

1. LLT, LWT, LFT AND WLT, WWT, WFT PRICE CHECKERS	2
1.1. Configuration via www site	2
1.2. Configuration using pendrive	3
1.2.1 Setting advertisements	4
1.2.2 Programming of additional bitmaps related to skins	5
1.2.3 Programming graphics and sounds related to product	6
1.2.4 Changing configuration password	6
1.2.5 Main configuration	6
2. CONFIGURATION OF WIRELESS NETWORK FOR WLT, WWT, WFT PRICE CHECKERS	7
3. LLT, LWT, LFT, WLT, WWT, WFT PRICE CHECKER'S TRANSMISSION PROTOCOL	9
4. FIRMWARE UPGRADE	14
5. RESTORING PASSWORD	15

v.2.00 24/08/2015

Programmer's manual

for LLT/LWT/LFG/WLT/WWT/WFT Price Checkers

Attention: To connect to WLT/WWT/WFT Price Checkers you must first configure wireless connection between Price Checker and Access Point. How to configure such connection is described in point 2.

1. LLT, LWT, LFT and WLT, WWT, WFT Price Checkers

1.1. Configuration via www site

Default IP address of Price Checker is: 10.0.0.1

If IP address is not known, it can be read using 000000000000 barcode (such barcode is available in User's manual).

After you write IP address of Price Checker in internet browser you will see login and password prompt.

Login: admin

Default password: ELZAB

Configuration parameters:

Name of parameter	Meaning of parameter	Range	Default setting
DHCP	If switched on it allows to set network parameters to Price Checker by DHCP server	On/off	off
IP	IP address of Price Checker (if DHCP client is switched off)	Static IP addresses; in WLT/WWT/WFT Price Checkers IP address 192.168.1.254 is reserved	10.0.0.1
Subnet	Subnet mask (if DHCP client is switched off)		255.0.0.0
Gateway	Gateway (if DHCP client is switched off)	Static IP addresses; in WLT/WWT/WFT Price Checkers IP address 192.168.1.254 is reserved	10.0.0.0
Port UDP	UDP port to receive data by Price Checker; UDP port to send data by Price Checker	1000-65534	1001
IP of price checker server	IP address of computer on which Price Checker's server is running	Static IP addresses	10.0.0.2

Password	Password to www site and to configuration via USB	Max 8 characters	ELZAB
Backlight	Backlight of TFT display	0-9	8
Volume	Volume of sound	0-5	3
Skin	Set of graphics displayed during operation	ELZAB1 (default) ELZAB2 USER TEXT	ELZAB1
Sound voice	Type of voice for spoken price. Price is spoken in case of using 'A' control sequence or detecting price by Price detect mode	No, male voice, female voice	Female voice
Signals	Sounds for important information (like scanning barcode)	On/off	On
Price detect mode	Mode of detecting price in '1' and '6' control sequences. It must be tested with specific Price Checker's server. This mode needs "Huf" or "Ft" string after price to work properly in case of Hungarian firmware	On/off	On
Checker mode	Type of asking (emulation of previous Price Checkers)	LL/LW/LF LFG LFT/LWT/LFT PRINTER WITH CUTTER PRINTER WITHOUT CUTTER	LL/LW/LF
Display time of advertising	Time of displaying files ad00.bmp to ad99.bmp (if they exists)	0 to 99 seconds	

1.2. Configuration using pendrive

After connecting pendrive Price Checker switches off internal and external barcode scanner.

There are switched on back after removing pendrive.

First operation is verification of pass.txt file. In this file in first line you should put password for configuration.

It is common password for www site and pendrive.

If pass.txt exists and contains proper password Price Checker reads configuration. All directories from pendrive replace data in Price Checker.

If there is no pass.txt file or it contains improper password Price Checker does not read configuration.

