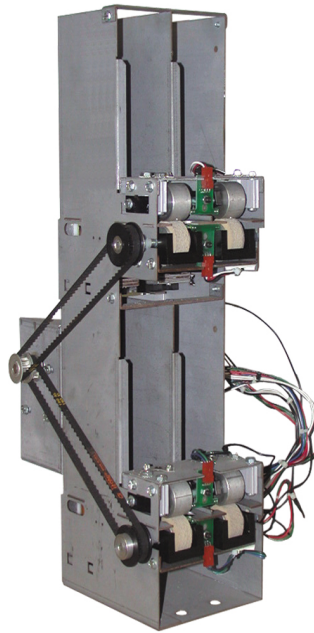


Dispenser.ocx

Activex control for Technik product dispensers and Windows applications



© Technik Mfg Inc, 2005

Table of Contents

Dispenser Control	2
Introduction	4
Introduction	4
Control board	4
Dispensing Mechanism	5
Bill Acceptor	5
Digital Inputs	6
Digital Outputs	6
LCD Display	6
Properties	7
CommPort Property	7
PortOpen Property	7
ColumnStatus Property	7
MechSize Property	8
Success Property	8
Status Property	8
Error Property	9
BAstatus Property	9
BillValue Property	10
GetCredits Property	10
PeekCredits Property	10
ModeSwitchValue Property	11
LowProduct Property	11
SortGateOpen Property	11
ReadDataLength Property	12
ReadData Property	12
Staged Property	12
Methods	14
Reset Method	14
UpdateStatus Method	14
Vend Method	15
Read Method	15
Outstack Method	16
Restack Method	16
MotorOn Method	17
MotorOff Method	17
SwitchEnable Method	18
SwitchDisable Method	18
BillEnable Method	19
BillDisable Method	19
LampOn Method	20
LampOff Method	21
LampMask Method	21
LCDwrite Method	22
LCDwrite2 Method	23
Events	24

Timeout Event	24
PowerFailed Event	24
BillAccepted Event	24
SwitchPress Event	24
SwitchRelease Event	25
ModeChange Event	25
Controller Connections	26
One Column Controller	26
Multi Column Controller	27

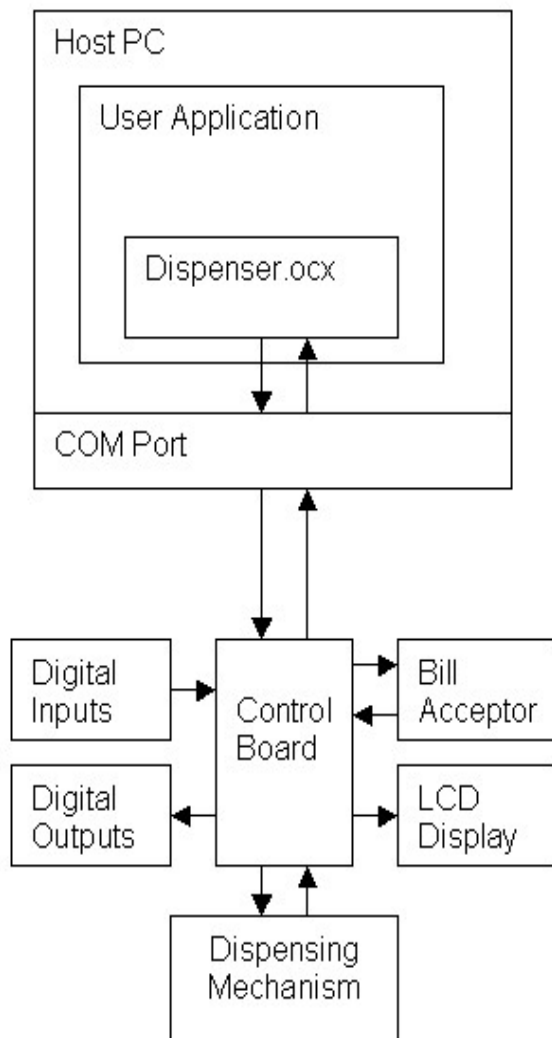
Dispenser Control

Introduction

Introduction

The TMI Dispenser control is an activex COM component that allows Windows based applications to easily communicate with a product dispenser. Basic dispensers with one to eight product columns are supported. The controller incorporated in the dispenser also provides the ability to interface to other devices such as bill acceptors, lamps, and switches.

