



Knihovna pro komunikaci s přenosnými paměťovými datovými terminály Cipherlab

Verze 1.00

Uživatelská příručka

Copyright © 2005 CODEWARE, s.r.o.

Obsah

1. Úvod	3
2. Obecné funkce	3
2.1 CODEWARE_CPTOPENCOMPORT	3
2.2 CODEWARE_CPTCLOSECOMPORT	3
2.3 CODEWARE_CPTWRITECOMPORT	3
2.4 CODEWARE_CPTWRITECOMPORTBIN	4
2.5 CODEWARE_CPTREADCOMPORT	4
3. Specializované funkce	5
3.1 PŘÍJEM DAT Z TERMINÁLU	5
3.1.2 CODEWARE_CPTStartDataUpload	5
3.1.3 CODEWARE_CPTReadDataRecord	5
3.1.4 CODEWARE_CPTFinishDataUpload	5
3.2 POSÍLÁNÍ DAT (SEZNAMU) DO TERMINÁLU	6
3.2.1 CODEWARE_CPTStartLookupDownload	6
3.2.2 CODEWARE_CPTSendLookupRecord	6
3.2.3 CODEWARE_CPTFinishLookupDownload	6
4. Pomocné funkce	7
4.1 CODEWARE_CPTGETCOMPORTHANDLE	7
4.2 CODEWARE_CPTSHOWERRORMESSAGE	7
4.3 CODEWARE_CPTBROWSEFORFILE	7

1. Úvod

Tato knihovna je určena pro komunikaci s přenosnými datovými terminály. Obsahuje jak obecné funkce, tak funkce specializované. Specializované funkce umožňují bez změny firmware terminálu jak posílání seznamu do terminálu, tak příjem dat z terminálu. Každý přenášený záznam musí být ukončen znakem CR ("\r", ASCII 13). Ke knihovně je přiložen příklad v EXCELU, kde jsou použity specializované funkce pro příjem a pro posílání dat.

2. Obecné funkce

Obecné funkce slouží pro čtení a zápis dat. Při zápisu je nutno na konec záznamu přidat znak CR ("\r", ASCII 13), při čtení tento znak obdobně znamená konec záznamu. Komunikace musí začínat funkcí `CODEWARE_CPTOpenComPort` a musí být ukončena funkcí `CODEWARE_CPTCloseComPort`.

2.1 *CODEWARE_CPTOpenComPort*

Funkce otvírá komunikační port pro čtení a zápis.

Deklarace: `DllExport int WINAPI CODEWARE_CPTOpenComPort(int iPortNumber,
int iBaudRate,
int iPortType);`

Parametry: `iPortNumber`: Číslo COM portu, může být 1..n
`iBaudRate`: Přenosová rychlost, 9600/19200/38400/57600/115200
`iPortType`: Typ připojení, `CPT_RS232`, `CPT_IRDA`, `CPT_CRADLE_IR`

Vrací: `CPT_OK`: Port byl v pořádku otevřen.
`CPT_COM_PORT_NOT_OPEN`: Port nemohl být otevřen.
`CPT_CRADLE_IR_NOT_SET`: CradleIR nemohl být nastaven.
`CPT_COM_PORT_NOT_INI`: Port nemohl být inicializován.

Příklad: `CODEWARE_CPTOpenComPort(1, 115200, CPT_CRADLE_IR);`

2.2 *CODEWARE_CPTCloseComPort*

Funkce zavírá předešle otevřený port.

Deklarace: `DllExport void WINAPI CODEWARE_CPTCloseComPort(void);`

Parametry: Žádné

Vrací: Nic

Příklad: `CODEWARE_CPTCloseComPort();`

2.3 *CODEWARE_CPTWriteComPort*

Pošle data na předešle otevřený port.

Deklarace: `DllExport void WINAPI CODEWARE_CPTWriteComPort(LPSTR szWriteString);`

Parametry: `szWriteString`: Posílaná data ukončená znakem '\0' (binární 0).

Vrací: Nic

Příklad: CODEWARE_CPTWriteComPort("READ\r");

2.4 CODEWARE_CPTWriteComPortBin

Pošle binární data na předešle otevřený port.