1.2.1 Setting advertisements

You should create ADS directory. Content of directory:

a) .bmp files

Format of .bmp files:

- Number of colours: 256
- compression: no compression
- max resolution: 480x272 pixels (bitmaps which have larger dimensions are not displayed)

File names:

ad?? .bmp, where ?? is in range from 00 to 99

b) .wav files

Format of .wav files:

- number of channels: 1 (mono),
- compression: no compression (PCM)
- sampling: up to 44.1 kHz
- resolution: 8 bits

File names:

ad?? .wav, where ?? if in range from 00 do 99

c) ads.xml file with configuration

File format:

```
<xml>
  <config>
    <ad00>
      <time>time</time>
    </ad00>
    <ad01>
      <time> time </time>
    </ad01>
    <ad02>
      <time> time </time>
    </ad02>
```

```

...
    <ad99>
        <time> time </time>
    </ad99>
</config>
</xml>

```

where time – time of displaying advertisement is second (range from 0 to 99 seconds). Value 0 means: continuously till next barcode is scanned.

Fields <ad00> to <ad99> refer to file names ad00.bmp to ad99.bmp and ad00.wav to ad99.wav.

Example:

ADS directory contains files : ad00.bmp, ad00.wav, ad02.bmp, ad05.bmp, ads.xml

ads.xml file has following parts:

```

<ad00>
    <time>5</time>
</ad00>

<ad02>
    <time>10</time>
</ad02>

<ad05>
    <time>2</time>
</ad05>

<ad15>
    <time>2</time>
</ad15>

```

Price Checker displays periodically files and:

- *ad00.bmp file will be displayed 5 seconds; during displaying time there will be ad00.wav played; if the time of playing ad00.wav will be longer than time of displaying declared in ads.xml, ad00.bmp will be displayed longer,*
- *ad02.bmp file will be displayed 10 seconds,*
- *ad.05.bmp file will be displayed 2 seconds,*
- *ad15.bmp will not be displayed (it does not exist in ADS directory) .*

1.2.2 Programming of additional bitmaps related to skins

You should create directory:

ADDBMP which contains add1.bmp, add2.bmp and add3.bmp files; these files contains additional bitmaps related to the given skin

ADDBMP directory should be put to subdirectory:

SKINS\ELZAB1 – for skin ELZAB1

SKINS\ELZAB2 – for skin ELZAB2

SKINS\UZYT – for skin USER

1.2.3 Programming graphics and sounds related to product

Price Checker allows to connect displaying graphics/playing sound to product information. Name of file with graphics/sound you should send in 'A' sequence (see LLT/LFT/LWT mode). Files formats are the same as formats of advertisement files described in point 1.2.1

Files should be put to directory ADS_WAV .

1.2.4 Changing configuration password

You should create newPass.txt file.

In first line of file new password should be placed.

To change password it must be pass.txt file with current password.

Password is common for pendrive and www site configuration.

1.2.5 Main configuration

File name: conf.xml, it should be placed in CONFIG directory

```
<xml>
  <config>
    <main>
      <lang>mode</lang>
      <backlight>backlight</backlight>
      <volume>volume</volume>
      <currentTemplates>skin</currentTemplates>
      <price_voice>voice</price_voice>
      <signals>signal</signals>
      <priceParse>parse</priceParse>
      <mode>mode</mode>
    </main>
  <lan>
    <ip>ip </ip>
```

```

<subnet>subnet</subnet>
<gateway>gateway</gateway>
<dhcp>dhcp</dhcp>
<serv>iserver</serv>
<udp>udp</udp>
</lan>
</config>
</xml>

```

Meaning of fields:

Lang – refers to price voice and price checker in English version; meaning of field: 1- price is spoken in euro, 2 – price is spoken in dollars, 3 – price is spoken in pounds

backlight – Backlight of TFT display (0-9)

volume – Volume of sound (0-5)

currentTemplates – Set of graphics displayed during operation (0-3)

price_voice – Type of voice for spoken price (0 – no voice, 1 – male, 2 – female)

signals – Sounds for important information (like reading barcode) (0 – off, 1 – on)

priceParse - mode of detecting price (0 – off, 1 – on)

mode Type of asking (emulation of previous Price Checkers) (0 – as LL/LW/LF, 1 – as LFG, 2 – as LLT/LWT/LFT, 3 – printer with cutter, 4 – printer without cutter

dhcp – DHCP client (0 – off , 1 – on)

ip – IP address of Price Checker

subnet – subnet mask

gateway – gateway IP address

serv – IP address of Price Checker's server

udp – UDP port for transmission from and to Price Checker

2. Configuration of wireless network for WLT, WWT, WFT Price Checkers

To connect with wireless network you should read Price Checker's status using 00000000000000 barcode. There will be list of parameters on TFT display. For wireless connection there are important such parameters:

