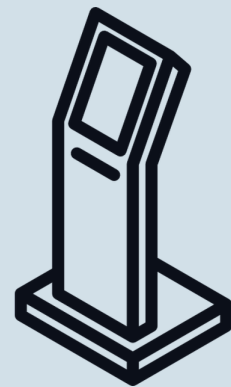


# CUSTOMIZED KIOSK

Integration Library



**Siccob Solutions S.A. de C.V.**

[www.siccob.com.mx](http://www.siccob.com.mx)

Insurgentes Sur No. 1647 , C.P. 03900, Ciudad de México.

Copyright © 2018. Siccob Solutions S.A. de C.V. All Rights Reserved

*La información contenida en este documento es confidencial y propiedad de Siccob Solutions S.A. de C.V..*

*Está prohibida su reproducción, copia o transmisión sin autorización directa de Siccob Solutions S.A. de C.V.*

## Tabla de Contenido

<b>Historial de Versiones</b>	<b>4</b>
<b>Requerimientos</b>	<b>4</b>
Productos	4
Software	5
Hardware	5
<b>App Config</b>	<b>5</b>
<b>Bill Dispenser [Fujitsu F53]</b>	<b>5</b>
Declaration	5
Public Methods	5
void open()	5
void close()	6
void enable()	6
bool isConnection()	6
void enable()	6
void returnCash()	6
Private Methods	7
void createCode()	7
void setFreeDevice()	7
void sendMessage()	7
void openConnection()	7
void Config_inicial()	7
void Mechal_Reset()	7
void Cancel()	7
void SensorLevelInformation()	8
void CheckConfig()	8
int search()	8
bool match()	8
byte[] getPositions()	8
<b>Bill Acceptor [SC Advance MEI ]</b>	<b>8</b>

Declaration	8
Public Methods	9
void open()	9
void close()	9
bool isConnection()	9
void enable()	9
void disable()	9
double[2] getCashDesposite()	9
Private Methods	10
AcceptorSCAd()	10
void openConnection()	10
void configDefault()	10
void connectedHandle	10
<b>Coin Dispenser [Azkoyen ComboT]</b>	<b>10</b>
Declaration	10
void open()	11
void close()	11
bool isConnection()	11
void enable()	11
void returnCash()	11
Private Methods	11
void enableContainerCoin()	11
bool getDeviceId()	12
byte[] getNumberSerie()	12
void checkStatusSensors()	12
<b>Coin Acceptor [Azkoyen ComboT]</b>	<b>12</b>
Declaration	12
Public Methods	12
void open()	12
void close()	12

bool isConnection()	12
void enable()	13
void disable()	13
double[] getCashDesposite()	13
Private Methods	13
bool getDeviceId()	13
void setConfigDefaultHoppers()	13
void setInibitCoins()	13
void clearCounterMoney()	14
void emptyMoneyBox()	14
<b>Clases Complementarias</b>	<b>14</b>
CCTalk	14
Declaration	14
Public Methods	14
static CCTalk GetInstance()	14
SerialPort openConnection()	14
void setMessage()	14
void getMessage()	14
bool getDeviceId()	15
void setDevices()	15
void sendMessage()	15
Private Methods	15
CCTalk()	15
string GetNameDevice()	15
void CleanEcho()	15
int search()	15
bool match()	16
string ByteArrayToString()	16
byte[] setChecksum()	16




## 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 integra dispositivo de empresas líderes mundiales en soluciones en medios de pago como Fujitsu, Azkoyen, MEI, CPI ,Asahiseiko.

Marca	Modelo	Descripción
Fujitsu	F53	
MEI	SCAdvance	
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
<code>void open()</code>	—	—	1.00	Establece conexión y configuración del dispositivo
<i>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.</i>				

Method	Parameters	Return	Version	Description
<code>void close()</code>	—	—	1.00	Cierra la conexión con el dispositivo.

Method	Parameters	Return	Version	Description
<code>void enable()</code>	—	—	1.00	Habilita la comunicación con el dispositivo.

Method	Parameters	Return	Version	Description
<code>bool isConnection()</code>	—	<code>bool</code>	1.00	Pregunta al dispositivo si se encuentra conectado.