The Dispenser control communicates with the dispenser controller through a COM port on the host PC. The ocx handles all the communication functions and protocol.



Control Board

The control board and Dispenser.ocx do more than just make it easy to interface a product dispenser to a personal computer. Support is provided for a bill acceptor, LCD display, and up to 20 lamps and switches. All the functions of a full featured vending machine can be controlled using a single COM port.

Properties

CommPort

PortOpen

Methods

Reset

UpdateStatus

Events

Timeout

PowerFailed

Dispensing Mechanism

Technik can supply dispensing mechanisms for a wide variety of products including cards of all sizes, boxes, CDs and DVDs. These range from single product dispensers to dispensers that can dispense up to eight different products.

Some mechanisms include the ability to read and dispense products that incorporate bar code or magnetic stripe information.

The Dispenser.ocx control provides a common set of facilities for using these dispensers

Properties

MechSize

Success

Status

Error

LowProduct (option)

SortGateOpen (option)

ReadDataLength (Mag Stripe or Bar Code)

ReadData (Mag Stripe or Bar Code)

Methods

Vend

MotorOn

MotorOff

Read

Outstack

Restack

Bill Acceptor

Using a Technik dispenser and the Dispenser.ocx control provides a simple way to interface a pulse mode bill acceptor to a PC.

Properties

BAstatus
BillValue
GetCredits
PeekCredits

Methods

BillEnable
BillDisable

Events

BillAccepted

Digital Inputs

Up to twelve digital inputs are available to the application, depending on the control board model. These are typically used for mechanical switches such as push buttons, but can be freely used with any 5V compatible signal.

Properties

ModeSwitchValue

Methods

SwitchEnable
SwitchDisable

Events

SwitchPress
SwitchRelease
ModeChange

Digital Outputs

Four or eight digital outputs are available to the application, depending on the control board model. These are typically used for incandescent lamps or LEDs, but can be used to drive any device within the power limits of the output.

Methods

LampOn
LampOff
LampMask

LCD Display

Messages can be displayed using a two line by sixteen character LCD display.

Methods

LCDwrite
LCDwrite2

Properties

CommPort Property

Sets and returns the communications port number.

Syntax

Dispenser.CommPort[= *value*]

Where *value* is an integer specifying the port number.

Data Type
Integer

PortOpen Property

Sets and returns the state of the communications port.

Syntax

Dispenser.PortOpen[= *value*]

Where *value* is a boolean expression specifying the state of the communications port.

Data Type
Boolean

ColumnStatus Property

Read Only Returns an integer value reflecting the status of the selected product column.

Syntax

Dispenser.ColumnStatus[(*column#*)]

column# is an optional parameter specifying the column whose status is to be returned.
Default is 1.

Return values are:

-1	Column does not exist.
0	OK. Column ready.
1	Sold Out. Only on dispensers with sold out sensors.
2	Down. Failed to dispense a product.

This value is updated after each command is completed. To make sure that the column is ready, refresh this property using the UpdateStatus method.

For one column dispensers, the same information can be obtained from the Status property.

Data Type
Integer

MechSize Property

Sets and returns the number of product columns in the mechanism.

Syntax

Dispenser.MechSize[= *value*]

Where *value* is an integer expression specifying the number of columns. This property defaults to one when the control is loaded. It must be set by the application to support mechanisms with more than one column.

Data Type
Integer

Success Property

Read Only Returns a boolean expression showing whether the last operation completed successfully.

Syntax

Dispenser.Success

Returns a boolean expression specifying the completion status of the last operation. True equals successful completion. Note that this value can be true, but there could also be a condition that would prevent some future operation from completing. See the Status property.

Data Type
Boolean

Status Property

Read Only Returns an integer value representing the state of the dispenser.

Syntax

Dispenser.Status

Return values are:

0	OK. Dispenser is ready.
1	Sold Out. No product is available in any column
2	Down. No column is able to dispense.

>2	Fatal error. The value represents the error	
	13	Product jam.
	14	Product sensed in wrong column (for multi-column dispensers). Can also be caused by inserting an object through the front of the dispenser in an attempt to steal product.

