

# PosnetLib.ocx

Instrukcja programisty.

## Spis treści

1. Konfiguracja połączenia z drukarką.....	3
ConnectCom.....	3
ConnectTcp.....	4
Disconnect.....	4
2. Wysyłanie i odbieranie danych.....	5
Send.....	5
Receive.....	5
SendFrame.....	6
ReceiveFrame.....	7
SendFile.....	8
WaitForResponse.....	9
3. Logowanie danych.....	10
LogBegin.....	10
LogEnd.....	10
4. Funkcje pomocnicze.....	11
PosnetLibGetVersion.....	11
CleanUp.....	11
CreateFrame.....	12
GetCRC.....	13
GetToken.....	14
GetParamValue.....	14
GetErrText.....	15
5. Informacje pomocnicze.....	16
Kody błędów wewnętrznych kontrolki.....	16
Wsparcie techniczne.....	16

# 1. Konfiguracja połączenia z drukarką.

## ConnectCom

Opis:

funkcja umożliwia nawiązanie połączenia z drukarką fiskalną podłączoną do komputera przez fizyczny lub wirtualny port szeregowy.

Wywołanie funkcji:

```
int ConnectCom( LPCTSTR portName, long baudRate, short flowControl );
```

Przekazywane parametry:

portName – identyfikator portu szeregowego COM,

baudRate – prędkość przesyłania danych (1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200),

flowControl – sterowanie przepływem, 0 – brak, 1 – Xon/Xoff, 2 – Rts/Cts.

Zwracana wartość:

„0” jeżeli wykonano poprawnie lub kod błędu ze znakiem „-”.

Przykład wywołania

```
ConnectCom( "COM1", 9600, 1 );
```

Uwagi:

Parametry komunikacji muszą być zgodne z ustawieniami w drukarce fiskalnej.

## ConnectTcp

Opis:

funkcja umożliwia nawiązanie połączenia z drukarką fiskalną podłączoną do komputera przez złącze Ethernet.

Wywołanie funkcji:

```
int ConnectTcp( LPCTSTR ipAddress, short netSocket );
```

Przekazywane parametry:

ipAddress – adres IP drukarki fiskalnej,

netSocket – numer portu ustawiony w drukarce dla połączenia z PC.

Zwracana wartość:

„0” jeżeli wykonano poprawnie lub kod błędu ze znakiem „-”.

Przykład wywołania

```
ConnectTcp( "10.10.10.10", 6666 );
```

Uwagi:

W przypadku wprowadzenia nieprawidłowych wartości próba nawiązania połączenia może trwać nawet kilka sekund, aplikacja może w tym czasie przestać odpowiadać.

## Disconnect

Opis:

funkcja umożliwia przerwanie połączenia z drukarką fiskalną, niezależnie od sposobu nawiązania połączenia.

Wywołanie funkcji:

```
int Disconnect();
```

Zwracana wartość:

„0” jeżeli wykonano poprawnie lub kod błędu ze znakiem „-”.

Przykład wywołania

```
Disconnect();
```

## 2. Wysyłanie i odbieranie danych.

### Send

Opis:

funkcja umożliwia przesłanie dowolnych danych do drukarki fiskalnej.

Wywołanie funkcji:

```
int Send( LPCTSTR command );
```

Przekazywane parametry:

command – ciąg znaków który chcemy przesłać do drukarki.

Zwracana wartość:

„0” jeżeli wykonano poprawnie lub kod błędu ze znakiem „-”.

Przykład wywołania

```
Send( STX + "sdev\t@1\t#8FE8" + ETX );
```

Gdzie:

STX – początek ramki protokołu Posnet '0x02'h,

ETX – koniec ramki protokołu Posnet '0x03'h.

### Receive

Opis:

funkcja umożliwia odczytanie danych odesłanych przez drukarkę fiskalną.

Wywołanie funkcji:

```
string Receive();
```

Zwracana wartość:

ciąg znaków odebrany od drukarki fiskalnej lub kod błędu.

Przykład wywołania

```
Receive();
```

Uwagi:

Funkcja nie kontroluje treści odebranych danych.

## SendFrame

Opis:

funkcja umożliwia przesłanie do drukarki fiskalnej ramki protokołu Posnet zbudowanej na podstawie przekazanych parametrów.