Method	Parameters	Return	Version	Description
<code>void enable()</code>	—	—	1.00	Habilita el dispositivo para el retiro de billetes.

Method	Parameters	Return	Version	Description
<code>void returnCash()</code>	<code>int[3]</code>	—	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
<code>void createCode()</code>	<code>byte[]</code>	—	1.00	Crear el código completo para la comunicación con el dispositivo (RS232c)

Method	Parameters	Return	Version	Description
<code>void setFreeDevice()</code>	—	—	1.00	Libera el dispositivo en caso de que se encuentre bloqueado con un error FC00.

Method	Parameters	Return	Version	Description
<code>void sendMessage()</code>	<code>byte[]</code>	—	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
<code>void openConnection()</code>	—	—	1.00	Obtiene el puerto del dispositivo del App.config y abre la comunicación.

Method	Parameters	Return	Version	Description
<code>void Config_inicial()</code>	—	—	1.00	Configura el dispositivo de acuerdo con la longitud y grosor del billete por cada casete.

Method	Parameters	Return	Version	Description
<code>void Mechal_Reset()</code>	—	—	1.00	Se genera un reseteo mecánico del dispositivo.

Method	Parameters	Return	Version	Description
<code>void Cancel()</code>	—	—	1.00	Borra el código de error o comando que se haya genera sobre el dispositivo.



Method	Parameters	Return	Version	Description
<code>void SensorLevelInformation ()</code>	—	—	1.00	Obtiene el estado físico de todos los sensores que se encuentran en el dispositivo.

Method	Parameters	Return	Version	Description
<code>void CheckConfig()</code>	—	—	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
<code>int search()</code>	<code>byte[], byte[]</code>	<code>int</code>	1.00	Función complementaria que permite encontrar una secuencia de bytes consecutivos en un arreglo.

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

Method	Parameters	Return	Version	Description
<code>byte[] getPositions()</code>	<code>byte[], byte[]</code>	—	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
<code>void open()</code>	—	—	1.00	Establece conexión y configuración del dispositivo

Method	Parameters	Return	Version	Description
<code>void close()</code>	—	—	1.00	Cierra la conexión con el dispositivo.

Method	Parameters	Return	Version	Description
<code>bool isConnection()</code>	—	<code>bool</code>	1.00	Pregunta al dispositivo si se encuentra conectado.

Method	Parameters	Return	Version	Description
<code>void enable()</code>	—	—	1.00	Habilita la aceptación de billetes.

Method	Parameters	Return	Version	Description
<code>void disable()</code>	—	—	1.00	Deshabilita la aceptación de billetes.

Method	Parameters	Return	Version	Description
<code>double[2] getCashDesposite()</code>	—	<code>double[]</code>	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
<code>AcceptorSCAd()</code>	—	—	1.00	Instancia de la clase para el control interno de los eventos del dispositivo.

Method	Parameters	Return	Version	Description
<code>void openConnection()</code>	—	—	1.00	Obtiene el puerto del dispositivo del appConfig y abre la comunicación con PowerUp.A

Method	Parameters	Return	Version	Description
<code>void configDefault()</code>	—	—	1.00	Configura las denominaciones a aceptar.

Event	Parameters	Return	Version	Description
<code>void connectedHandle</code>	<code>object sender, EventArgs e</code>	—	1.00	Cuando el dispositivo recibe el evento de conectado, se cargara 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
<code>void open()</code>	—	—	1.00	Establece conexión y configuración del dispositivo

Method	Parameters	Return	Version	Description
<code>void close()</code>	—	—	1.00	Cierra la conexión con el dispositivo.

Method	Parameters	Return	Version	Description
<code>bool isConnected()</code>	—	<code>bool</code>	1.00	Pregunta al dispositivo si se encuentra conectado.

Method	Parameters	Return	Version	Description
<code>void enable()</code>	—	—	1.00	Habilita el dispositivo para el retiro de monedas.