- SSID – you should set the same value in Price Checker and in Access Point (AP); default SSID in Price Checker is WLANAP,
- Encryption – you should set the same mode of encryption in Price Checker and in Access Point (AP); default encryption is disabled
- State – state of wireless transmission between Price Checker and Access Point (AP). State: „connected” means, that wireless connection is established.

To set SSID and encryption in conf.xml file you should place <wi-fi> section.

```

<xml>
<config>

```

```
<wi-fi>
  <ssid>ssid</ssid>
  <auth_mode>encryption</auth_mode>
  <key>password</key>
</wi-fi>
</config>
</xml>
```

Meaning of the fields:

ssid –SSID, max 21 characters; default SSID is WLANAP, SSID can be displayed after scanning 0000000000000000 barcode

encryption – mode of encryption

0 – no encryption

1 – WEP-OPEN (64-bit)

2 – WEP –SHARED (64-bit)

3 - WPA – PSK

4 - WPA2-PSK

password – password for encryption mode, space and underline characters are forbidden in this field

for encryption mode 1 and 2 it is 10 hexadecimal characters (e.g. „12F03A10CD”)

for encryption mode 3 i 4 – at least 8 ASCII characters,

3. LLT, LWT, LFT, WLT, WWT, WFT Price Checker's transmission protocol

General characteristics

Each price checker has got its own IP number.

This number is stored in a price checker's memory and can be configured through a LAN network. Each price checker requires unique IP number in the given LAN network.

To communicate with a price checker have to be known its IP number.

For communication with a price checker is used UDP protocol (default 1001 port number).

Reception of each data frame have to be confirmed by resent data frame or by acknowledgement command.

In case of not receiving confirmation within 1 second a data frame should be sent again (maximum 3 times).

Used symbols.

Symbol	Value	Description
STX	02H	Text block beginning
ETX	03H	Text block end
ENQ	05H	Emulation mode
ACK	06H	Confirmation of data frame reception
LF	0AH	Line feed
CR	0DH	Data end
ESC	1BH	Beginning of sequence for graphical display

Data frame.

STX	<FRAME_NO>	<DATA_FRAME>	ETX
1	1	Variable length (1..1024)	1

Acknowledgement frame.

STX	<FRAME_NO>	ACK	ETX
1	1	1	1

<FRAME_NO>

ASCII character from the range 80H.. FFH (128..255)

<DATA_FRAME>

ASCII characters from the range 20H.. FFH (32..255) plus LF, CR, ESC, ENQ symbols

< DATA_FRAME > = <COMMAND><DATA>

or

< DATA_FRAME > = <COMMAND>

<COMMAND> – command byte, ASCII character from the range 20H..FFH (32..255)

<DATA> – ASCII character from the range 20H..FFH (32..255) plus LF, CR, ESC symbols, variable quantity of characters depending on the command type.

PC – price checker sequence syntax

During barcode reading a price checker is a MASTER device and a PC is a SLAVE one.
 In other cases a price checker is a SLAVE device and a PC is a MASTER one.

Depending on ‘Checker mode’ parameter Price Checker sends:

- for LL/LW/LF setting Price Checker sends sequence ‘1’ (see point **Two lines mode**);
- for LFG setting Price Checker sends sequence ‘6’ (see point **Three lines mode**);
- for LLT/LWT/LFT setting Price Checker sends sequence 'A' (see point **LLT/LFT/LWT/WLT/WWT/WFT mode**);
- for “printer with cutter” or “printer without cutter” setting Price Checker sends sequence ‘2’ (see point **Printer mode**)

Independently from type of sequence all responses from server are acceptable.

Preferred type of server’s answer is ‘A’ sequence.

