

AN047**Application Note zu Multi-LAB/2 und MODULAR-4/486****Funktionen zur Verwaltung der
Registrierdatenbankeinträge unter Windows NT**

Autor: MIH

AN047.DOC (6 Seiten)

INHALT:

1. Allgemeines	2
2. Datenstrukturen	2
2.1. Allgemeine Treiberparameter (Struktur REGDRV PARA)	2
2.2. Kartenparameter (Struktur REGCARD PARA)	2
3. Schnittstellenfunktionen	3
3.1. Allgemeine Funktionen	3
3.2. Dateienverwaltung	3
3.3. Einträge in der Registrierdatenbank vornehmen	4
3.4. Einträge aus der Registrierdatenbank löschen	5
3.5. Einträge aus der Registrierdatenbank lesen	5
4. Anwendung der Funktionen	6
4.1. Installation des Treibers	6
4.2. Deinstallation des Treibers	6

1. Allgemeines

Hardwarekonfigurationen werden unter Windows NT zentral in der Registrierdatenbank des Systems gehalten. Die für SORCUS-Karten notwendigen Einträge, wie z.B. Kartentyp, I/O-Adresse und Interruptkanal, werden mit Hilfe eines Installierungstools durchgeführt. Die dort eingetragenen Parameter werden von dem Gerätetreiber 'MLXDRV.SYS' verwendet.

Innerhalb von Softwarepaketen kann es erwünscht sein, die Installation von SORCUS-Karten zu integrieren. Um dabei vom Aufbau der Registrierdatenbankeinträge unabhängig zu bleiben, wird dem Anwender eine definierte Schnittstelle von Funktionen zur Verfügung gestellt, welche einerseits Informationen über die aktuellen Einträge liefern und andererseits Einträge einrichten.

Im dieser Application Note werden Datenstrukturen, Schnittstellenfunktionen und das Vorgehen bei der Installation des Treibers erläutert.

2. Datenstrukturen

Die folgenden Datenstrukturen werden innerhalb der Schnittstellenfunktionen verwendet. Das Alignment der Strukturen ist in der Headerdatei fest vorgegeben.

2.1. Allgemeine Treiberparameter (Struktur REGDRV PARA)

<i>Typ</i>	<i>Name</i>	<i>Erklärung</i>
ULONG	NumberOfProcess	im Augenblick nicht benutzt
ULONG	LogBufferSize	im Augenblick nicht benutzt
ULONG	Reserve1	reserviert
ULONG	Reserve2	reserviert

2.2. Kartenparameter (Struktur REGCARD PARA)

<i>Typ</i>	<i>Name</i>	<i>Erklärung</i>
ULONG	Active	Gibt an, ob die folgende Struktur ausgewertet wird
ULONG	CardNumber	Nummer der zu installierenden Karte
ULONG	CardType	Kartentyp (ML2=2; ML8=8)
ULONG	InterruptChannel	Interruptkanal
ULONG	BaseAddress	I/O-Basisadresse
ULONG	SrqBufferSize	im Augenblick nicht benutzt
ULONG	Reserve1	reserviert
ULONG	Reserve2	reserviert

3. Schnittstellenfunktionen

Die Schnittstellenfunktionen werden in der DLL 'MLXINST.DLL' zur Verfügung gestellt. Die Funktionsprototypen sind in 'MLXINST.H' enthalten.

Die meisten Funktionen liefern einen Fehlerwert zurück. Ist der Rückgabewert ungleich null, so entspricht der Code dem eines Systemfehlers, wie er z.B. von der Funktion GetLastError() zurückgegeben wird.

Die aktuelle Version unterstützt das Betriebssystem Windows NT. Eine Anpassung an Windows 95 ist vorgesehen.

3.1. Allgemeine Funktionen

mlx_inst_get_os Betriebssystem ermitteln

C: long mlx_inst_get_os(void);

Funktion: Das momentan aktive Betriebssystem wird ermittelt. Die Rückgabewerte entsprechen den in der Win32-API definierten Konstanten.

Parameter:

RETURN:	VER_PLATFORM_WIN32_WINDOWS:	Windows 95
	VER_PLATFORM_WIN32_WINNT:	Windows NT

mlx_inst_drv_installed Prüfen, ob Treiber installiert ist

C: long mlx_inst_drv_installed(void);

Funktion: Aus der Registrierdatenbank wird ermittelt, ob der Treiber MLXDRV installiert ist.

Parameter:

RETURN: = 0: Treiber ist nicht installiert
 !=0: Treiber ist installiert

3.2. Dateienverwaltung

mlx_inst_copy_files Treiberdateien kopieren

C: long mlx_inst_copy_files(void);

Funktion: DLLs und Treiberdateien werden ins Systemverzeichnis kopiert.

Parameter:

RETURN: !=0, falls Fehler beim Kopieren aufgetreten ist

mlx_inst_delete_files**Treiberdateien löschen**

C: long mlx_inst_delete_files(void);

Funktion: DLLs und Treiberdateien werden aus dem Systemverzeichnis gelöscht.

