**Tabla de Contenido**

[Historial de Versiones 4](#_Toc534148029)

[Requerimientos 4](#_Toc534148030)

[Productos 4](#_Toc534148031)

[Software 5](#_Toc534148032)

[Hardware 5](#_Toc534148033)

[App Config 5](#_Toc534148034)

[Bill Dispenser [Fujitsu F53] 5](#_Toc534148035)

[Declaration 5](#_Toc534148036)

[Public Methods 5](#_Toc534148037)

[void open() 5](#_Toc534148038)

[void close() 5](#_Toc534148039)

[void enable() 6](#_Toc534148040)

[bool isConnection() 6](#_Toc534148041)

[void enable() 6](#_Toc534148042)

[void returnCash() 6](#_Toc534148043)

[Private Methods 6](#_Toc534148044)

[void createCode() 6](#_Toc534148045)

[void setFreeDevice() 6](#_Toc534148046)

[void sendMessage() 7](#_Toc534148047)

[void openConnection() 7](#_Toc534148048)

[void Config\_inicial() 7](#_Toc534148049)

[void Mechal\_Reset() 7](#_Toc534148050)

[void Cancel() 7](#_Toc534148051)

[void SensorLevelInformation() 7](#_Toc534148052)

[void CheckConfig() 7](#_Toc534148053)

[int search() 8](#_Toc534148054)

[bool match() 8](#_Toc534148055)

[byte[] getPositions() 8](#_Toc534148056)

[Bill Acceptor [SC Advance MEI ] 8](#_Toc534148057)

[Declaration 8](#_Toc534148058)

[Public Methods 8](#_Toc534148059)

[void open() 8](#_Toc534148060)

[void close() 8](#_Toc534148061)

[bool isConnection() 9](#_Toc534148062)

[void enable() 9](#_Toc534148063)

[void disable() 9](#_Toc534148064)

[double[2] getCashDesposite() 9](#_Toc534148065)

[Private Methods 9](#_Toc534148066)

[AcceptorSCAd() 9](#_Toc534148067)

[void openConnection() 9](#_Toc534148068)

[void configDefault() 10](#_Toc534148069)

[void connectedHandle 10](#_Toc534148070)

[Coin Dispenser [Azkoyen ComboT] 10](#_Toc534148071)

[Declaration 10](#_Toc534148072)

[void open() 10](#_Toc534148073)

[void close() 10](#_Toc534148074)

[bool isConnection() 10](#_Toc534148075)

[void enable() 10](#_Toc534148076)

[void returnCash() 11](#_Toc534148077)

[Private Methods 11](#_Toc534148078)

[void enableContainerCoin() 11](#_Toc534148079)

[bool getDeviceId() 11](#_Toc534148080)

[byte[] getNumberSerie() 11](#_Toc534148081)

[void checkStatusSensors() 11](#_Toc534148082)

[Coin Acceptor [Azkoyen ComboT] 11](#_Toc534148083)

[Declaration 11](#_Toc534148084)

[Public Methods 12](#_Toc534148085)

[void open() 12](#_Toc534148086)

[void close() 12](#_Toc534148087)

[bool isConnection() 12](#_Toc534148088)

[void enable() 12](#_Toc534148089)

[void disable() 12](#_Toc534148090)

[double[] getCashDesposite() 12](#_Toc534148091)

[Private Methods 13](#_Toc534148092)

[bool getDeviceId() 13](#_Toc534148093)

[void setConfigDefaultHoppers() 13](#_Toc534148094)

[void setInibitCoins() 13](#_Toc534148095)

[void clearCounterMoney() 13](#_Toc534148096)

[void emptyMoneyBox() 13](#_Toc534148097)

[Clases Complementarias 13](#_Toc534148098)

[CCTalk 13](#_Toc534148099)

[Declaration 13](#_Toc534148100)

[Public Methods 14](#_Toc534148101)

[static CCTalk GetInstance() 14](#_Toc534148102)

[SerialPort openConnection() 14](#_Toc534148103)

[void setMessage() 14](#_Toc534148104)

[void getMessage() 14](#_Toc534148105)

[bool getDeviceId() 14](#_Toc534148106)

[void setDevices() 14](#_Toc534148107)

[void sendMessage() 14](#_Toc534148108)

[Private Methods 15](#_Toc534148109)

[CCTalk() 15](#_Toc534148110)

[string GetNameDevice() 15](#_Toc534148111)

[void CleanEcho() 15](#_Toc534148112)

[int search() 15](#_Toc534148113)

[bool match() 15](#_Toc534148114)

[string ByteArrayToString() 15](#_Toc534148115)