Two lines mode (emulation of LL/LW/LF Price Checkers)

Price checker	PC
STX FRAME_NO 1 BarCode ETX	
	STX FRAME_NO 1 SCode CHARACTERS_DISPLAY1 SIGNAL TIME ETX
STX FRAME_NO ACK ETX	

While executing this task: PRICE CHECKER = MASTER, PC = SLAVE

FRAME_NO is generated by the price checker and placed in the request frame.

The PC is placing FRAME_NO received from the price checker in the response frame what acknowledges at the same time reception of the request by the PC.

After receiving response from the PC the price checker acknowledges reception of the response by sending the frame with ACK and FRAME_NO to the PC.

BarCode – barcode ASCII characters finished with CR sign.

In version 1 of price checker’s firmware version are sent maximum 20 barcode characters and from version 2 are sent maximum 23 barcode characters.

SCode – CRC16 checksum of the barcode sent to the PC,

in the Appendix No. 1 is shown the source code of the function calculating the CRC16 sum in Pascal language, as the starting value for CRC16 calculation algorithm should be considered 0,

after calculating this sum the highest bit in both bytes should be set to 1,

lowest checksum byte is sent first,

SIGNAL – voice signal – one ASCII character:

‘0’ – no signal

‘1’ – acknowledgement signal

‘2’ – error signal.

TIME – time of text displaying in seconds, two ASCII characters, e.g. ‘03’ means 3 seconds,

if TIME = '00' then text is displayed till the next barcode is read.

CHARACTERS_DISPLAY1 – characters to display, two lines of text, 20 characters maximum each, totally maximum 40 characters, each line should be finished with LF sign if there is less than 20 characters to display.

Three lines mode (emulation of LFG Price Checker)

PRICE CHECKER	PC
STX FRAME_NO 6 BarCode ETX	
	STX FRAME_NO 6 SCode 0 CHARACTERS_DISPLAY2 SIGNAL TIME ETX
STX FRAME_NO ACK ETX	

While executing this task: PRICE CHECKER = MASTER, PC = SLAVE

FRAME_NO is generated by the price checker and placed in the request frame.

The PC is placing FRAME_NO received from the price checker in the response frame what acknowledges at the same time reception of the request by the PC.

After receiving response from the PC the price checker acknowledges reception of the response by sending the frame with ACK and FRAME_NR to the PC.

BACKGROUND_NO – the number of the background on which text is displayed = one of '0'...'9' or 'A'...'F' characters.

CHARACTERS_DISPLAY2 – characters to display, three lines of text, 32 characters maximum each, totally maximum 96 characters, each line should be finished with LF sign if there is less than 32 characters to display.

Because of character generator there are max 24 characters displayed in one line of text

LLT/LFT/LWT/WLT/WWT/WFT mode (displaying formatted by Price Checker)

PRICE CHECKER	PC
STX FRAME_NO A BarCode ETX	
	STX FRAME_NO A SCode STATUS NAME PRICE CURRENCY GRAPHICS GRAPHICS_TIME SOUND ATTR ADD_TEXT ETX
STX NrRamki ACK ETX	

While executing this task: PRICE CHECKER = MASTER, PC = SLAVE

FRAME_NO is generated by the price checker and placed in the request frame.

The PC is placing FRAME_NO received from the price checker in the response frame what acknowledges at the same time reception of the request by the PC.

After receiving response from the PC the price checker acknowledges reception of the response by sending the frame with ACK and FRAME_NR to the PC.

STATUS one character;

'1' – next data valid (product was found)

'0' – next data invalid (product not found)

NAME – name of product; up to 20 ASCII characters; code page 852; if field is shorter than 20 characters it must be ended by LF

PRICE – max 8 digits + optionally dot or comma; if field is shorter it must be ended by LF

CURRENCY – currency symbol; max 3 ASCII characters; if field is shorter it must be ended by LF

GRAPHICS – name of bmp file which should be displayed after information about name and price;

Max length of this field is 12 characters (up to 8 characters of file name, one character for dot, up to 3 characters of file extension); if field is shorter than 12 characters or not exists it must be ended by LF

GRAPHICS_TIME – 2 digits, time of displaying graphic defined in GRAPHICS field

SOUND – name of wav file which should be played after information about name and price;

