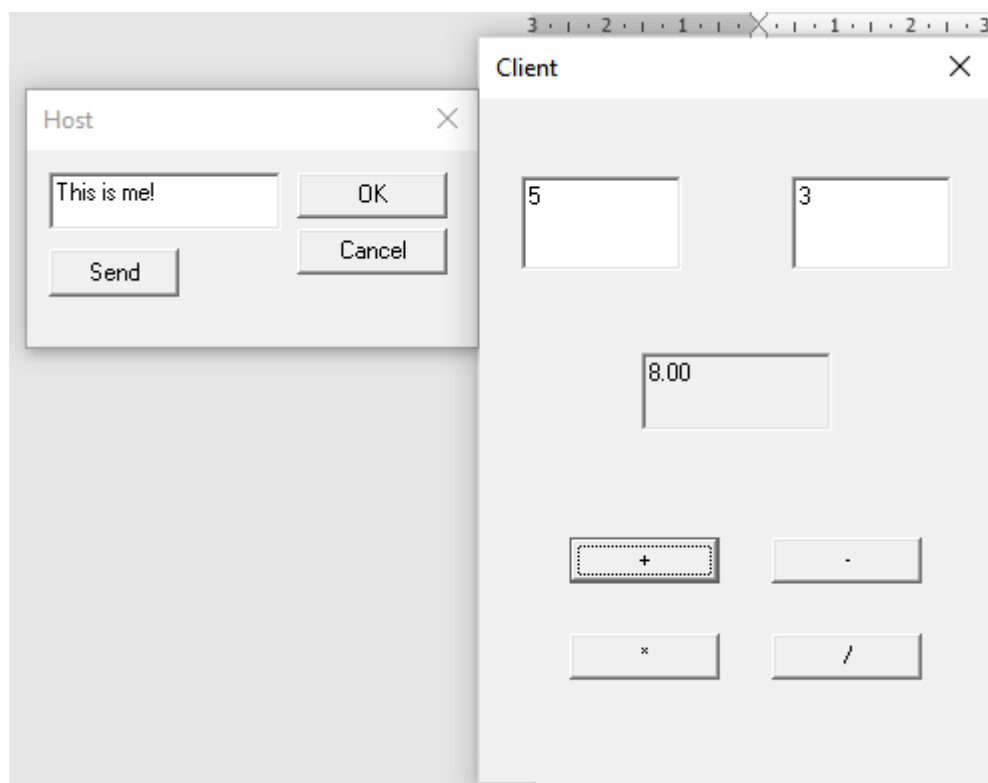


Рисунок 7 До обмена данными

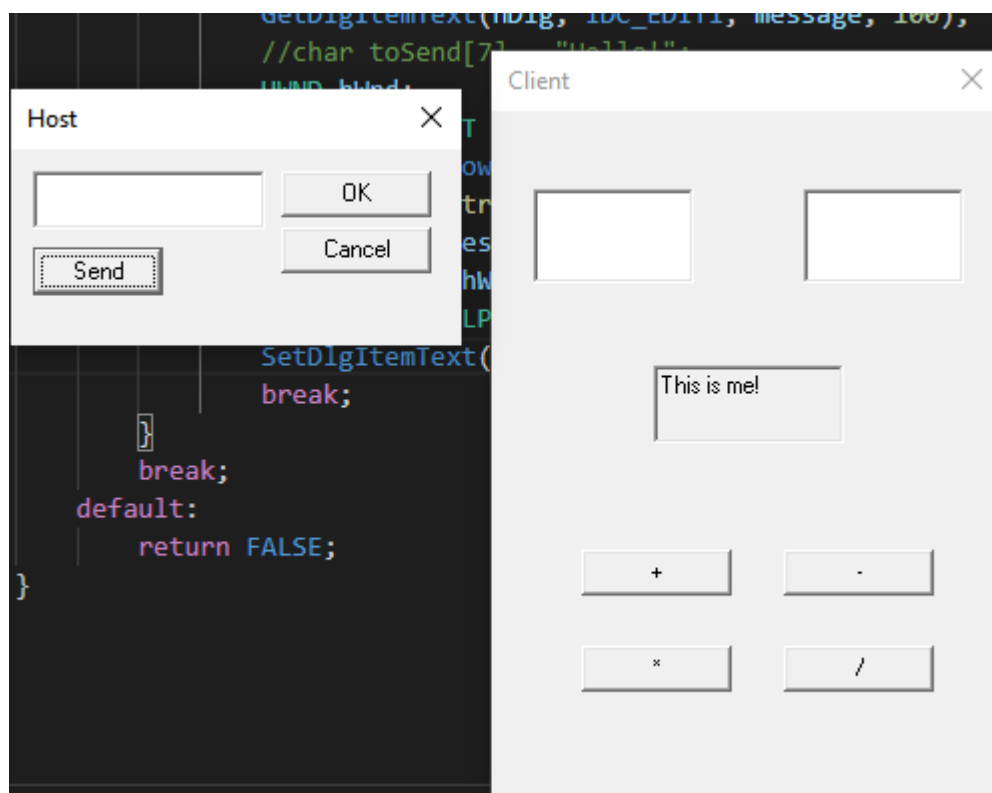


Рисунок 8 После обмена

## Листинг

//d1.c

```
#include <windows.h>
```

```
#include <stdio.h>
```

```
#include "resource.h"
```

```
LRESULT CALLBACK DlgProc(HWND hDlg, UINT message, WPARAM wParam, LPARAM lParam);
```

```
int WINAPI WinMain(HINSTANCE hInstance,
```

```
    HINSTANCE hPrevInstance,
```

```
    LPSTR lpCmdLine,
```

```
    int nCmdShow)
```

```
{
```

```
    MSG msg;
```

```
    DialogBox( hInstance, (LPCTSTR)IDD_DLGTST, NULL, (DLGPROC)DlgProc);
```

```
    while (GetMessage(&msg, NULL, 0, 0))
```

```
    {
```

```
        TranslateMessage(&msg);
```

```
        DispatchMessage(&msg);
```

```
    }
```

```
    return msg.wParam;
```

```
}
```

```
LRESULT CALLBACK DlgProc( HWND hDlg, UINT message, WPARAM wParam,
```

```
    LPARAM lParam )
```

```
{
```

```
    char op1[100];
```

```
    char op2[100];
```

```
    char result[100];
```

```
    double res;
```

```
    switch (message)
```

```
    {
```

```
        case WM_INITDIALOG:
```

```
            return TRUE;