[byte[] setChecksum() 15](#_Toc534148116)

# 

# Historial de Versiones

|  |  |  |
| --- | --- | --- |
| Versión | Fecha | Descripción |
| 1.00 | 01/01/2019 | Revisión Inicial |
|  |  |  |
|  |  |  |
|  |  |  |

# Requerimientos

## Productos

La LibreriaKioscoCash.dll está diseñada para integrar diferentes dispositivos de diferentes marcas con la finalidad de crear un entorno flexible y seguro, así como cubrir todas las necesidades de cada cliente. Esta versión íntegra dispositivos de empresas líderes mundiales en soluciones en medios de pago como Fujistu, Azkoyen, MEI, CPI, Asahiseiko.

|  |  |  |
| --- | --- | --- |
| Marca | Modelo | Descripción |
| Fujistsu | F53 | Imagen relacionada |
| MEI | SCAdvance | Resultado de imagen para mei sc advance |
| Azkoyen | ComboT |  |

## Software

Esta librería está hecha en el lenguaje de programación C# y soporta el .NET Framework 4.5 o superior. (Windows 7 / Windows 10)

## Hardware

La computadora que utilizará la librería debe tener puertos serial o puerto USB. La compatibilidad con USB es solo disponible en el sistema operativo Windows a través de un controlador de comunicaciones serial a USB (FTDI drivers).

# App Config

Dentro del archivo App.config del programa de integración se realizará la configuración de los puertos COM de los dispositivos, así como la clasificación de los dispositivos de acuerdo con su funcionamiento.

<appSettings>

<add key="BillDispenser" value="F53"/>

<add key="COMBillDispenser" value="COM#"/>

<add key="BillAcceptor" value="SCAdvance"/>

<add key="COMBillAcceptor" value="COM#"/>

<add key="CoinDispenser" value="ComboT"/>

<add key="CoinAcceptor" value="ComboT"/>

<add key="COMComboT" value="COM#"/>

</appSettings>

# Bill Dispenser [Fujitsu F53]

## Declaration

static name factory = new FactoryDevice();

IDispenser name = factory.GetBillDispenser();

## Public Methods

***Estos métodos son públicos formando parte de la interfaz IDispenser y se encuentran dentro de “LibreriaKioscoCash.dll”.***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Method** | **Parameters** | **Return** | **Version** | **Description** |
| void open() | ━ | ━ | 1.00 | Establece conexión y configuración del dispositivo |
| *Este método tardará más tiempo cuando el dispositivo se encienda por primera vez, se hará la configuración de los billetes y casetes. Posteriormente el dispositivo sabrá que ya está configurado.* | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Method** | **Parameters** | **Return** | **Version** | **Description** |
| void close() | ━ | ━ | 1.00 | Cierra la conexión con el dispositivo. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Method** | **Parameters** | **Return** | **Version** | **Description** |
| void enable() | ━ | ━ | 1.00 | Habilita la comunicación con el dispositivo. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Method** | **Parameters** | **Return** | **Version** | **Description** |
| bool isConnection() | ━ | bool | 1.00 | Pregunta al dispositivo si se encuentra conectado, regresando un valor True. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Method** | **Parameters** | **Return** | **Version** | **Description** |
| void enable() | ━ | ━ | 1.00 | Habilita el dispositivo para el retiro de billetes. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Method** | **Parameters** | **Return** | **Version** | **Description** |
| void returnCash() | int[3] | ━ | 1.00 | Dispensa la cantidad de billetes por denominación. |
| Example:  int[3] billCount = { cantidad de $20.00 MXN, cantidad de $50.00 MXN, cantidad de $100.00 MXN };  name.returnCash(billCount);  *Los casetes del dispositivo se encuentran configurados manualmente con el tamaño de cada billete, si se desea cambiar las denominaciones a dispensar, contactar al soporte técnico para la actualización de los casetes.* | | | | |