Max length of this field is 12 characters (up to 8 characters of file name, one character for dot, up to 3 characters of file extension); if field is shorter than 12 characters or not exists it must be ended by LF

ATTR – 3 ASCII digits

1st digit – status of 1st attribute (0 – do not display addbmp1, 1 – display addbmp1)

2nd digit – status of 2nd attribute (0 – do not display addbmp2, 1 – display addbmp2)

3rd digit – status of 3rd attribute (0 – do not display addbmp3, 1 – display addbmp3)

ADD_TEXT – additional text (e.g. promotion); max 20 characters

Printer mode

For Price Checkers with printer

PRICE CHECKER	PC
STX FRAME_NO_1 2 BarCode ETX	
	STX FRAME_NO_1 1 SCode CHARACTERS_DISPLAY1 SIGNAL TIME ETX
STX FRAME_NO_1 ACK ETX	
	STX FRAME_NO_2 2 SCode CHARACTERS_PRINTER ETX
STX, FRAME_NO_2 ACK ETX	

While executing this task in its first part: PRICE CHECKER = MASTER, PC = SLAVE

FRAME_NO_1 is generated by the price checker and placed in the request frame.

The PC is placing FRAME_NO_1 received from the price checker in the response frame what acknowledges at the same time reception of the request by the PC.

After receiving response from the PC the price checker acknowledges reception of the response by sending the frame with ACK and FRAME_NO_1 to the PC.

In second part of this task PRICE_CHECKER = SLAVE, PC = MASTER

FRAME_NO_2 is generated by computer and placed in the sending frame.

Price Checker after receiving frame acknowledges reception of the response by sending the frame with ACK and FRAME_NO_2 to the PC

Attention:

In case there is no data to print do not send

STX FRAME_NO_2 SKod CHARACTERS_PRINTER ETX

(of course there is also no answer STX FRAME_NO_2 ACK ETX)

Price Checker status request

PC	PRICE CHECKER
STX FRAME_NO 3 ETX	
	STX FRAME_NO 3 DISP 0 STATE VERSION ETX
STX FRAME_NO ACK ETX	

While executing this task: PRICE CHECKER = SLAVE, PC = MASTER

FRAME_NO is generated by the PC and placed in the request frame.

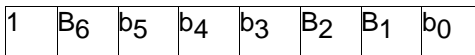
After receiving the request from the PC the price checker acknowledges reception of the request by sending the frame with ACK and FRAME_NO to the PC.

After receiving response from the price checker the PC acknowledges reception of the response by sending the frame with ACK and FRAME_NO to the price checker.

DISP – price checker’s display type (one ASCII character):

'3' – TFT 480x272 pixels

STATE – Price Checker’s state



b₀ = 0 – Price Checker with LAN interface 1 – Price Checker with WiFi interface

b₁ = 0

b₂ = 0

b₃ = 1 – display is out of order

b₄ = 0

b₅ = 0

b₆ = 0

b₇ = 1

VERSION – price checker’s firmware version, text finished with CR sign, maximum 20 characters.