```

```
        case WM_COMMAND:
```

```

switch (LOWORD(wParam))
{
    case IDOK:
        PostQuitMessage(0);
        return TRUE;

    case IDCANCEL:
        PostQuitMessage(0);
        return TRUE;

    case IDC_BUTTON1:
        GetDlgItemText(hDlg, IDC_EDIT1, op1, 100);
        GetDlgItemText(hDlg, IDC_EDIT2, op2, 100);
        res = atof(op1) + atof(op2);
        sprintf(result, "%f", res);
        SetDlgItemText(hDlg, IDC_EDIT3, result);
        return TRUE;

    case IDC_BUTTON2:
        GetDlgItemText(hDlg, IDC_EDIT1, op1, 100);
        GetDlgItemText(hDlg, IDC_EDIT2, op2, 100);
        res = atof(op1) - atof(op2);
        sprintf(result, "%f", res);
        SetDlgItemText(hDlg, IDC_EDIT3, result);
        return TRUE;

    case IDC_BUTTON3:
        GetDlgItemText(hDlg, IDC_EDIT1, op1, 100);
        GetDlgItemText(hDlg, IDC_EDIT2, op2, 100);
        res = atof(op1) * atof(op2);
        sprintf(result, "%f", res);
        SetDlgItemText(hDlg, IDC_EDIT3, result);
        return TRUE;

    case IDC_BUTTON4:
        GetDlgItemText(hDlg, IDC_EDIT1, op1, 100);
        GetDlgItemText(hDlg, IDC_EDIT2, op2, 100);

