Драйвер BluePad-50

Руководство программиста

**Функции и свойства для работы с OLE – сервером**

Все функции возвращают значение типа Bool (булево). Значение функции false означает, что функция не отработала. Ошибку выполнения можно посмотреть, обратившись к свойствам LastErrorCode и (или) LastErrorMsg, где указаны код и описание возникшей ошибки.

1. **Открытие COM порта терминала (OpenPort)**

**openPort(\_ timeout:Int):Bool**

**Назначение:** Функция предназначена для открытия СОМ порта, к которому подключен терминал BLUEPAD-50. Эту функцию следует вызывать первой для начала работы OLE сервера с терминалом. Терминал в момент вызова функции **ОБЯЗАТЕЛЬНО** должен быть включен, иначе СОМ порт открыт не будет.

**Параметры:**

Port – Строка с именем порта, например “COM1”, “COM2”.

BaudRate – целое число, скорость СОМ порта, к которому подключен терминал. Рекомендуемая скорость подключения - 115200 бод.

1. **Закрытие СОМ порта терминала (ClosePort)**

**closePort():Bool**

**Назначение:** Функция предназначена для закрытия СОМ порта, открытого функцией OpenPort.

**3. Свойство проверки состояния COM порта (PortOpened)**

**Property portOpened : Bool**

Назначение: Свойство показывает состояниеСОМ порта (открыт или закрыт).

1. **Проверка связи с терминалом (Ping)**

**ping():Bool**

1. **Проверка установленного в терминале времени (GetDateTime)**

**GetDateTime(out DT: String): Bool**

**Назначение:** Функция предназначена для проверки установленного в терминале времени и даты.

**Параметры:**

DT – выходной параметр, возвращает строковое текущую дату и время в терминале в строковом виде.

1. **Установка времени и даты в терминале (SetDateTime)**

**SetDateTime(Year: UInt16; Month: UInt8; Day: UInt8; Hour: UInt8; Minute: UInt8; Sec: UInt8): Bool**

**Назначение:** Функция предназначена для установки в терминале текущих времени и даты.

**Параметры:**

Year – год

Month – месяц

Day – число месяца

Hour – часы

Minute – минуты

Sec – секунды

1. **Информация о терминале (GetPinpadInfo)**

**GetPinpadInfo(out ModelName: String; out SerialNumber: String; out SoftVer: String; out TerminalID: String; out MenuType: UInt8): Bool**

Назначение: Функция предназначена для просмотра информации о приборе.

Параметры:

ModelName – выходной параметр, возвращает строку с названием модели прибора.

SerialNumber – выходной параметр, возвращает строку с серийным номером прибора.

SoftVer – выходной параметр, строка с версией ПО прибора.

TerminalID – выходной параметр, строка с ID прибора.

MenuType – выходной параметр, число тип меню прибора.

Возможные значения MenuType:

1 - BOR\_MENU\_TYPE\_TRADING

2 - BOR\_MENU\_TYPE\_TRADING\_REF

3 - BOR\_MENU\_TYPE\_TRADING\_CASHBACK

4 - BOR\_MENU\_TYPE\_HOTEL\_RENT\_A\_CAR

5 - BOR\_MENU\_TYPE\_BAR\_RESTAURANT

6 - BOR\_MENU\_TYPE\_BANK\_CHANGE

7 - BOR\_MENU\_TYPE\_TEST

1. **Установка таймаутов (SetTimeouts)**

**SetTimeouts(timeoutTransaction:Int):Bool**

**Назначение:** Функция предназначена для установки таймаута ожидания виполнения транзакции

**Параметры:**

TimeoutTransaction – таймаут цикла транзакции, ms. По умолчанию 40 000 ms.

1. **Последнее сообщение об ошибке(LastErrorMsg)**

**LastErrorMsg ():String**

**Назначение:** Функция предназначена просмотра последнего сообщения об ошибке.

**Параметры:**

нет.