Source code of the function calculating the CRC16 sum in Pascal language.

```
function TypReceivingChannel.SumCRC16(DataForSum : string) : string;
```

```
const
```

```
TableCRC16 : array [0..255] of word =
```

```
(
```

```

$0000, $C0C1, $C181, $0140, $C301, $03C0, $0280, $C241,
$C601, $06C0, $0780, $C741, $0500, $C5C1, $C481, $0440,
$CC01, $0CC0, $0D80, $CD41, $0F00, $CFC1, $CE81, $0E40,
$0A00, $CAC1, $CB81, $0B40, $C901, $09C0, $0880, $C841,
$D801, $18C0, $1980, $D941, $1B00, $DBC1, $DA81, $1A40,
$1E00, $DEC1, $DF81, $1F40, $DD01, $1DC0, $1C80, $DC41,
$1400, $D4C1, $D581, $1540, $D701, $17C0, $1680, $D641,
$D201, $12C0, $1380, $D341, $1100, $D1C1, $D081, $1040,
$F001, $30C0, $3180, $F141, $3300, $F3C1, $F281, $3240,
$3600, $F6C1, $F781, $3740, $F501, $35C0, $3480, $F441,
$3C00, $FCC1, $FD81, $3D40, $FF01, $3FC0, $3E80, $FE41,
$FA01, $3AC0, $3B80, $FB41, $3900, $F9C1, $F881, $3840,
$2800, $E8C1, $E981, $2940, $EB01, $2BC0, $2A80, $EA41,
$EE01, $2EC0, $2F80, $EF41, $2D00, $EDC1, $EC81, $2C40,
$E401, $24C0, $2580, $E541, $2700, $E7C1, $E681, $2640,
$2200, $E2C1, $E381, $2340, $E101, $21C0, $2080, $E041,
$A001, $60C0, $6180, $A141, $6300, $A3C1, $A281, $6240,
$6600, $A6C1, $A781, $6740, $A501, $65C0, $6480, $A441,
$6C00, $ACC1, $AD81, $6D40, $AF01, $6FC0, $6E80, $AE41,
$AA01, $6AC0, $6B80, $AB41, $6900, $A9C1, $A881, $6840,
$7800, $B8C1, $B981, $7940, $BB01, $7BC0, $7A80, $BA41,
$BE01, $7EC0, $7F80, $BF41, $7D00, $BDC1, $BC81, $7C40,
$B401, $74C0, $7580, $B541, $7700, $B7C1, $B681, $7640,
$7200, $B2C1, $B381, $7340, $B101, $71C0, $7080, $B041,
$5000, $90C1, $9181, $5140, $9301, $53C0, $5280, $9241,
$9601, $56C0, $5780, $9741, $5500, $95C1, $9481, $5440,
$9C01, $5CC0, $5D80, $9D41, $5F00, $9FC1, $9E81, $5E40,
$5A00, $9AC1, $9B81, $5B40, $9901, $59C0, $5880, $9841,
$8801, $48C0, $4980, $8941, $4B00, $8BC1, $8A81, $4A40,
$4E00, $8EC1, $8F81, $4F40, $8D01, $4DC0, $4C80, $8C41,
$4400, $84C1, $8581, $4540, $8701, $47C0, $4680, $8641,
$8201, $42C0, $4380, $8341, $4100, $81C1, $8081, $4040

```

);

variables

```

Counter      : byte;
TableIndicator : byte;
SumValue     : word;

```

begin

```
SumValue := 0;
```

```
for Counter := 1 to Length(DataForSum) to
```

```
begin
```

```
TableIndicator := (byte(SumValue)) xor byte(DataForSum[Counter]);
```

```
SumValue := (SumValue shr 8) xor TableCRC16[TableIndicator];
```

```
end;
```

```
SumValue := SumValue or $8080;
```

```
Result := chr(Lo(SumValue))+chr(Hi(SumValue));
```

end;

4. Firmware upgrade

SPT1 – LAN/WiFi Price Checker with **external** database and **USB** management

SPT4 – LAN/ WiFi Price Checker with **external** database and **FTP** management

Way of programming:

USB access -> file „pass.txt” with password must be put in root directory of pendrive

Default password: ELZAB

FTP access -> you need to know user name and password

user: admin

password: ELZAB

If you change firmware in SPT1 model you must have proper firmware file (spt04 or spt01), change its name to **firmware.bin** and make upgrade via USB

If you change firmware in SPT4 model you must have proper firmware file (spt04 or spt01), change its name to **spt4.bin** and make upgrade via FTP (e.g. using LxT program)

Upgrade via USB:

-put on pendrive root directory files „firmware.bin” and „pass.tx”;

-put pendrive in Price Checker

-remove pendrive after appropriate text on Price Checker’s display

-wait for firmware upgrade,

Upgrade via FTP

-log to Price Checker using any FTP client (it can be also LxT program),

-copy to „checker_ftp” directory file spt4.bin,

- close FTP session, after it upgrade will be done

5. Restoring password

During first minute after start you can restore default „ELZAB” password. To do it you must put special „RESETPASS” barcode under Price Checker’s scanner.