Wywołanie funkcji:

```
int SendFrame( LPCTSTR requestId, LPCTSTR requestParams );
```

Przekazywane parametry:

requestId – identyfikator polecenia,

requestParams – parametry polecenia rozdzielone znakiem TAB '09'h.

Zwracana wartość:

„0” jeżeli wykonano poprawnie lub kod błędu ze znakiem „-”.

Przykład wywołania

```
SendFrame( "dailyrep", "da2013-08-19");
```

Uwagi:

Przekazywane parametry w „requestId” oraz „requestParams” powinny być zgodne z Instrukcją Programisty Protokołu Posnet dla konkretnego typu drukarki.

Pierwszy parametr w wywołaniu funkcji musi być niezerowy( != "" ).

Jeżeli w requestParams chcemy przekazać więcej niż jeden parametr dla polecenia protokołu należy je rozdzielić znakiem TAB '09'h np.: "naNazwaTowaru\tvt0\tp1000" dla rozkazu "trline"(linia sprzedaży). Ciągu parametrów **nie należy poprzedzać ani kończyć znakiem TAB**, kontrolka zadba o właściwe opakowanie przekazanych danych.

Jeżeli polecenie protokołu nie posiada pól opcjonalnych jako drugi parametr w wywołaniu funkcji należy przekazać pusty ciąg znaków.

## ReceiveFrame

Opis:

funkcja umożliwia odczytanie ramki protokołu odesłanej przez drukarkę fiskalną.

Wywołanie funkcji:

```
string ReceiveFrame();
```

Zwracana wartość:

ciąg znaków odebrany od drukarki fiskalnej lub kod błędu.

Przykład wywołania

```
ReceiveFrame();
```

Uwagi:

Funkcja kontroluje treść odebranych danych, poprawny wynik przychodzi tylko wtedy gdy z odebranego od drukarki fiskalnej ciągu znaków będzie można wyodrębnić ramkę protokołu.

W odpowiedzi zawsze przychodzi tylko jedna ramka protokołu. Jeżeli spodziewamy się więcej niż jednej odpowiedzi odczyt należy powtórzyć.

Po wysłaniu rozkazu należy odczekać chwilę przed odczytaniem odpowiedzi.

Jeżeli funkcja ReceiveFrame wywołana bezpośrednio po przesłaniu rozkazu do drukarki zwróci wynik negatywny może to oznaczać, że czas pomiędzy zapisem i próbą odczytu nie jest wystarczający. W przypadku negatywnej odpowiedzi próbę odczytu można powtórzyć.

## SendFile

Opis:

funkcja umożliwia przesłanie do drukarki fiskalnej danych zapisanych w pliku tekstowym.

Wywołanie funkcji:

```
int SendFile( LPCTSTR fileName, bool readOn);
```

Przekazywane parametry:

fileName – adres pliku,

readOn – włączenie/wyłączenie odczytu odpowiedzi drukarki.

Zwracana wartość:

„0” jeżeli wykonano poprawnie lub kod błędu ze znakiem „-”.

Przykład wywołania

```
SendFile( "C:\\plik.txt", true);
```

Uwagi:

Zawartość pliku tekstowego jest kontrolowana przed wysłaniem.

Funkcja wyodrębnia z pliku ramki protokołu i wysyła je pojedynczo.

Operacja wysyłania danych z pliku może zostać przerwana jeżeli podczas próby wysłania danych do drukarki wystąpi błąd

Jeżeli "readOn" jest ustawione na "true" funkcja odczytuje odpowiedzi drukarki i loguje je do pliku „SendFile\_Result.txt”. W przeciwnym wypadku należy samodzielnie odczytywać odpowiedzi drukarki aby nie dopuścić do zapełnienia kolejki odbiorczej.



## WaitForResponse

Opis:

funkcja umożliwia czekanie na odpowiedź urządzenia przez określoną ilość czasu.

Wywołanie funkcji:

```
string WaitForResponse( long timeout, bool parser );
```

Przekazywane parametry:

timeout – czas w sekundach,

parser – czy z odebranych danych wyodrębnić ramkę protokołu.

Zwracana wartość:

ciąg znaków odebrany od drukarki fiskalnej lub kod błędu.