Deklarace: CODEWARE_CPTWriteComPortBin(BYTE * btWriteData,
DWORD dwDataLen);

Parametry: btWriteString: Posílaná data.
dwDataLen: Délka dat.

Vrací: Nic

Příklad: CODEWARE_CPTWriteComPortBin((BYTE *)"READ\r", 5);

2.5 CODEWARE_CPTReadComPort

Přijme datový řetězec (ukončený CR, nebo po prodlevě) z předešle otevřeného portu (terminálu).

Deklarace: DllExport int WINAPI CODEWARE_CPTReadComPort(LPSTR szReadString);

Parametry: szReadString: Přijatý datový řetězec.

Vrací: >= 0: Délka přijatého řetězce
CPT_TIMEOUT: Prodleva, nebo chyba při přenosu

Příklad: nChar = CODEWARE_CPTReadComPort(szBuf)

3. Specializované funkce

Specializované funkce umožňují komunikaci s terminálem, ve kterém jsou používány standardní komunikační funkce.

3.1 Příjem dat z terminálu

Používají se tyto funkce:

CODEWARE_CPTStartDataUpload-> CODEWARE_CPTReadDataRecord(opakovaně)->
CODEWARE_CPTFinishDataUpload

3.1.2 CODEWARE_CPTStartDataUpload

Zahajuje příjem dat z terminálu.

Deklarace: `DllExport int WINAPI CODEWARE_CPTStartDataUpload(int iPortNumber,
int iBaudRate,
int iPortType);`

Parametry: iPortNumber: Číslo COM portu, může být 1..n
iBaudRate: Přenosová rychlost, 9600/19200/38400/57600/115200
iPortType: Typ připojení, CPT_RS232, CPT_IRDA, CPT_CRADLE_IR

Vrací: CPT_OK: Port byl v pořádku otevřen.
CPT_COM_PORT_NOT_OPEN: Port nemohl být otevřen.
CPT_CRADLE_IR_NOT_SET: CradleIR nemohl být nastaven.
CPT_COM_PORT_NOT_INIT: Port nemohl být inicializován.
CPT_TIMEOUT: Prodleva, nebo chyba při přenosu.

Příklad: `CODEWARE_CPTStartDataUpload (1, 115200, CPT_CRADLE_IR);`

3.1.3 CODEWARE_CPTReadDataRecord

Přečte jeden záznam z terminálu.

Deklarace: `DllExport int WINAPI CODEWARE_CPTReadDataRecord (LPSTR szReadRecord);`

Parametry: szReadRecord: Přijatý datový řetězec.

Vrací: `>= 0`: Délka přijatého řetězce
CPT_TIMEOUT: Prodleva, nebo chyba při přenosu

Příklad: `nChar = CODEWARE_CPTReadDataRecord (szRecord)`

3.1.4 CODEWARE_CPTFinishDataUpload

Ukončuje příjem dat z terminálu a uzavře komunikační port.

Deklarace: `DllExport void WINAPI CODEWARE_CPTFinishDataUpload(void);`

Parametry: Žádné

Vrací: Nic

Příklad: CODEWARE_CPTFinishDataUpload ();

3.2 Posílání dat (seznamu) do terminálu

Používají se tyto funkce:

CODEWARE_CPTStartLookupDownload-> CODEWARE_CPTSendLookupRecord (opak.)->
CODEWARE_CPTFinishLookupDownload

3.2.1 CODEWARE_CPTStartLookupDownload

Zahajuje posílání dat (seznamu) do terminálu.

Deklarace: DllExport int WINAPI CODEWARE_CPTStartLookupDownload(int iPortNumber,
int iBaudRate,
int iPortType);

Parametry: iPortNumber: Číslo COM portu, může být 1..n
iBaudRate: Přenosová rychlost, 9600/19200/38400/57600/115200
iPortType: Typ připojení, CPT_RS232, CPT_IRDA, CPT_CRADLE_IR

Vrací: CPT_OK: Port byl v pořádku otevřen.
CPT_COM_PORT_NOT_OPEN: Port nemohl být otevřen.
CPT_CRADLE_IR_NOT_SET: CradleIR nemohl být nastaven.
CPT_COM_PORT_NOT_INI: Port nemohl být inicializován.
CPT_TIMEOUT: Prodleva, nebo chyba při přenosu.