```

```

        res = atof(op1) / atof(op2);

        sprintf(result, "%f", res);

        SetDlgItemText(hDlg, IDC_EDIT3, result);

        return TRUE;

// case IDC_BTN1:

//  GetDlgItemText(hDlg, IDC_EDIT1, strText, 100);

//  SetDlgItemText(hDlg, IDC_EDIT2, strText);

//  SetDlgItemText(hDlg, IDC_EDIT1, "");

//  break;

    }

    break;

default:

    return FALSE;

}

}

//d1.rc

#include <windows.h>

#include "resource.h"


IDD_DLGTEST DIALOG DISCARDABLE 0, 0, 170, 210
STYLE DS_MODALFRAME | DS_FIXEDSYS | DS_CENTER | WS_POPUP | WS_CAPTION |
WS_SYSMENU
CAPTION "Calculator"
FONT 8, "MS Sans Serif"
BEGIN
    PUSHBUTTON    "+",IDC_BUTTON1,30,135,50,14
    PUSHBUTTON    "- ",IDC_BUTTON2,97,135,50,14
    PUSHBUTTON    "*",IDC_BUTTON3,30,164,50,14
    PUSHBUTTON    "/",IDC_BUTTON4,97,164,50,14
    EDITTEXT      IDC_EDIT1,14,24,53,29,ES_AUTOHSCROLL
    EDITTEXT      IDC_EDIT2,104,24,53,29,ES_AUTOHSCROLL
    EDITTEXT      IDC_EDIT3,54,78,63,24,ES_AUTOHSCROLL | ES_READONLY

```

END

//resource.h

```
#define IDD_DLGTEST          101
#define IDC_BUTTON1          1001
#define IDC_BUTTON2          1002
#define IDC_BUTTON3          1003
#define IDC_BUTTON4          1004
#define IDC_EDIT1            1005
#define IDC_EDIT3            1006
#define IDC_EDIT2            1007
```

//client.c

#include <windows.h>

#include "resource.h"

#include <stdio.h>

LRESULT CALLBACK

DlgProc(HWND hDlg, UINT message, WPARAM wParam, LPARAM lParam);

int WINAPI WinMain(HINSTANCE hInstance,

    HINSTANCE hPrevInstance,

    LPSTR lpCmdLine,

    int nCmdShow) {

    MSG msg;

    DialogBox(hInstance,(LPCTSTR)IDD\_DLGTEST,NULL,

        (DLGPROC)DlgProc);

    while (GetMessage(&msg, NULL, 0, 0)) {

        TranslateMessage(&msg);

        DispatchMessage(&msg);

    }

    return msg.wParam;

}

LRESULT CALLBACK

DlgProc(HWND hDlg, UINT message, WPARAM wParam, LPARAM lParam) {



```

char strText[100], strText2[100], result[100];

float res;

switch (message) {
    case WM_INITDIALOG:
        return TRUE;

    case WM_COPYDATA:
        SetDlgItemText(hDlg, IDC_EDIT3, (char*)(
            ((COPYDATASTRUCT*)lParam)->lpData));

        break;

    case WM_COMMAND:
        switch (LOWORD(wParam) ) {
            case IDOK:
                PostQuitMessage(0);
                return TRUE;

            case IDCANCEL:
                PostQuitMessage(0);
                return TRUE;

            case IDC_BUTTON1:
                GetDlgItemText(hDlg, IDC_EDIT1, strText, 100);
                GetDlgItemText(hDlg, IDC_EDIT2, strText2, 100);
                res = atof(strText) + atof(strText2);
                sprintf(result, "%0.2f", res);
                SetDlgItemText(hDlg, IDC_EDIT3, result);
                break;

            case IDC_BUTTON2:
                GetDlgItemText(hDlg, IDC_EDIT1, strText, 100);
                GetDlgItemText(hDlg, IDC_EDIT2, strText2, 100);
                res = atof(strText) - atof(strText2);
                sprintf(result, "%0.2f", res);