This value is updated after each command is completed. To make sure that the dispenser is ready, refresh this property using the UpdateStatus method.

Data Type
Integer

Error Property

Read Only Returns an integer value showing the completion status of the last operation.

Syntax

Dispenser.Success

Values of the Error property are:

0	Success
1	Failure. Check Status property for more information.
2	Command mismatch. This can occur if a previous command timed-out, but completed after a new command was issued.
3	Time out. The Timeout event will also occur prior to this status being returned.

This is the same value that is returned by each of the Dispenser methods.

Data Type
Integer

BAstatus Property

Read Only Returns an integer value showing the status of the Bill Acceptor.

Syntax

Dispenser.BAstatus

Values of the BAstatus property are:

0	Up
2	Down. (The bill acceptor has detected a problem)
11	Short Pulse. A pulse received from the bill acceptor was out of tolerance.
12	Long Pulse. A pulse received from the bill acceptor was out of tolerance.

The timing of the pulses received on the bill acceptor input is monitored by the controller for unexpectedly long or short pulse widths. If a pulse that is outside acceptable limits is found, the status will be 12 (long pulse) or 11 (short pulse). These errors can be the result of electronic attempts to defraud the machine or some other electrical problem. In any case these are regarded as non-recoverable errors and will remain asserted until the controller is reset.

The bill acceptor itself may detect an error condition. These vary by manufacturer and can range from a full bill stacker to electronic failure. Any of these will cause the bill acceptor status to be set to 2 (Down). Once the problem is corrected the status will return to 0 (Up). **IMPORTANT:** The status message will only show Down status while the bill acceptor is enabled (see the BillEnable method).

Data Type
Integer

BillValue Property

Sets and returns value to be assigned to each pulse from the bill acceptor.

Syntax

```
Dispenser.BillValue[ = value ]
```

The bill acceptor issues a string of pulses each time a bill is deposited. The number of pulses indicates the value of the bill. On most U.S. currency bill acceptors, one pulse equals one dollar. The number of pulses is multiplied by the BillValue property to arrive at the value of the bill. The default for the BillValue property is 100 (100 pennies).

Data Type
Long integer

GetCredits Property

Read Only Returns a long integer containing the number of credits that have been accumulated by the control. This value is reset to zero after each time it is read.

Syntax

```
Dispenser.GetCredits
```

As bills are deposited in the bill acceptor, the pulses sent by the bill acceptor are multiplied by the BillValue property and accumulated by the control. This property is a long integer containing the credits accumulated since the last time the GetCredits property was examined. Each time this property is read, the accumulated credits are reset to zero.

The value of this property and the PeekCredits property is the same. The only difference is that reading this value using PeekCredits does not reset it to zero.

Data Type
Long integer

PeekCredits Property

Read Only Returns a long integer containing the number of credits accumulated by the control.

Syntax

Dispenser.PeekCredits

As bills are deposited in the bill acceptor, the pulses sent by the bill acceptor are multiplied by the BillValue property and accumulated by the control. This property is a long integer containing the credits accumulated since the last time the GetCredits property was examined.

Data Type
Long integer

ModeSwitchValue Property

Read Only Returns an integer containing the value of the "mode" switch inputs.

Syntax

Dispenser.ModeSwitchValue

There are four mode input pins that are combined to form a value in the range 0 through 15. Refer to the Controller Connections section for more information.

See also:
ModeChange event.
One Column Controller
Multi Column Controller

Data Type
Integer

LowProduct Property

Read Only Returns a boolean expression expressing whether the product stack has fallen below the low product sensor.

Syntax

Dispenser.LowProduct

This property is set to true when the product stack is below the low product sensor and false when the product stack is blocking the sensor.

NOTE: The low product sensor is an option on the 110-022 controller only. If it is not installed this property will always be False.

Data Type

Boolean

SortGateOpen Property

Read Only Returns a boolean expression indicating whether the sort gate is open. Only available on dispensers equipped with a sorter.

Syntax

Dispenser.SortGateOpen

The value of this property will be true is the gate is open. This will only occur as the result of a product becoming lodged in the sorter or a mechanical failure. It should always be regarded as an error.