Příklad: CODEWARE_CPTStartLookupDownload(1, 115200, CPT_CRADLE_IR);

3.2.2 CODEWARE_CPTSendLookupRecord

Pošle jeden záznam do terminálu.

Deklarace: DllExport int WINAPI CODEWARE_CPTSendLookupRecord(LPSTR szSendRecord);

Parametry: szSendRecord: Posílaný záznam ukončený binární nulou.

Vrací: CPT_OK: Port byl v pořádku otevřen.
CPT_TIMEOUT: Prodleva, nebo chyba při přenosu.
CPT_INVALID_RECORD: Záznam byl chybně přenesen, je možno ho poslat znovu.

Příklad: iStat = CODEWARE_CPTSendLookupRecord(szRecord)

3.2.3 CODEWARE_CPTFinishLookupDownload

Ukončuje posílání dat (seznamu) do terminálu a uzavře komunikační port.

Deklarace: DllExport void WINAPI CODEWARE_CPTFinishLookupDownload (void);

Parametry: Žádné

Vrací: Nic

Příklad: CODEWARE_CPTFinishLookupDownload();

4. Pomocné funkce

4.1 CODEWARE_CPTGetComPortHandle

Funkce vrátí handle (pro C) komunikačního portu.

Deklarace: `DllExport HANDLE WINAPI CODEWARE_CPTGetComPortHandle(void);`

Parametry: Žádné

Vrací: Handle otevřeného komunikačního portu (pro C).

Příklad: `hComPort = CODEWARE_CPTGetComPortHandle();`

4.2 CODEWARE_CPTShowErrorMessage

Povoluje, nebo zakazuje výpis chybových hlášení knihovny na obrazovku.

Deklarace: `DllExport int WINAPI CODEWARE_CPTShowErrorMessage(int nShow);`

Parametry: nShow: 1 = povolit, 0 = zakázat

Vrací: Vždy CPT_OK

Příklad: `CODEWARE_CPTShowErrorMessage(0);` // zakáže výpisy

4.3 CODEWARE_CPTBrowseForFile

Funkce umožní vyhledat požadovaný soubor.

Deklarace: `DllExport int WINAPI CODEWARE_CPTBrowseForFile(LPSTR szResultedFile,
LPSTR szStartDirectory,
LPSTR szFileTypeComment,
LPSTR szFileType);`

Parametry: szResultedFile: Vybraný soubor.
szStartDirectory: Adresář, kde má vyhledávání začít.
szFileTypeComment: Popis vyhledávaných souborů.
szFileType: Typ hledaných souborů.

Vrací: 0: Chybné ukončení (storno...)
>0 Délka specifikace (szresultedFile) nalezeného souboru.

Příklad: `CODEWARE_CPTBrowseForFile(szResultedFile,
"c:\\lookups",
"Text"
"*.txt");`



Communication Library for Batch Portable Terminals CipherLab

Version 1.00

User's Guide

Copyright © 2005 CODEWARE, s.r.o.

Contens

1. Preface	3
2. General Functions	3
2.1 CODEWARE_CPTOPENCOMPORT	3
2.2 CODEWARE_CPTCLOSECOMPORT	3
2.3 CODEWARE_CPTWRITECOMPORT	4
2.4 CODEWARE_CPTWRITECOMPORTBIN	4
2.5 CODEWARE_CPTREADCOMPORT	4
3. Specialized Functions	5
3.1 DATA RECEPTION FROM TERMINAL	5
3.1.2 CODEWARE_CPTStartDataUpload	5
3.1.3 CODEWARE_CPTReadDataRecord	5
3.1.4 CODEWARE_CPTFinishDataUpload	6
3.2 SENDING DATA (LOOKUP FILE) INTO TERMINAL	6
3.2.1 CODEWARE_CPTStartLookupDownload	6
3.2.2 CODEWARE_CPTSendLookupRecord	6
3.2.3 CODEWARE_CPTFinishLookupDownload	7
4. Auxilliary Functions	7
4.1 CODEWARE_CPTGETCOMPORTHANDLE	7
4.2 CODEWARE_CPTSHOWERRORMESSAGE	7
4.3 CODEWARE_CPTBROWSEFORFILE	8

1. Preface

The library is dedicated for communication with portable data terminals. It contains general functions as well as the specialized functions. The specialized functions allow you both data receive from terminal as well as lookup file transfer into terminal. Each record of transferred data must be terminated with CR character ("\r", ASCII 13). You can find MS Excel example in the package and learn there how to use functions for data sending and for data receiving.