1. **Код последней ошибки(LastError** **Code)**

**LastErrorCode ():Int**

**Назначение:** Функция предназначена просмотра кода последней ошибки.

**Параметры:** нет.

**Функции управления транзакциями**

После удачного выполнения транзакции эти функции возвращают информацию о транзакции в виде набора данных в строке outValues, разделенных символом «;» :

“par1;par2;par3;par4;par5;par6;par7;par8”? где

par1 - TRANSACTION\_RESULT

par2 - TRANSACTION\_ERROR\_CODE

par3 - RRN

par4 - HOST\_AUTH\_ID

par5 - HOST\_CODE

par6 - HOST\_RN

par7 - AMOUNT\_AUTHORISED

par8 - TRANSACTION\_SEQUENCE\_COUNTER

1. **Тестовое соединение с сервером (TestConnection)**

**testConnection(out outValues: String): Bool**

**Назначение:** Проверить связь с сервером.

**Параметры:** Нет.

1. **Покупка (Purchase)**

**purchase(amount: Int; out outValues: String): Bool**

**Назначение:** Оплатить за покупку.

**Параметры:**

Amount – Целое число, сумма в копейках.

1. **Покупка (PurchaseCashback)**

**purchaseCashback(amount: Int; cashack: Int; out outValues: String): Bool**

**Назначение:** Оплатить за покупку с кэшбеком.

**Параметры:**

Amount – Целое число, сумма в копейках.

Cashback – Целое число, сумма в копейках.

1. **Возврат (VoidOfPurchase)**

**voidOfPurchase(amount: Intr; const RRN: String; const authID: String; out outValues: String): Bool**

**Назначение:** Возврат денег за покупку.

**Параметры:**

Amount – Целое число, сумма в копейках.

RRN – Строка, код транзакции.

AuthID – строка, код авторизации.

1. **Еще один вид возврата (PurchaseReturn)**

**purchaseReturn (amount: Int; const RN: String; out outValues: String): Bool**

**Назначение:** Возврат денег за покупку.

**Параметры:**

Amount – Целое число, сумма в копейках.

RN – Строка, RN транзакции.

1. **Отмена чека (UniversalReversalAdvice)**

**universalReversalAdvice (amount: Int; numCheck: Int; out outValues: String): Bool**

**Назначение:** Отмена чека.

**Параметры:**

Amount – Целое число, сумма в копейках.

NumCheck – Целое число, номер чека в терминале.

1. **Авторизация (Authorization)**

**authorization (amount: Int; out outValues: String): Bool**

**Назначение:** Заблокировать сумму на счету.

**Параметры:**

Amount – Целое число, сумма в копейках.

1. **Подтверждение авторизации (AuthorizationConfirm)**

**authorizationConfirm (amount: Int; const RRN: String; out outValues: String): Bool**

Назначение: Подтвердить заблокированную сумму на счету.

Параметры:

Amount – Целое число, сумма в копейках.

RRN – Строка, код транзакции.

1. **Отказ от авторизации (VoidAuthorization)**

**voidAuthorization(amount: Int; const RRN: String; const AuthID: String; out outValues: String): Bool**

**Назначение:** Заблокировать сумму на счету.

**Параметры:**

Amount – Целое число, сумма в копейках.

RRN – Строка, код транзакции.

AuthID – строка, код авторизации.

В конце транзакций запрашивается отчет по чеку (GetReceiptTags). Результаты выполнения отчета можно получить при помощи функций GetLastTags и GetLastTag (см.ниже).

1. **Отмена операции (UniversalReversal)**

**universalReversal (Amount: Int; const numCheck: Int; out outValues: String): Bool**

**Назначение:** Отменить указанную операцию по номеру чека.

**Параметры:**

Amount – Целое число, сумма в копейках.

NumCheck – Целое число, номер чека в терминале.

1. **Закрытие дня (EndOfDay)**

**endOfDay (out outValues: String): Bool**

**Назначение:** Закрыть день. Перед выполнением «EndOfDay» необходимо распечатать ​​ежедневный отчет. Его нужно распечатать ДО транзакции, потому что, если конец дня будет успешным, он удалит все транзакции из журнала, и информация о транзакциях больше не будет доступна.