## Private Methods

***Estos métodos son internos del dispositivo para un correcto funcionamiento y solo se encuentran dentro de la Clase Privada “DispenserF53”.***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Method** | **Parameters** | **Return** | **Version** | **Description** |
| void createCode() | byte[] | ━ | 1.00 | Crear el código completo para la comunicación con el dispositivo (RS232c) |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Method** | **Parameters** | **Return** | **Version** | **Description** |
| void setFreeDevice() | ━ | ━ | 1.00 | Libera el dispositivo en caso de que se encuentre bloqueado con un error FC00. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Method** | **Parameters** | **Return** | **Version** | **Description** |
| void sendMessage() | byte[] | ━ | 1.00 | Crea la secuencia necesaria para enviar y recibir peticiones con el dispositivo. Antes se necesita usar el método createCode. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Method** | **Parameters** | **Return** | **Version** | **Description** |
| void openConnection() | ━ | ━ | 1.00 | Obtiene el puerto del dispositivo del App.config y abre la comunicación. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Method** | **Parameters** | **Return** | **Version** | **Description** |
| void Config\_inicial() | ━ | ━ | 1.00 | Configura el dispositivo de acuerdo con la longitud y grosor del billete por cada casete. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Method** | **Parameters** | **Return** | **Version** | **Description** |
| void Mechal\_Reset() | ━ | ━ | 1.00 | Se genera un reseteo mecánico del dispositivo. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Method** | **Parameters** | **Return** | **Version** | **Description** |
| void Cancel() | ━ | ━ | 1.00 | Borra el código de error o comando que se haya genera sobre el dispositivo. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Method** | **Parameters** | **Return** | **Version** | **Description** |
| void SensorLevelInformation() | ━ | ━ | 1.00 | Obtiene el estado físico de todos los sensores que se encuentran en el dispositivo. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Method** | **Parameters** | **Return** | **Version** | **Description** |
| void CheckConfig() | ━ | ━ | 1.00 | Cada que se abre una conexión con el dispositivo obtiene la configuración de este y si no se encuentra configurado correrá la configuración inicial. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Method** | **Parameters** | **Return** | **Version** | **Description** |
| int search() | byte[], byte[] | int | 1.00 | Función complementaria que permite encontrar una secuencia de bytes consecutivos en un arreglo. Regresando la última posición. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Method** | **Parameters** | **Return** | **Version** | **Description** |
| bool match() | byte[],byte[] ,int | bool | 1.00 | Esta es una subfunción del método anterior search(), este devuelve un valor true si se encontró la secuencia y false en caso contrario. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Method** | **Parameters** | **Return** | **Version** | **Description** |
| byte[] getPositions() | byte[], byte[] | ━ | 1.00 | Este método regresa las posiciones de una secuencia en un arreglo de bytes. |

# Bill Acceptor [SC Advance MEI ]

## Declaration

static name factory = new FactoryDevice();

IAcceptor name = factory.GetBillAcceptor();

## Public Methods

***Estos métodos son públicos formando parte de la interfaz IAcceptor y se encuentran dentro de “LibreriaKioscoCash.dll”.***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Method** | **Parameters** | **Return** | **Version** | **Description** |
| void open() | ━ | ━ | 1.00 | Establece conexión y configuración del dispositivo |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Method** | **Parameters** | **Return** | **Version** | **Description** |
| void close() | ━ | ━ | 1.00 | Cierra la conexión con el dispositivo. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Method** | **Parameters** | **Return** | **Version** | **Description** |
| bool isConnection() | ━ | bool | 1.00 | Pregunta al dispositivo si se encuentra conectado. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Method** | **Parameters** | **Return** | **Version** | **Description** |
| void enable() | ━ | ━ | 1.00 | Habilita la aceptación de billetes. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Method** | **Parameters** | **Return** | **Version** | **Description** |
| void disable() | ━ | ━ | 1.00 | Deshabilita la aceptación de billetes. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Method** | **Parameters** | **Return** | **Version** | **Description** |
| double[2] getCashDesposite() | ━ | double[] | 1.00 | Regresa la cantidad de billetes por denominación. |
| Example:  double[2] recibido = name.getCashDesposite();  recibido[0] → Regresa la denominación que se depositó ($20 MXN,$50 MXN,$100 MXN,$200 MXN,$500 MXN)  recibido[1] → Empty  *Los billetes de $1000MXN se encuentran deshabilitados, si se desean habilitar, contactar al soporte técnico para la actualización de la librería*. | | | | |

## Private Methods

***Estos métodos son internos del dispositivo para un correcto funcionamiento y solo se encuentran dentro de la Clase Privada “AcceptorSCAd”.***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Instance** | **Parameters** | **Return** | **Version** | **Description** |
| AcceptorSCAd() | ━ | ━ | 1.00 | Instancia de la clase para el control interno de los eventos del dispositivo. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Method** | **Parameters** | **Return** | **Version** | **Description** |
| void openConnection() | ━ | ━ | 1.00 | Obtiene el puerto del dispositivo del appConfig y abre la comunicación con PowerUp.A |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Method** | **Parameters** | **Return** | **Version** | **Description** |
| void configDefault() | ━ | ━ | 1.00 | Configura las denominaciones a aceptar. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Event** | **Parameters** | **Return** | **Version** | **Description** |
| void connectedHandle | object sender, EventArgs e | ━ | 1.00 | Cuando el dispositivo recibe el evento de conectado, se cargará el método configDefault. |