2. General Functions

General functions usage is for reading and writing data. You must terminate each record with Carriage Return character CR ("\r", ASCII 13) during writing. Analogically CR during reading means end of record. Your communication must be initialized with CODEWARE_CPTOpenComPort function and must be closed with CODEWARE_CPTCloseComPort function.

2.1 CODEWARE_CPTOpenComPort

Opens communication port for read and write.

Declaration: `DllExport int WINAPI CODEWARE_CPTOpenComPort(int iPortNumber,
int iBaudRate,
int iPortType);`

Parameters: iPortNumber: COM port number, can be 1..n
iBaudRate: Baud rate, 9600/19200/38400/57600/115200
iPortType: Connection type, CPT_RS232, CPT_IRDA, CPT_CRADLE_IR

Returns: CPT_OK: No error.
CPT_COM_PORT_NOT_OPEN: Port cannot be opened.
CPT_CRADLE_IR_NOT_SET: CradleIR cannot be set up.
CPT_COM_PORT_NOT_INI: Unable to initialize the port.

Example: `CODEWARE_CPTOpenComPort(1, 115200, CPT_CRADLE_IR);`

2.2 CODEWARE_CPTCloseComPort

Closes the opened port.

Declaration: `DllExport void WINAPI CODEWARE_CPTCloseComPort(void);`

Parameters: None

Returns: Nothing

Example: `CODEWARE_CPTCloseComPort();`

2.3 CODEWARE_CPTWriteComPort

Sends data into opened port.

Declaration: DllExport void WINAPI CODEWARE_CPTWriteComPort(LPSTR szWriteString);

Parameters: szWriteString: Transferred data terminated with NULL character '\0' (binary 0).

Returns: Nothing

Example: CODEWARE_CPTWriteComPort("READ\r");

2.4 CODEWARE_CPTWriteComPortBin

Sends binary data into opened port.

Declaration: CODEWARE_CPTWriteComPortBin(BYTE * btWriteData,
DWORD dwDataLen);

Parameters: btWriteString: Transferred data.
dwDataLen: Length of data.

Returns: Nothing

Example: CODEWARE_CPTWriteComPortBin((BYTE *)"READ\r", 5);

2.5 CODEWARE_CPTReadComPort

Receives data string (terminated with CR or after timeout) from the opened port (i.e. from terminal).

Declaration: DllExport int WINAPI CODEWARE_CPTReadComPort(LPSTR szReadString);

Parameters: szReadString: Received data string.

Returns: >= 0: Length of received string
CPT_TIMEOUT: Timeout or transfer error

Example: nChar = CODEWARE_CPTReadComPort(szBuf)

3. Specialized Functions

Specialized functions are dedicated for communication with standard CipherLab terminal, i.e. for communication with terminal which uses standard CipherLab transfer protocol.

3.1 Data reception from terminal

Use below showed procedure for successful data receive:

```
CODEWARE_CPTStartDataUpload-> CODEWARE_CPTReadDataRecord(repeatedly)->  
CODEWARE_CPTFinishDataUpload
```

3.1.2 CODEWARE_CPTStartDataUpload

Starts data receive.

Declaration: `DllExport int WINAPI CODEWARE_CPTStartDataUpload(int iPortNumber,
int iBaudRate,
int iPortType);`

Parameters: `iPortNumber`: COM port number, can be 1..n
`iBaudRate`: Baud rate, 9600/19200/38400/57600/115200
`iPortType`: Connection type, `CPT_RS232`, `CPT_IRDA`, `CPT_CRADLE_IR`

Returns: `CPT_OK`: Port has been successfully opened.
`CPT_COM_PORT_NOT_OPEN`: Error during opening the port.
`CPT_CRADLE_IR_NOT_SET`: Cannot setup CradleIR.
`CPT_COM_PORT_NOT_INI`: Cannot initialize the port.
`CPT_TIMEOUT`: Timeout or transfer error.

