

## **Programmer's manual for SPR\_LAN.DLL library**

### **Purpose of library**

Library SPR\_LAN.DLL contains communication functions for following Price Checkers:

- 1) LLT, LWT, LFT (LAN communication)
- 2) WLT, WWT, WFT (WiFi communication)

Library is 32-bit but can be also used in 64-bit Windows operating systems with 32-bit applications.

Before first use each price checker should be once configured according to "Programmer's manual for LLT,LWT,LFT,WLT,WWT,WFT Price Checkers.pdf" available at [ftp.elzab.com.pl/kody/i\\_tft.zip](ftp.elzab.com.pl/kody/i_tft.zip).

Basic configuration contains:

- setting IP address of Price Checker (static or dynamic by DHCP),
- setting IP address of Price Checkers server

### **Used data types**

#### **1) For texts:**

- PChar -  
memory for variable is not allocated by library and should be allocated by application  
when variable is passed from application to function:  
PChar – pointer to string with 00H code at the end  
when variable is passed from function to application:  
PChar – pointer to text buffer with at least 255-byte length; this buffer is filled by function with ASCII standard text

#### **2) Numbers:**

- Longword – variable with 4-byte length (range 0..4294967295)

#### **3) procedures**

Warning: procedures do not allocate and do not release memory for text data types – application should do it itself.

- TBarcodeEvent = procedure (IPAddress : PChar; Barcode : PChar;  
TextToDisplay : PChar; var BeepType : Longword; var DisplayTime : Longword; var  
TextToPrinter : PChar); StdCall;

Function of this type catches questions from price checker. Application should give address of this type function to library as parameter of PriceCheckerServiceStarted function.

Function TBarcodeEvent type is called when price checker reads barcode.

Function TBarcodeEvent type has parameters:

IPAddress – IP address of price checker which sent question

Barcode – barcode read on price checker

Function TBarcodeEvent type should set parameters:

TextToDisplay – 40 characters to display of price checker (20 characters at front means data for first display line, 20 next characters means data for second display line). Lines shorter than 20 characters should end with LF character (0AH)

BeepType – type of beep when answer is displayed. Possible beep types:

0 – no beep

1 – OK beep

2 – error beep

DisplayTime – time in seconds how long price checker display computer's answer

TextToPrinter – data to print in case of Price Checker with printer

- TStatusEvent = procedure (IPAddress : PChar; DisplayType : longword; PrinterType: longword; Status : Longword; Version : PChar); StdCall;

Function of this type catches answers from price checker when application sent question about status.

Application should give address of this type function to library as parameter of PriceCheckerServiceStarted function.

Function TStatusEvent type has parameters:

IPAddress – IP address of price checker which sent answer about status

DisplayType – always '3' which means graphical TFT 480 x 272 pixels

PrinterType – 0 (Price Checker without printer) or 1 (Price Checker with printer)

Status – status byte is stored in the lowest byte of Status variable

Bits:

1	b <sub>6</sub>	b <sub>5</sub>	b <sub>4</sub>	b <sub>3</sub>	b <sub>2</sub>	b <sub>1</sub>	b <sub>0</sub>
---	----------------	----------------	----------------	----------------	----------------	----------------	----------------

b<sub>0</sub> = 0 – Price Checker with LAN interface; 1 – Price Checker with WiFi interface

b<sub>1</sub> = reserved

b<sub>2</sub> = reserved

b<sub>3</sub> = 1 – display failure

b<sub>4</sub> = reserved

b<sub>5</sub> = reserved

b<sub>6</sub> = reserved

b<sub>7</sub> = always 1

Version – version of price checker firmware (text with CR character at end, max. 20 characters)

- TErrorEvent = procedure (IPAddress : PChar; TransmissionType : Longword; ErrorNo : Longword; ErrorDescription : PChar); StdCall;

Function of this type catches errors.

Application should give address of this type function to library as parameter of PriceCheckerServiceStarted function.

Function TErrorEvent type has parameters:

IPAddress – IP address of price checker which sent information about error

TransmissionType – transmission type which caused an error:

1 – error cause is ReadStatus function

ErrorNo – error number:

4 – improper or undefined IP address

5 – transmission timeout

ErrorDescription – text description of error

## Functions exported from library

- function **PriceCheckerServiceStarted**(var ErrorNo : Longword;  
ErrorDescription : PChar; BarcodeService : TBarcodeEvent; StatusService :  
TStatusEvent; ErrorService : TErrorEvent) : boolean; StdCall;

Function should be called one time at start price checker service.

Function returns true on success.

Parameters:

ErrorNo – error number; if function returns true, then always ErrorNo=0;

Possible errors:

0 – operation finished successful

1 – barcode service function not defined

2 – status service function not defined

3 – error service function not defined

6 – price checker service already started

ErrorDescription – text error description of ErrorNo error

BarcodeService - pointer to barcode service function (see TBarcodeEvent type)

StatusService - pointer to status service function (see TStatusEvent type)

ErrorService – pointer to error service function (see TErrorEvent type)

- procedure **PriceCheckerServiceStop**; StdCall;

Stop of price checker service. This function should be called before end of application.

- procedure **StatusRead**(IPAddress : PChar); StdCall;

Send question about status to price checker of given IPAddress.

Value of status byte is return by function TStatusEvent type which address was defined in function PriceCheckerServiceStarted.

## Library versions

The newest version is always available on ELZAB's ftp server

[ftp://ftp.elzab.com.pl/kody/spr\\_lan.zip](ftp://ftp.elzab.com.pl/kody/spr_lan.zip)