Parameter:

RETURN: !=0, falls Fehler beim Löschen aufgetreten ist

3.3. Einträge in der Registrierdatenbank vornehmen**mlx_inst_set_card_entries****Karteneinträge vornehmen**

C: long mlx_inst_set_card_entries(PREGCARDPARA pCard, ushort count);

Funktion: Die Variable *pCard* wird ausgewertet und die entsprechenden Eintragungen in die Registrierdatenbank vorgenommen.

Parameter:

pCard: Zeiger auf eine Struktur, die alle für die Registrierdatenbank notwendigen Kartenparameter enthält, wie Kartentyp, Basisadresse und Interruptkanal. *pCard* kann auf ein Array zeigen, wobei jedes Arrayelement eine Karte identifiziert.

count: Anzahl der Einträge, die gemacht werden sollen. Dieser Parameter entspricht der Länge des auszuwertenden Arrays PREGCARDPARA. Der Parameter muß zwischen 0 und 7 (inclusive) liegen.

RETURN: !=0, falls Fehler aufgetreten ist

mlx_inst_set_drv_entries**Allgemeine Treibereinträge vornehmen**

C: long mlx_inst_set_drv_entries(PREGDRVPARA pDrv);

Funktion: Die Variable *pDrv* wird ausgewertet und die entsprechenden Eintragungen werden in der Registrierdatenbank vorgenommen.

Parameter:

pDrv: Zeiger auf eine Struktur, die alle für den Treiber in der Registrierdatenbank notwendigen allgemeinen Parameter enthält.

RETURN: !=0, falls Fehler aufgetreten ist

mlx_inst_set_vdd_entry**Virtuellen Gerätetreiber eintragen**

C: long mlx_inst_set_vdd_entry();

Funktion: Der Virtuelle Gerätetreiber 'MLXVDD.DLL' wird in die Registrierdatenbank eingetragen.

Parameter:

RETURN: !=0, falls Fehler aufgetreten ist

3.4. Einträge aus der Registrierdatenbank löschen

mlx_inst_delete_reg_entries

Einträge löschen

C: long mlx_delete_reg_entries(void);

Funktion: Alle Einträge der Registrierdatenbank - sowohl allgemeine Treibereinträge als auch Karteneinträge - werden gelöscht. Ausgenommen ist der Virtuelle Gerätetreiber.

Parameter:

RETURN: !=0, falls Fehler aufgetreten ist

mlx_inst_delete_vdd_entry

Virtuellen Gerätetreiber löschen

C: long mlx_delete_vdd_entry(void);

Funktion: Der Virtuelle Gerätetreiber 'MLXVDD.DLL' wird aus der Registrierdatenbank gelöscht.

Parameter:

RETURN: !=0, falls Fehler aufgetreten ist

3.5. Einträge aus der Registrierdatenbank lesen

mlx_inst_get_card_entries

Karteneinträge lesen

C: long mlx_inst_get_card_entries(PREGCARDPARA pCard, ushort count);

Funktion: In die Variable *pCard* werden die aktuellen Eintragungen aus der Registrierdatenbank eingetragen.

Parameter:

pCard: Zeiger auf eine Struktur, die nach dem Funktionsaufruf die Registrierdatenbankeinträge wie Anzahl der Karten, Kartentyp, Basisadresse und Interruptkanal enthält. *pCard* kann auf ein Array zeigen, wobei in jedes Arrayelement eine Karte eingetragen wird. Eine Karteneintrag ist nur vorhanden, wenn das Element 'Active' der Struktur ungleich null ist.

count: Anzahl der Einträge, die gelesen werden sollen. Dieser Parameter entspricht der Länge des Arrays PREGCARDPARA. Der Parameter muß zwischen 0 und 7 (inclusive) liegen.

RETURN: !=0, falls Fehler aufgetreten ist

mlx_inst_get_drv_entries**Allgemeine Treibereinträge lesen**

C: long mlx_inst_get_drv_entries(PREGDRV PARA pDrv);

Funktion: Die Struktur *pDrv* wird aus der Registrierdatenbank gelesen.

Parameter:

pDrv: Zeiger auf eine Struktur, in die die Treibereinträge geschrieben werden.

RETURN: !=0, falls Fehler aufgetreten ist

4. Anwendung der Funktionen

4.1. Installierung des Treibers

Bei der Installierung des Treibers müssen folgende Funktionen ausgeführt werden:

1. mlx_inst_set_drv_entries(...)
2. mlx_inst_set_card_entries(...)
3. mlx_inst_set_vdd_entry()
4. mlx_inst_copy_files()

4.2. Deinstallierung des Treibers

Bei der Deinstallierung des Treibers müssen folgende Funktionen ausgeführt werden:

1. mlx_inst_delete_reg_entries()
2. mlx_inst_delete_vdd_entry()
3. mlx_inst_delete_files()