# Coin Dispenser [Azkoyen ComboT]

## Declaration

static name factory = new FactoryDevice();

IDispenser name = factory.GetCoinDispenser();

**Public Methods**

***Estos métodos son públicos formando parte de la interfaz IDispenser y se encuentran dentro de “LibreriaKioscoCash.dll”.***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Method** | **Parameters** | **Return** | **Version** | **Description** |
| void open() | ━ | ━ | 1.00 | Establece conexión y configuración del dispositivo |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Method** | **Parameters** | **Return** | **Version** | **Description** |
| void close() | ━ | ━ | 1.00 | Cierra la conexión con el dispositivo. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Method** | **Parameters** | **Return** | **Version** | **Description** |
| bool isConnection() | ━ | bool | 1.00 | Pregunta al dispositivo si se encuentra conectado, regresando un valor True. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Method** | **Parameters** | **Return** | **Version** | **Description** |
| void enable() | ━ | ━ | 1.00 | Habilita el dispositivo para el retiro de monedas. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Method** | **Parameters** | **Return** | **Version** | **Description** |
| void returnCash() | int[] | ━ | 1.00 | Dispensa la cantidad de monedas por denominación. |
| Example:  int[3] coinCount = { cantidad de $1.00 MXN, cantidad de $5.00 MXN, cantidad de $10.00 MXN };  name.returnCash(coinCount);  *El dispositivo se encuentra configurado para no dispensar las monedas de $2.00, si se desea cambiar la configuración del reciclado de monedas, contactar al soporte técnico para la actualización de la librería.* | | | | |

## Private Methods

***Estos métodos son internos del dispositivo para un correcto funcionamiento y solo se encuentran dentro de la Clase Privada “DispenserCBT”.***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Method** | **Parameters** | **Return** | **Version** | **Description** |
| void enableContainerCoin() | byte | ━ | 1.00 | Habilita los hopper para dispensar. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Method** | **Parameters** | **Return** | **Version** | **Description** |
| bool getDeviceId() | ━ | bool | 1.00 | Obtiene los ID o direcciones de los dispositivos conectados a la tarjeta CCTalk y regresa un valor true si se obtuvieron de manera correcta. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Method** | **Parameters** | **Return** | **Version** | **Description** |
| byte[] getNumberSerie() | byte | ━ | 1.00 | Obtiene el número de serie del dispositivo. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Method** | **Parameters** | **Return** | **Version** | **Description** |
| void checkStatusSensors() | ━ | ━ | 1.00 | Checa los sensores de nivel de monedas de cada hopper. |

# Coin Acceptor [Azkoyen ComboT]

## Declaration

static name factory = new FactoryDevice();

IAcceptor name = factory.GetCoinAcceptor();

## Public Methods

***Estos métodos son públicos formando parte de la interfaz IAcceptor y se encuentran dentro de “LibreriaKioscoCash.dll”.***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Method** | **Parameters** | **Return** | **Version** | **Description** |
| void open() | ━ | ━ | 1.00 | Establece conexión y configuración del dispositivo |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Method** | **Parameters** | **Return** | **Version** | **Description** |
| void close() | ━ | ━ | 1.00 | Cierra la conexión con el dispositivo. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Method** | **Parameters** | **Return** | **Version** | **Description** |
| bool isConnection() | ━ | ━ | 1.00 | Pregunta al dispositivo si se encuentra conectado. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Method** | **Parameters** | **Return** | **Version** | **Description** |
| void enable() | ━ | ━ | 1.00 | Habilita la aceptación de monedas. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Method** | **Parameters** | **Return** | **Version** | **Description** |
| void disable() | ━ | ━ | 1.00 | Deshabilita la aceptación de monedas. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Method** | **Parameters** | **Return** | **Version** | **Description** |
| double[] getCashDesposite() | ━ | double[] | 1.00 | Regresa la cantidad de monedas por denominación. |
| Example:  double[2] recibido = name.getCashDesposite();  recibido[0] → Regresa la denominación que se depositó ($1.00 MXN,$2.00 MXN,$5.00 MXN,$10.00 MXN)  recibido[1] → Regresa la cantidad de monedas y se queda guardado.  *El dispositivo se encuentra configurado para que la moneda de $2.00 MXN se vaya directamente a la alcancía.*  *Los dos tipos de monedas de $50.00 centavos son rechazadas por el dispositivo, se encuentran deshabilitadas, si se desea cambiar la configuración, contactar al soporte técnico para la actualización de la librería*. | | | | |