```

```

        SetDlgItemText(hDlg, IDC_EDIT3, result);

        break;

        case IDC_BUTTON3:

            GetDlgItemText(hDlg, IDC_EDIT1, strText, 100);

            GetDlgItemText(hDlg, IDC_EDIT2, strText2, 100);

            res = atof(strText) * atof(strText2);

            sprintf(result, "%0.2f", res);

            SetDlgItemText(hDlg, IDC_EDIT3, result);

            break;

        case IDC_BUTTON4:

            GetDlgItemText(hDlg, IDC_EDIT1, strText, 100);

            GetDlgItemText(hDlg, IDC_EDIT2, strText2, 100);

            res = atof(strText) / atof(strText2);

            sprintf(result, "%0.2f", res);

            SetDlgItemText(hDlg, IDC_EDIT3, result);

            break;

    }

    break;

default:

    return FALSE;

}

HWND hWnd;

COPYDATASTRUCT data;

hWnd=FindWindow(NULL,"Host");

data.cbData=strlen(result);

data.lpData=result;

SendMessage (hWnd, WM_COPYDATA, (WPARAM)GetFocus(), (LPARAM)&data);

return TRUE;

}

//host.c

#include <windows.h>

```

```
return TRUE;
```

```

        case IDCANCEL:

            PostQuitMessage(0);

            return TRUE;

        case IDC_BTN1:

            //SetDlgItemText(hDlg, IDC_EDIT1, "");

            char message[100];

            GetDlgItemText(hDlg, IDC_EDIT1, message, 100);

            //char toSend[7] = "Hello!";

            HWND hWnd;

            COPYDATASTRUCT data;

            hWnd=FindWindow(NULL,"Client");

            data.cbData=strlen(message);

            data.lpData=message;

            SendMessage (hWnd, WM_COPYDATA, (LPARAM)GetFocus(),

                        (LPARAM)&data);

            SetDlgItemText(hDlg, IDC_EDIT1, "");

            break;

    }

    break;

default:

    return FALSE;

}

} //d1.rc

#include <windows.h>

#include "resource.h"

IDD_DLGTEST DIALOG DISCARDABLE 0, 0, 170, 210

STYLE DS_MODALFRAME | WS_POPUP | DS_CENTER| WS_CAPTION |

WS_SYSMENU

CAPTION "Client"

FONT 8, "MS Sans Serif"

BEGIN

```

```

PUSHBUTTON "+" , IDC_BUTTON1, 30, 135, 50, 14
PUSHBUTTON "-" , IDC_BUTTON2, 97, 135, 50, 14
PUSHBUTTON "*" , IDC_BUTTON3, 30, 164, 50, 14
PUSHBUTTON "/" , IDC_BUTTON4, 97, 164, 50, 14
EDITTEXT IDC_EDIT1, 14, 24, 53, 29, ES_AUTOHSCROLL
EDITTEXT IDC_EDIT2, 104, 24, 53, 29, ES_AUTOHSCROLL
EDITTEXT IDC_EDIT3, 54, 78, 63, 24, ES_AUTOHSCROLL | ES_READONLY
END

//d2.rc

#include <windows.h>
#include "host_resource.h"

IDD_HOSTTEST DIALOG DISCARDABLE 0, 0, 150, 60
STYLE DS_MODALFRAME | WS_POPUP | WS_CAPTION |
WS_SYSMENU
CAPTION "Host"
FONT 8, "MS Sans Serif"
BEGIN
    DEFPUSHBUTTON "OK", IDOK, 90, 7, 50, 14
    PUSHBUTTON "Cancel", IDCANCEL, 90, 24, 50, 14
    PUSHBUTTON "Send", IDC_BTN1, 7, 30, 43, 15
    EDITTEXT IDC_EDIT1, 7, 7, 77, 18, ES_AUTOHSCROLL
END

//resource.h

#define IDD_DLGTTEST 101
#define IDC_BUTTON1 1001
#define IDC_BUTTON2 1002
#define IDC_BUTTON3 1003
#define IDC_BUTTON4 1004
#define IDC_EDIT1 1005
#define IDC_EDIT3 1006
#define IDC_EDIT2 1007

```

```
// host_resource.h  
  
#define IDD_HOSTTEST 101  
  
#define IDC_EDIT1 1000  
  
#define IDC_BTN1 1002
```