Method	Parameters	Return	Version	Description
<code>void returnCash()</code>	<code>int[]</code>	—	1.00	Dispensa la cantidad de monedas por denominación.
<p><b>Example:</b></p> <pre>int[3] coinCount = { cantidad de \$1.00 MXN, cantidad de \$5.00 MXN, cantidad de \$10.00 MXN }; name.returnCash(coinCount);</pre> <p>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.</p>				

## 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
<code>void enableContainerCoin()</code>	<code>byte</code>	—	1.00	Habilita los hopper para dispensar.

Method	Parameters	Return	Version	Description
<code>bool getDeviceId()</code>	—	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
<code>byte[] getNumberSerie()</code>	byte	—	1.00	Obtiene el número de serie del dispositivo.

Method	Parameters	Return	Version	Description
<code>void checkStatusSensors()</code>	—	—	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
<code>void open()</code>	—	—	1.00	Establece conexión y configuración del dispositivo

Method	Parameters	Return	Version	Description
<code>void close()</code>	—	—	1.00	Cierra la conexión con el dispositivo.

Method	Parameters	Return	Version	Description
<code>bool isConnection()</code>	—	—	1.00	Pregunta al dispositivo si se encuentra conectado.

Method	Parameters	Return	Version	Description
<code>void enable()</code>	—	—	1.00	Habilita la aceptación de monedas.

Method	Parameters	Return	Version	Description
<code>void disable()</code>	—	—	1.00	Deshabilita la aceptación de monedas.

Method	Parameters	Return	Version	Description
<code>double[] getCashDesposite()</code>	—	<code>double[]</code>	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
<code>bool getDeviceId()</code>	—	<code>bool</code>	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
<code>void setConfigDefaultHoppers()</code>	—	—	1.00	Configura cada denominación a reciclar en un Hopper.

Method	Parameters	Return	Version	Description
<code>void setInibitCoins()</code>	—	—	1.00	Configura la denominación a aceptar.

Method	Parameters	Return	Version	Description
<code>void clearCounterMoney()</code>	—	—	1.00	Limpia el contador interno de monedas del dispositivo.

Method	Parameters	Return	Version	Description
<code>void emptyMoneyBox()</code>	—	—	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
<code>static CCTalk GetInstance()</code>	—	—	1.00	Se obtiene la instancia y si no existe se crea una.

Method	Parameters	Return	Version	Description
<code>SerialPort openConnection()</code>	<code>string</code>	<code>SerialPort</code>	1.00	Se obtiene el COM de cada dispositivo y si no se encuentra abierto se abre de acuerdo con su configuración.

Method	Parameters	Return	Version	Description
<code>void setMessage()</code>	<code>byte[]</code>	—	1.00	Se envía un arreglo de bytes al dispositivo

Method	Parameters	Return	Version	Description
<code>void getMessage()</code>	—	—	1.00	Se recibe una cadena de bytes del dispositivo.

Method	Parameters	Return	Version	Description
<code>bool getDeviceId()</code>	—	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
<code>void setDevices()</code>	—	—	1.00	Clasifica los ID de los dispositivos CCTalk de acuerdo con su nombre y funcionamiento.

Method	Parameters	Return	Version	Description
<code>void sendMessage()</code>	<code>byte[], byte[]</code>	—	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
<code>CCTalk()</code>	—	—	1.00	Se crea una instancia de CCTalk para organizar los dispositivos en un Hashtable.

Method	Parameters	Return	Version	Description
<code>string GetNameDevice()</code>	—	string	1.00	Devuelve el key del dispositivo del App.config

Method	Parameters	Return	Version	Description
<code>void CleanEcho()</code>	—	—	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
<code>int search()</code>	<code>byte[], byte[]</code>	int	1.00	Función complementaria que permite encontrar una secuencia de bytes consecutivos en un arreglo.



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

Method	Parameters	Return	Version	Description
<code>string ByteArrayToString ()</code>	<code>byte[]</code>	<code>string</code>	1.00	Convierte un arreglo de bytes de decimal a hexadecimal

Method	Parameters	Return	Version	Description
<code>byte[] setChecksum()</code>	<code>byte[], byte[]</code>	—	1.00	Calcula el Checksum en módulo 256 para los dispositivos CCTalk