**Параметры:** Нет.

1. **Проверка баланса (BalanceInquiry)**

**balanceInquiry (out outValues: String):Bool**

**Назначение:** Проверить баланс на карте.

**Параметры:** Нет.

1. **Обмен ключами с хостом (KeyChange)**

**keyChange (out outValues: String): Bool**

**Назначение:** Обмен ключами с хостом.

**Параметры:** Нет.

**ОТЧЕТЫ**

1. **Показать тэги последнего отчета (GetLastTags)**

**getLastTags(out tagCodes: String): Bool;**

**Назначение:** После удачного выполнения любого отчета его результат помещается во временный буфер, доступный к просмотру. Функция **GetLastTags** позволяет увидеть присутствующие в нем тэги, а функция GetLastTag выводит значение конкретного тэга в последнем отчете. После вызова следующего отчета информация в буфере заменяется на его результаты.

**Параметры:**

TagCodes – Строка, список тэгов в шестнадцатеричном виде, разделенных символом «;», которые присутствуют в последнем отчете. Например: «81;DF07;DF08».

1. **Показать конкретный тэг последнего отчета (GetLastTag)**

**getLastTag(Tag: UInt16; out valueStr: String): Bool;**

**Назначение:** После удачного выполнения любого отчета его результат помещается во временный буфер, доступный к просмотру. Функция GetLastTag выводит значение конкретного тэга в последнем отчете. После вызова следующего отчета информация в буфере заменяется на его результаты.

**Параметры:**

Tag – число, код тэга.

ValueStr – строка, значение тэга.

1. **Отчет по чеку (GetReceiptTags)**

**getReceiptTags(const RequestTags: String): String?**

**Назначение:** В конце транзакции эту команду можно использовать для получения информации о последней выполненной транзакции. Таким образом, может быть напечатан чек.

**Параметры:**

RequestTags – Строка, список затребованных тэгов в шестнадцатеричном виде, разделенных символом «;». Например: «81;DF07;DF08».

1. **Отчет по транзакциям (GetReportTags)**

**getReportTags(recordType: Byte; const RequestTags: String): String?**

**Назначение:** Команда используется, когда нам нужно получить некоторую информацию для какой-то конкретной транзакции. Мы можем получить информацию для первой транзакции (1 - ПЕРВАЯ ЗАПИСЬ), а затем вызвать одну и ту же команду несколько раз получить информацию для всех следующих записей одну за другой (с параметром 2 - СЛЕДУЮЩИЕ ЗАПИСИ). Аналогично можно получить последнюю запись журнала (3 - ПОСЛЕДНЯЯ ЗАПИСЬ), а потом перебором пройти все остальные записи к началу (4 - ПРЕДЫДУЩАЯ ЗАПИСЬ). Итерации можно использовать в цикле до тех пор, пока функция не вернет FALSE, то есть больше записей в журнале нет.

**Параметры:**

RecordType: порядок затребованной записи

Возможные значения RecordType:

1 – показать информацию для первой записи в журнале

2 – показать информацию для следующей записи в журнале

3 – показать информацию для последней записи в журнале

4 – показать информацию для предыдущей записи в журнале

RequestTags – Строка, список затребованных тэгов в шестнадцатеричном виде, разделенных символом «;». Например: «81;DF07;DF08».

1. **Отчет по конкретной транзакции (GetReportTagsByStan)**

**getReportTagsByStan(transNumber: Int; const RequestTags:String): String?**

**Назначение:** Команда используется для получения информации по конкретной транзакции. **Параметры:**

TransNumber: номер затребованной транзакции

RequestTags – Строка, список затребованных тэгов в шестнадцатеричном виде, разделенных символом «;». Например: «81;DF07;DF08».

1. **Информация о журнале (GetReportInfo)**

getReportInfo(out recordCount: UInt16): Bool

**Назначение:** Используется для определения количества записей в журнале.

**Параметры:**

RecordCount – выходной параметр, число, показывает текущее количество записей в журнале транзакций.