## Private Methods

***Estos métodos son internos del dispositivo para un correcto funcionamiento y solo se encuentran dentro de la Clase Privada “AcceptorCBT”.***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Method** | **Parameters** | **Return** | **Version** | **Description** |
| bool getDeviceId() | ━ | bool | 1.00 | Obtiene los ID o direcciones de los dispositivos conectados a la tarjeta CCTalk y regresa un valor true si se obtuvieron de manera correcta. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Method** | **Parameters** | **Return** | **Version** | **Description** |
| void setConfigDefaultHoppers() | ━ | ━ | 1.00 | Configura cada denominación a reciclar en un Hopper. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Method** | **Parameters** | **Return** | **Version** | **Description** |
| void setInibitCoins() | ━ | ━ | 1.00 | Configura la denominación a aceptar. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Method** | **Parameters** | **Return** | **Version** | **Description** |
| void clearCounterMoney() | ━ | ━ | 1.00 | Limpia el contador interno de monedas del dispositivo. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Method** | **Parameters** | **Return** | **Version** | **Description** |
| void emptyMoneyBox() | ━ | ━ | 1.00 | Activa la compuerta del dispositivo para mandar las monedas contenidas a la alcancía. |

# Clases Complementarias

## CCTalk

***Esta clase la cual permite compartir funciones entre dispositivos con el fin de optimizar código y generar una librería universal y flexible a cambios.***

## Declaration

CCTalk name = CCTalk.GetInstance()

## Public Methods

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Instance** | **Parameters** | **Return** | **Version** | **Description** |
| static CCTalk GetInstance() | ━ | ━ | 1.00 | Se obtiene la instancia y si no existe se crea una. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Method** | **Parameters** | **Return** | **Version** | **Description** |
| SerialPort openConnection() | string | SerialPort | 1.00 | Regresa el puerto COM de cada dispositivo y si no se encuentra abierto se abre de acuerdo con su configuración. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Method** | **Parameters** | **Return** | **Version** | **Description** |
| void setMessage() | byte[] | ━ | 1.00 | Se envía un arreglo de bytes al dispositivo. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Method** | **Parameters** | **Return** | **Version** | **Description** |
| void getMessage() | ━ | ━ | 1.00 | Se recibe una cadena de bytes del dispositivo. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Method** | **Parameters** | **Return** | **Version** | **Description** |
| bool getDeviceId() | ━ | bool | 1.00 | Obtiene los ID o direcciones de los dispositivos conectados a la tarjeta CCTalk y regresa un valor true si se obtuvieron de manera correcta. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Method** | **Parameters** | **Return** | **Version** | **Description** |
| void setDevices() | ━ | ━ | 1.00 | Clasifica los ID de los dispositivos CCTalk de acuerdo con su nombre y funcionamiento. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Method** | **Parameters** | **Return** | **Version** | **Description** |
| void sendMessage() | byte[], byte[] | ━ | 1.00 | Envía una petición con el código necesario para los dispositivos CCTalk y espera hasta que el dispositivo regrese una respuesta. |

## Private Methods

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Instance** | **Parameters** | **Return** | **Version** | **Description** |
| CCTalk() | ━ | ━ | 1.00 | Se crea una instancia de CCTalk para organizar los dispositivos en un Hashtable. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Method** | **Parameters** | **Return** | **Version** | **Description** |
| string GetNameDevice() | ━ | string | 1.00 | Devuelve el key del dispositivo del App.config |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Method** | **Parameters** | **Return** | **Version** | **Description** |
| void CleanEcho() | ━ | ━ | 1.00 | Al recibir un mensaje por parte del dispositivo, verifica y limpia el echo en la comunicación serial RS232 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Method** | **Parameters** | **Return** | **Version** | **Description** |
| int search() | byte[], byte[] | int | 1.00 | Función complementaria que permite encontrar una secuencia de bytes consecutivos en un arreglo. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Method** | **Parameters** | **Return** | **Version** | **Description** |
| bool match() | byte[],byte[] ,int | bool | 1.00 | Esta es una subfunción del método anterior search(), este devuelve un valor true si se encontró la secuencia y false en caso contrario. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Method** | **Parameters** | **Return** | **Version** | **Description** |
| string ByteArrayToString() | byte[] | string | 1.00 | Convierte un arreglo de bytes de decimal a hexadecimal |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Method** | **Parameters** | **Return** | **Version** | **Description** |
| byte[] setChecksum() | byte[], byte[] | ━ | 1.00 | Calcula el Checksum en módulo 256 para los dispositivos CCTalk |