Una duda frecuente y que se explica en este pequeño tutorial:

Bien, primero deben saber que necesitamos un compilador que soporte la biblioteca windows.h puede ser el Borland C++ 5.5 o el Dev-C++ 4.9.9.2.

Para abrir la unidad de CD usamos la siguiente porción de código:

```
codeDivStart()MCI_OPEN_PARMS mciParams;
DWORD dwFlags;
LPCTSTR szCharDrive = "D";

memset(&mciParams, 0, sizeof(MCI_OPEN_PARMS));
mciParams.lpstrDeviceType = (LPCSTR) MCI_DEVTYPE_CD_AUDIO;
mciParams.lpstrElementName = szCharDrive;

dwFlags= MCI_OPEN_TYPE | MCI_OPEN_TYPE_ID;

if (!mciSendCommand(0, MCI_OPEN, dwFlags, (DWORD) mciParams))
{
    mciSendCommand(mciParams.wDeviceID, MCI_SET, MCI_SET_DOOR_OPEN | MCI_WAIT, 0);
    mciSendCommand(mciParams.wDeviceID, MCI_CLOSE, MCI_WAIT, 0);
}
```

Como siempre un poco de explicación:

- Para abrir o cerrar la unidad de CD se usa la función mciSendCommand. Primero hay que crear una variable de tipo MCI_OPEN_PARAMS.

- De esta estructura, nos interesa lpstrDeviceType, a la que le asignaremos MCI_DEVTYPE_CD_AUDIO y lpstrElementName, a la que le asignaremos el nombre de la unidad (D, E, F, etc).

- Luego, creamos un DWORD para las banderas, que serán MCI_OPEN_TYPE |

MCI_OPEN_TYPE_ID .

- Llamamos la función mciSendCommand. El primer parámetro será 0, el segundo, MCI_OPEN , el tercero las banderas en el DWORD y el cuarto, la dirección de memoria de la estructura MCI_OPEN_PARAMS , casteada a DWORD .

- Si la función regresa false, todo fué bien, y ahora abrimos el dispositivo llamando a mciSendCommand , donde el primer parámetro será el miembro wDeviceID de nuestra estructura MCI_OPEN_PARAMS , el segundo parámetro será MCI_SET , el tercero será MCI_SET_DOOR_OPEN para abrir, o MCI_SET_DOOR_CLOSED para cerrar; el cuarto parámetro será 0 .

- Finalmente, cerramos el manejador con mciSendCommand, donde el segundo parámetro será MCI_CLOSE y el tercero, MCI_WAIT .

Con el código anterior se abre la unidad de CD. Si se cambia MCI_SET_DOOR_OPEN por MCI_SET_DOOR_CLOSED , se cierra (aunque esto depende del tipo de CDROM instalado; por ejemplo, el cerrar el CDROM no funciona con la mayoría de las laptops).

A continuación viene el código fuente completo para abrir y cerrar la unidad de CD:

codeDivStart()/* Fichero: cd.c */

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

```
void AbrirCD();
void CerrarCD();

int main()
{
    printf("Presione ENTER para abrir CD");
    getchar();
    printf("\nAbriendo CD...");
    AbrirCD();

    printf("\n\nPresione ENTER para cerrar CD");
    getchar();
    printf("\nCerrando CD...\n\n");
    CerrarCD();

    system("PAUSE");

    return 0;
}

void AbrirCD()
{
    MCI_OPEN_PARMS mciParams;
    DWORD dwFlags;
    LPCTSTR szCharDrive = "D";

    memset(&mciParams, 0, sizeof(MCI_OPEN_PARMS));
    mciParams.lpstrDeviceType = (LPCSTR) MCI_DEVTYPE_CD_AUDIO;
    mciParams.lpstrElementName = szCharDrive;

    dwFlags= MCI_OPEN_TYPE | MCI_OPEN_TYPE_ID;

    if (!mciSendCommand(0, MCI_OPEN, dwFlags, (DWORD) &mciParams))
    {
        mciSendCommand(mciParams.wDeviceID, MCI_SET, MCI_SET_DOOR_OPEN | MCI_WAIT, 0
        mciSendCommand(mciParams.wDeviceID, MCI_CLOSE, MCI_WAIT, 0);
    }
}

void CerrarCD()
{
    MCI_OPEN_PARMS mciParams;
    DWORD dwFlags;
    LPCTSTR szCharDrive = "D";

    memset(&mciParams, 0, sizeof(MCI_OPEN_PARMS));
```

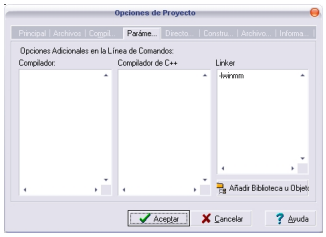
```
    mciParams.lpstrDeviceType = (LPCSTR) MCI_DEVTYPE_CD_AUDIO;
    mciParams.lpstrElementName = szCharDrive;

    dwFlags= MCI_OPEN_TYPE | MCI_OPEN_TYPE_ID;

    if (!mciSendCommand(0, MCI_OPEN, dwFlags, (DWORD) &mciParams))
    {
        mciSendCommand(mciParams.wDeviceID, MCI_SET, MCI_SET_DOOR_CLOSED|MCI_WAIT,(
        mciSendCommand(mciParams.wDeviceID, MCI_CLOSE, MCI_WAIT, 0);
    }
}
```

Consideraciones Finales

- Si usas Dev-C++ 4.9.9.2 primero hay que crear un proyecto nuevo y colocas el código anterior, luego vas a Proyecto -> Opciones del proyecto -> Pará metros
y en el cuadro
Linker
escribes
-lwinmm
tal como se muestra a continuación:



- Si usas el Borland C++ 5.5 puedes compilar el código sin problemas, pero para ejecutarlo es mejor hacerlo dando doble click sobre el programa generado desde el Explorador de Windows