Przykład wywołania

```
WaitForResponse( 1 , true );
```

Uwagi:

Jeżeli „parser == true” funkcja kontroluje treść odebranych danych, poprawny wynik przychodzi tylko wtedy gdy z odebranego od drukarki fiskalnej ciągu znaków będzie można wyodrębnić ramkę protokołu.

### 3. Logowanie danych.

#### LogBegin

Opis:

Funkcja umożliwia włączenie logowania danych do pliku tekstowego.

Wywołanie funkcji:

```
int LogBegin( bool data_log, bool entry_log );
```

Przekazywane parametry:

data\_log – włączenie logowanie danych wysłanych/odebranych do/z urządzenia,

entry\_log – włączenie logowania punktów wejścia/wyjścia wywołanych funkcji.

Zwracana wartość:

„0” jeżeli wykonano poprawnie lub kod błędu ze znakiem „-”.

Przykład wywołania

```
LogBegin( true , true );
```

Uwagi:

Nie ma możliwości włączenia logowania z ustawieniem obu przekazywanych parametrów na false.

Nie ma możliwości zmiany konfiguracji po utworzeniu pliku loga, w takiej sytuacji należy przerwać logowanie i włączyć je z nowymi ustawieniami.

#### LogEnd

Opis:

Funkcja umożliwia wyłączenie logowania danych do pliku tekstowego.

Wywołanie funkcji:

```
int LogEnd();
```

Zwracana wartość:

„0” jeżeli wykonano poprawnie lub kod błędu ze znakiem „-”.

Przykład wywołania

```
LogEnd();
```

## 4. Funkcje pomocnicze.

### PosnetLibGetVersion

Opis:

funkcja umożliwia odczytanie wersji zarejestrowanej w systemie kontrolki PosnetLib.

Wywołanie funkcji:

```
string PosnetLibGetVersion();
```

Zwracana wartość:

ciąg znaków, oznaczenie wersji kontrolki zarejestrowanej w systemie.

Przykład wywołania:

```
PosnetLibGetVersion();
```

### CleanUp

Opis:

Funkcja umożliwia usunięcie wszystkich danych dotyczących komunikacji z drukarką fiskalną, buforów kontrolki oraz kolejek odczytu/zapisu do/z urządzenia.

Wywołanie funkcji:

```
void CleanUp();
```

Przykład wywołania:

```
CleanUp();
```

## CreateFrame

Opis:

funkcja umożliwia zbudowanie ramki protokołu Posnet na podstawie przekazanych parametrów.

Wywołanie funkcji:

```
string CreateFrame( LPCTSTR requestId, LPCTSTR requestParams );
```

Przekazywane parametry:

requestId – identyfikator polecenia,

requestParams – parametry polecenia rozdzielone znakiem TAB '09'h.

Zwracana wartość:

ciąg znaków, ramka protokołu Posnet lub kod błędu.

Przykład wywołania

```
CreateFrame( "dailyrep", "da2013-08-19");
```

Uwagi:

Przekazywane parametry w „requestId” oraz „requestParams” powinny być zgodne z Instrukcją Programisty Protokołu Posnet dla konkretnego typu drukarki.

Pierwszy parametr w wywołaniu funkcji musi być niezerowy ( != "" ).

Jeżeli w „requestParams” chcemy przekazać więcej niż jeden parametr dla polecenia protokołu należy je rozdzielić znakiem TAB '09'h np.: „naNazwaTowaru\tvt0\tpr1000” dla rozkazu „trline”(linia sprzedaży). Ciągu parametrów **nie należy poprzedzać ani kończyć znakiem TAB**, kontrolka zadba o właściwe opakowanie przekazanych danych.

Jeżeli polecenie protokołu nie posiada pól opcjonalnych jako drugi parametr w wywołaniu funkcji należy przekazać pusty ciąg znaków.

Funkcja nie kontroluje poprawności przekazanych danych.

## GetCRC

Opis:

funkcja umożliwia obliczenie sumy kontrolnej dla samodzielnie zbudowanej ramki protokołu.

Wywołanie funkcji:

```
string GetCRC( LPCTSTR request);
```

Przekazywane parametry:

request – dane do obliczenia sumy kontrolnej.