Data Type

Boolean

ReadDataLength Property

Read Only Returns the number of characters read from the magnetic stripe or bar code reader.

Syntax

Dispenser.ReadDataLength

This value is the number of characters in the ReadData property.

Data Type

Integer

ReadData Property

Read Only Contains the data read from the magnetic stripe or bar code.

Syntax

Dispenser.ReadData

Updated after each CardRead operation. The number of characters in the string can be retrieved from the ReadDataLength property.

Data Type

String

Staged Property (mag stripe reader only)

Read Only Returns a boolean expression of whether a card has been advanced to the 'staged' position.

Syntax

Dispenser.Staged

After invoking the Read method, a card is moved from the bottom of the stack, past the read heads and stops at the staged position. From here it can be dispensed using the Vend or Outstack methods or returned to the stack with the Restack method. While there is a card in this position, the value of this property will be True. If the Read method is invoked while the card is in the staged position, it will be restacked and then read again.

Data Type

Boolean

Methods

Reset Method

Resets the dispenser controller to it's initial state.

Syntax

Dispenser.Reset[*Timeout*]

Timeout is an optional long integer that specifies the time in milliseconds to wait for the command to complete. Default is 10000ms (10 seconds)

Return

An integer value is returned indicating the success or failure of the operation:

0	Success
1	Failure (should not occur)
2	Command mismatch. This can occur if a previous command timed-out, but completed after a new command was issued.
3	Time out. The Timeout event will also occur prior to this status being returned.

Following this, the following properties will return to their default states:

Status	0 (or 1 if sold out)
BAstatus	0
ColStat	0 (or 1 if sold out)

Also, switches will be enabled and bill acceptor will be disabled (see the SwitchEnable, SwitchDisable, BillEnable and BillDisable methods).

UpdateStatus Method

Refreshes all properties to reflect the current dispenser status.

Syntax

Dispenser.UpdateStatus[*Timeout*]

Timeout is an optional long integer that specifies the time in milliseconds to wait for the command to complete. Default is 1000ms (1 seconds)

Return

An integer value is returned indicating the success or failure of the operation:

0	Success
1	Failure (should not occur)
2	Command mismatch. This can occur if a previous command timed-out, but completed after a new command was issued.

3	Time out. The Timeout event will also occur prior to this status being returned.
---	--

This method should be invoked periodically while the application is idle to detect changes in dispenser readiness.

Vend Method

Dispense a product.

Syntax

Dispenser.Vend[*Column*, *Timeout*]

Column is an optional integer value specifying the column to dispense from. Valid values are 0 through n where n equals the MechSize property. If 0 is used, the vend will be from the first available (not down or sold-out) column. Default is 0.

Timeout is an optional long integer that specifies the time in milliseconds to wait for the command to complete. Default is 9000ms (9 seconds)

Return

An integer value is returned indicating the success or failure of the operation:

0	Success
1	Failure. Check Status property for more information.
2	Command mismatch. This can occur if a previous command timed-out, but completed after a new command was issued.
3	Time out. The Timeout event will also occur prior to this status being returned.

On the magnetic stripe dispenser, the card must first be in the Staged position.

See also: MotorOff Method

Read Method

Reads the bar code or magnetic stripe (reader dispensers only).

Syntax

Dispenser.Read[*Timeout*]

Timeout is an optional long integer that specifies the time in milliseconds to wait for the command to complete. Default is 9000ms (9 seconds)

Return

An integer value is returned indicating the success or failure of the operation:

0	Success
1	Failure (should not occur)

2	Command mismatch. This can occur if a previous command timed-out, but completed after a new command was issued.
3	Time out. The Timeout event will also occur prior to this status being returned.

This causes the information to be read, but does not dispense the product. Use the Vend or Outstack method to complete the dispense.

See also:

ReadData property

ReadDataLength property

Data Type

Integer

Outstack Method

Causes the product to be delivered to the alternate chute on dispensers so equipped. This can be useful for discarding unreadable product.

Syntax

Dispenser.Outstack[*Timeout*]

Timeout is an optional long integer that specifies the time in milliseconds to wait for the command to complete. Default is 9000ms (9 seconds)

Return

An integer value is returned indicating the success or failure of the operation:

0	Success
1	Failure (should not occur)
2	Command mismatch. This can occur if a previous command timed-out, but completed after a new command was issued.
3	Time out. The Timeout event will also occur prior to this status being returned.