Example: `CODEWARE_CPTStartDataUpload (1, 115200, CPT_CRADLE_IR);`

3.1.3 CODEWARE_CPTReadDataRecord

Reads one record from terminal.

Declaration: `DllExport int WINAPI CODEWARE_CPTReadDataRecord (LPSTR szReadRecord);`

Parameters: `szReadRecord`: Received data string.

Returns: `>= 0`: Length of received string
`CPT_TIMEOUT`: Timeout or transfer error

Example: `nChar = CODEWARE_CPTReadDataRecord (szRecord)`

3.1.4 CODEWARE_CPTFinishDataUpload

Finishes data reception and closes the communication port.

Declaration: DllExport void WINAPI CODEWARE_CPTFinishDataUpload(void);

Parameters: None

Returns: Nothing

Example: CODEWARE_CPTFinishDataUpload ();

3.2 Sending data (lookup file) into terminal

Use below showed procedure for successful sending data:

CODEWARE_CPTStartLookupDownload-> CODEWARE_CPTSendLookupRecord (repeat)->
CODEWARE_CPTFinishLookupDownload

3.2.1 CODEWARE_CPTStartLookupDownload

Initiation of data transfer into terminal.

Declaration: DllExport int WINAPI CODEWARE_CPTStartLookupDownload(int iPortNumber,
int iBaudRate,
int iPortType);

Parameters: iPortNumber: COM port number, can be 1..n

iBaudRate: Baud rate, 9600/19200/38400/57600/115200

iPortType: Connection type, CPT_RS232, CPT_IRDA, CPT_CRADLE_IR

Returns: CPT_OK: Port has been successfully opened.
CPT_COM_PORT_NOT_OPEN: Error during opening the port.
CPT_CRADLE_IR_NOT_SET: Cannot setup CradleIR.
CPT_COM_PORT_NOT_INIT: Cannot initialize the port.
CPT_TIMEOUT: Timeout or transfer error.

Example: CODEWARE_CPTStartLookupDownload(1, 115200, CPT_CRADLE_IR);

3.2.2 CODEWARE_CPTSendLookupRecord

Sends one record into terminal.

Declaration: DllExport int WINAPI CODEWARE_CPTSendLookupRecord(LPSTR szSendRecord);

Parameters: szSendRecord: Data string terminated with NULL character '\0'.

Returns: CPT_OK: Port has been successfully opened.
CPT_TIMEOUT: Timeout or transfer error.

CPT_INVALID_RECORD: Recoverable error during record transmission. You may send the record once more.

Example: iStat = CODEWARE_CPTSendLookupRecord(szRecord)

3.2.3 CODEWARE_CPTFinishLookupDownload

Finishes data transmission and closes the communication port.

Declaration: DllExport void WINAPI CODEWARE_CPTFinishLookupDownload (void);

Parameters: None

Returns: Nothing

Example: CODEWARE_CPTFinishLookupDownload();

4. Auxilliary Functions

4.1 CODEWARE_CPTGetComPortHandle

Returns communication port handle (in C language).

Declaration: DllExport HANDLE WINAPI CODEWARE_CPTGetComPortHandle(void);

Parameters: None

Returns: communication port handle (in C language).

Example: hComPort = CODEWARE_CPTGetComPortHandle();

4.2 CODEWARE_CPTShowErrorMessage

Enables or disables on-screen error messages.

Declaration: DllExport int WINAPI CODEWARE_CPTShowErrorMessage(int nShow);

Parameters: nShow: 1 = enable, 0 = disable

Returns: Always CPT_OK

Example: CODEWARE_CPTShowErrorMessage(0); // disables error messages

4.3 CODEWARE_CPTBrowseForFile

Launches dialog for opening a file.

Declaration: `DllExport int WINAPI CODEWARE_CPTBrowseForFile(LPSTR szResultedFile,
LPSTR szStartDirectory,
LPSTR szFileTypeComment,
LPSTR szFileType);`

Parameters: `szResultedFile:` Selected file.
`szStartDirectory:` Initial directory.
`szFileTypeComment` File descripton.
`szFileType:` File extension.

Returns: `0:` Operation has been cancelled
`>0` Length of returned string (`szresultedFile`).

Example: `CODEWARE_CPTBrowseForFile(szResultedFile,
"c:\\lookups",
"Text files"
".txt");`