Zwracana wartość:

ciąg znaków, suma kontrolna (4 znaki z zakresu 0-9 A-F) poprzedzona znakiem '#' np.:  
"#A4B5".

Przykład wywołania

```
GetCRC( "dailyrep\tda2013-08-19");
```

Uwagi:

Funkcja nie kontroluje poprawności przekazanych danych.

Dane do obliczenia sumy kontrolnej należy przekazywać zgodnie z specyfikacją protokołu.

## GetToken

Opis:

funkcja umożliwia pobranie numeru Token dla kolejnej budowanej ramki protokołu.

Wywołanie funkcji:

`string GetToken();`

Zwracana wartość:

ciąg znaków, numer token poprzedzony znakiem '@' np.: "@1".

Przykład wywołania

`GetToken();`

Uwagi:

Funkcja umożliwia zachowanie ciągłości numeru token w przypadku ręcznego budowania ramek.

## GetParamValue

Opis:

funkcja umożliwia pobranie wartości dowolnego parametru z przekazanej ramki protokołu.

Wywołanie funkcji:

`string GetParamValue(LPCTSTR frame, LPCTSTR paramID);`

Przekazywane parametry:

frame – ramka protokołu komunikacyjnego,

paramID – identyfikator parametru zgodny ze specyfikacją.

Zwracana wartość:

ciąg znaków, wartość parametru wyodrębniona z przesłanej ramki.

Przykład wywołania

`GetParamValue( "<stx>trline\t@67\tnaTowar A\tvt0\tpr999\t#997B<etx>" , "na" );`

Uwagi:

Za pomocą funkcji można wyodrębnić również wartość token lub kod błędu.

## **GetErrText**

Opis:

funkcja umożliwia pobranie opisu kodu błędu.

Wywołanie funkcji:

```
string GetErrText( long errorCode);
```

Przekazywane parametry:

errorCode – kod błędu.

Zwracana wartość:

ciąg znaków, wartość opis przekazanego kodu błędu.

Przykład wywołania

```
GetErrText( -10 );
```

Uwagi:

Za pomocą funkcji można pobrać również opis kodu błędu protokołu komunikacyjnego.

## 5. Informacje pomocnicze.

### Kody błędów wewnętrznych kontrolki.

- 1 - Błąd podczas wykonywania polecenia.
- 5 - Nawiązano już połączenie.
- 6 - Brak aktywnego połączenia.
- 7 - Nieprawidłowy typ połączenia.
- 10 - Nie udało się nawiązać połączenia COM.
- 11 - Błąd ustawienia parametrów portu.
- 12 - Błąd zapisu danych(COM).
- 13 - Błąd odczytu danych(COM).
- 14 - Port nie został otwarty, a ktoś próbuje wysłać lub odebrać dane.
- 15 - Nieprawidłowa prędkość transmisji danych.
- 16 - Nieprawidłowe ustawienie sterowania przepływem.
- 20 - Nie udało się nawiązać połączenia TCP.
- 21 - Błąd tworzenia gniazda sieciowego.
- 22 - Błąd zapisu danych(TCP).
- 23 - Błąd odczytu danych(TCP).
- 24 - Nie nawiązano połączenia TCP, a ktoś próbuje wysłać lub odebrać dane.
- 25 - Błąd inicjalizacji WSA.
- 30 - Nieprawidłowe dane wejściowe, nie przekazano parametru.
- 31 - Brak danych do odczytu.
- 32 - Zupełny brak odpowiedzi (w wyznaczonym czasie).
- 33 - Nie jest możliwe wygenerowanie ramki (zbyt długa - bufor zbyt krótki).
- 34 - Upłynął czas oczekiwania, nie otrzymano znacznika końca ramki.
- 35 - Wykryto błąd danych w buforze odbiorczym.
- 50 - Logowanie danych zostało już włączone.
- 51 - Logowanie danych nie zostało włączone.
- 52 - Błędna konfiguracja logowania danych.
- 53 - Błąd tworzenia pliku loga.

### Wsparcie techniczne.

Wsparcie techniczne w przypadku problemów z wykorzystaniem kontrolki lub jej działaniem można uzyskać pod adresem [helpdesk@posnet.com](mailto:helpdesk@posnet.com)