On the magnetic stripe dispenser, the card must first be in the Staged position.

See also:

Vend Method

Data Type

Integer

Restack Method (mag stripe reader only)

Returns a card from the Staged position to the product stack.

Syntax

Dispenser.Restack[*Timeout*]

Timeout is an optional long integer that specifies the time in milliseconds to wait for the command to complete. Default is 9000ms (9 seconds)

Return

An integer value is returned indicating the success or failure of the operation:

0	Success
1	Failure (should not occur)
2	Command mismatch. This can occur if a previous command timed-out, but completed after a new command was issued.
3	Time out. The Timeout event will also occur prior to this status being returned.

See also:

Staged Property

Vend Method

Outstack Method

Data Type

Integer

MotorOn Method

Runs the dispenser motor for maintenance purposes.

Syntax

Dispenser.MotorOn[*Timeout*]

Timeout is an optional long integer that specifies the time in milliseconds to wait for the command to complete. Default is 2000ms (2 seconds)

Return

An integer value is returned indicating the success or failure of the operation:

0	Success
1	Failure (should not occur)
2	Command mismatch. This can occur if a previous command timed-out, but completed after a new command was issued.
3	Time out. The Timeout event will also occur prior to this status being returned.

See also: MotorOff Method

MotorOff Method

Runs the dispenser motor for maintenance purposes.

Syntax

Dispenser.MotorOff[*Timeout*]

Timeout is an optional long integer that specifies the time in milliseconds to wait for the command to complete. Default is 2000ms (2 seconds)

Return

An integer value is returned indicating the success or failure of the operation:

0	Success
1	Failure (should not occur)
2	Command mismatch. This can occur if a previous command timed-out, but completed after a new command was issued.
3	Time out. The Timeout event will also occur prior to this status being returned.

See also: MotorOn Method

SwitchEnable Method

Allows the SwitchPress and SwitchRelease events to occur.

Syntax

Dispenser.SwitchEnable[*Timeout*]

Timeout is an optional long integer that specifies the time in milliseconds to wait for the command to complete. Default is 1000ms (1 second)

Return

An integer value is returned indicating the success or failure of the operation:

0	Success
1	Failure (should not occur)
2	Command mismatch. This can occur if a previous command timed-out, but completed after a new command was issued.
3	Time out. The Timeout event will also occur prior to this status being returned.

See also:

SwitchDisable Method

SwitchPress Event

SwitchRelease Event

SwitchDisable Method

Prevents the SwitchPress and SwitchRelease events from occurring.

Syntax

Dispenser.SwitchDisable[*Timeout*]

Timeout is an optional long integer that specifies the time in milliseconds to wait for the command to complete. Default is 1000ms (1 second)

Return

An integer value is returned indicating the success or failure of the operation:

0	Success
1	Failure (should not occur)
2	Command mismatch. This can occur if a previous command timed-out, but completed after a new command was issued.
3	Time out. The Timeout event will also occur prior to this status being returned.

See also:

SwitchEnable Method
SwitchPress Event
SwitchRelease Event

BillEnable Method

Allows the Bill Acceptor to accept cash.

Syntax

Dispenser.BillEnable[*Timeout*]

Timeout is an optional long integer that specifies the time in milliseconds to wait for the command to complete. Default is 1000ms (1 second)

Return

An integer value is returned indicating the success or failure of the operation:

0	Success
1	Failure (should not occur)
2	Command mismatch. This can occur if a previous command timed-out, but completed after a new command was issued.
3	Time out. The Timeout event will also occur prior to this status being returned.

Under certain conditions, such as when the dispenser is out of product, it is important to prevent the bill acceptor from accepting money. This method controls the bill acceptor's enable input to allow bills to be accepted.

The bill acceptor is disabled when the dispenser is first powered on and also following a reset.

See also:

BillDisable Method
BillAccepted Event

BillDisable Method

Prevents the Bill Acceptor from accepting cash.

