

MultiPro.DLL

This document contains the instructions that can be send to the multipro programmer. Normally you only need to detect the programmer (with DetectProgrammer or AutoDetect) and set the programmer in a certain mode with the Set???? instruction. The comport is automatically opened and closed by an instruction. The instruction/command can only be send if the comport is not used/opened by an other program.

How can you implement these routines in your program? Make sure that the application can access 'MultiPro.dll' and insert the lines (from below) into your (Delphi) program and call the function. That's it.

Known problem: You can't use these functions at Windows start-up. The dll can't communicate with the programmer at that time.

Note : MultiPro.exe does not use the dll-file. The dll-file is added for software developers.

```
Function DetectProgrammer(ComPort:Byte):Boolean; stdcall; external  
'MultiPro.dll';  
{Detect if the programmer is on Serial port <ComPort> (com1 to com16).  
Returns true when detected correctly.}
```

```
Function AutoDetect:Byte; stdcall; external 'MultiPro.dll';  
{Autodetects the comport of the connected multipro programmer. Returns 0 if  
no programmer is connected.}
```

```
{The following instructions set the programmer in a certain mode. Return  
true if the instruction was successfully send.}
```

```
Function SetIdle(ComPort:Byte):Boolean; stdcall; external 'MultiPro.dll';
```

```
Function SetPhoenix358(ComPort:Byte):Boolean; stdcall; external  
'MultiPro.dll';
```

```
Function SetPhoenix600(ComPort:Byte):Boolean; stdcall; external  
'MultiPro.dll';
```

```
Function SetSmartMouse358(ComPort:Byte):Boolean; stdcall; external  
'MultiPro.dll';
```

```
Function SetSmartMouse600(ComPort:Byte):Boolean; stdcall; external  
'MultiPro.dll';
```

```
Function SetPonyProg358(ComPort:Byte):Boolean; stdcall; external  
'MultiPro.dll';
```

```
Function SetPonyProg600(ComPort:Byte):Boolean; stdcall; external  
'MultiPro.dll';
```

```
Function SetJDM(ComPort:Byte):Boolean; stdcall; external 'MultiPro.dll';
```

```
Function GetProgrammerName(ComPort:Byte):ShortString; stdcall; external  
'MultiPro.dll';  
{Returns the product name with version. Eg. 'MultiPro V1.0'}
```

```
Function CheckCard(ComPort:Byte):Boolean; stdcall; external 'MultiPro.dll';  
{Checks if card is in the cardslot. Returns true when a card is in the  
cardslot.}
```

```
Function SendInstruction(ComPort,Command,SubCommand:Byte):Boolean; stdcall; external  
'MultiPro.dll';  
{Sends an instruction to the programmer. Returns true if the instruction is  
successfully send to the programmer.}
```

Function **GetResult**(ComPort,Command,SubCommand:Byte;NoChecksum:Boolean):ShortString;
stdcall; external 'MultiPro.dll';
{Send an instruction (command, subcommand) to the programmer. The answer of this
instruction is returned as a string. The answer contains a checksum character on the
end. Set NoChecksum to False to exclude this character for the result returned by
GetResult.}

Function **GetProgrammerMode**(ComPort:Byte):Integer; stdcall; external 'MultiPro.dll';
{Returns the program mode of the programmer.

These values can be returned:

-1 : not a valid mode or not connected
0 : Idle mode
1 : Phoenix 3.58MHz mode
2 : Phoenix 6.00MHz mode
3 : SmartMouse 3.58MHz mode
4 : SmartMouse 6.00MHz mode
5 : Ponyprog 3.58MHz mode
6 : Ponyprog 6.00MHz mode
7 : JDM mode}