Syntax

Dispenser.BillDisable[*Timeout*]

Timeout is an optional long integer that specifies the time in milliseconds to wait for the command to complete. Default is 1000ms (1 second)

Return

An integer value is returned indicating the success or failure of the operation:

0	Success
1	Failure (should not occur)
2	Command mismatch. This can occur if a previous command timed-out, but completed after a new command was issued.
3	Time out. The Timeout event will also occur prior to this status being returned.

Under certain conditions, such as when the dispenser is out of product, it is important to prevent the bill acceptor from accepting money. This method controls the bill acceptor's enable input to prevent bills from being accepted.

The bill acceptor is disabled when the dispenser is first powered on and also following a reset.

See also:

BillEnable Method

BillAccepted Event

LampOn Method

Turns on one lamp.

Syntax

Dispenser.LampOn[*Lamp Number*, *Timeout*]

Lamp Number is an integer containing the number of the lamp to turn on. Possible values are 1-4 for the one column controller and 1-8 for the multi column controller.

Timeout is an optional long integer that specifies the time in milliseconds to wait for the command to complete. Default is 1000ms (1 second)

Return

An integer value is returned indicating the success or failure of the operation:

0	Success
1	Failure (should not occur)
2	Command mismatch. This can occur if a previous command timed-out, but completed after a new command was issued.
3	Time out. The Timeout event will also occur prior to this status being returned.

This method turns on one medium current output at a time. These are commonly used to drive incandescent lamps or LEDs, hence the name of the method. These same outputs could easily be used for other things requiring a digital output.

All lamps are turned off when the dispenser is first powered on and also following a reset.

See also:

LampOff Method

LampMask Method

One Column Controller

Multi Column Controller

LampOff Method

Turns off one lamp.

Syntax

Dispenser.LampOff[*Lamp Number, Timeout*]

Lamp Number is an integer containing the number of the lamp to turn off. Possible values are 1-4 for the one column controller and 1-8 for the multi column controller.

Timeout is an optional long integer that specifies the time in milliseconds to wait for the command to complete. Default is 1000ms (1 second)

Return

An integer value is returned indicating the success or failure of the operation:

0	Success
1	Failure (should not occur)
2	Command mismatch. This can occur if a previous command timed-out, but completed after a new command was issued.
3	Time out. The Timeout event will also occur prior to this status being returned.

This method turns of one medium current output at a time. These are commonly used to drive incandescent lamps or LEDs, hence the name of the method. These same outputs could easily be used for other things requiring a digital output.

All lamps are turned off when the dispenser is first powered on and also following a reset.

See also:

LampOn Method

LampMask Method

One Column Controller

Multi Column Controller

LampMask Method

Turns on/off all lamps to match a bit pattern.

Syntax

Dispenser.LampMask[*Lamp Pattern*, *Timeout*]

Lamp Pattern is an integer containing a bit mask describing which lamps are to be on or off. Possible values are 0-15 for the one column controller and 0-255 for the multi column controller.

Timeout is an optional long integer that specifies the time in milliseconds to wait for the command to complete. Default is 1000ms (1 second)

Return

An integer value is returned indicating the success or failure of the operation:

0	Success
1	Failure (should not occur)
2	Command mismatch. This can occur if a previous command timed-out, but completed after a new command was issued.
3	Time out. The Timeout event will also occur prior to this status being returned.

The lower bits of the *Lamp Pattern* correspond to the digital outputs.

Bit 0 = lamp 1
 Bit 1 = lamp 2
 etc..

A one bit turns the lamp on, a zero turns it off. For example:

0 = All lamps off
 255 = All lamps on
 5 = Lamps 0 and 2 on, all others off.

All lamps are turned off when the dispenser is first powered on and also following a reset.

See also:

- LampOn Method
- LampOff Method
- One Column Controller
- Multi Column Controller

LCDwrite Method

Shows a message on the 2x16 LCD display starting in the first position on line 1.

Syntax

Dispenser.LCDwrite[*Message*, *Timeout*]

Message is a string of printable characters. The first 16 characters will display on the first line of the display. Remaining characters will display on the second line.

Timeout is an optional long integer that specifies the time in milliseconds to wait for the command to complete. Default is 1000ms (1 second)

Return

An integer value is returned indicating the success or failure of the operation:

0	Success
1	Failure (should not occur)
2	Command mismatch. This can occur if a previous command timed-out, but completed after a new command was issued.
3	Time out. The Timeout event will also occur prior to this status being returned.

This method does not clear the display so 16 or 32 character messages should be used if one or two lines are to be completely replaced.

See also:
LCDwrite2 Method

LCDwrite2 Method

Shows a message on the 2x16 LCD display starting in the first position on line 2.

Syntax

Dispenser.LCDwrite2[*Message*, *Timeout*]

Message is a string of printable characters. The first 16 characters will display on the second line of the display. Any remaining characters will be ignored.

Timeout is an optional long integer that specifies the time in milliseconds to wait for the command to complete. Default is 1000ms (1 second)

Return

An integer value is returned indicating the success or failure of the operation:

0	Success
1	Failure (should not occur)
2	Command mismatch. This can occur if a previous command timed-out, but completed after a new command was issued.
3	Time out. The Timeout event will also occur prior to this status being returned.

This method does not clear the display so 16 character messages should be used if the line is to be completely replaced.

See also:
LCDwrite Method

Events

Timeout Event

Occurs when the dispenser doesn't respond to a command before the timeout expires.

Syntax

Dispenser.Timeout(*Timeout*)

Timeout is a long integer that specifies the time in milliseconds that was exceeded.

PowerFailed Event

Occurs when the dispenser resets. This can be caused by the power to the dispenser falling out of tolerance or by pressing the manual reset switch. It is not fired by the Reset method.

Syntax

Dispenser.PowerFailed()

BillAccepted Event

Occurs when money is deposited in the bill acceptor.

Syntax

Dispenser.BillAccepted(*Bill Value*)

Bill Value is a long integer that represents the number of pulses received from the bill acceptor. On most U.S. currency bill acceptors, one pulse equals one dollar. The number of pulses is multiplied by the BillValue property to arrive at the value of the bill. The default for the BillValue property is 100 (100 pennies) so a *Bill Value* of 2000 would indicate that a \$20 bill was accepted.

See also:

- GetCredits property
- PeekCredits property
- BillValue property
- BillEnable Method

SwitchPress Event

Occurs when a momentary switch connected to the controller is closed (connected to ground).

Syntax

Dispenser.SwitchPress(*Switch Number*)

Switch Number is an integer that specifies the switch that was pressed. Possible values are 1 to 12. On the one column controller (PN: 110-022) switches 5 through 8 are unavailable.

Refer to the Controller Connections section for more information.

See also:

One Column Controller
Multi Column Controller

SwitchRelease Event

Occurs when a momentary switch connected to the controller is opened.

Syntax

Dispenser.SwitchRelease(*Switch Number*)

Switch Number is an integer that specifies the switch that was released. Possible values are 1 to 12. On the one column controller (PN: 110-022) switches 5 through 8 are unavailable. Refer to the Controller Connections section for more information.

See also:

One Column Controller
Multi Column Controller

ModeChange Event

Occurs when the value of the "mode" inputs change.

Syntax

Dispenser.ModeChange(*Value*)

Value is an integer that contains the decimal value of the mode inputs. There are four mode input pins that are combined to form a value in the range 0 through 15. Refer to the Controller Connections section for more information.

See also:

One Column Controller
Multi Column Controller

Controller Connections

I/O Connectors on the one column controller

J10 "Front Panel"

Pin #	Type	Name	Description
1	Power	Gnd	Logic ground
2	Input	Switch 1	Triggers SwitchPressed event when connected to ground.
3	Input	Switch 2	Triggers SwitchPressed event when connected to ground.
4	Output	Lamp 1	Sinks up to 500mA @ 50V*.
5	Output	Lamp 2	Sinks up to 500mA @ 50V*.
6	Output	Lamp 3	Sinks up to 500mA @ 50V*.
7	Output	Lamp 4	Sinks up to 500mA @ 50V*.
8	Power	+15V	Power output.

* Total power of all outputs should be limited to 1.0W

J12 "Bill Acceptor"

Pin #	Type	Name	Description
1	Input	Billin	Bill acceptor bill pulse input
2	Power	Gnd	Logic ground
3	Input	Coinin	Coin acceptor pulse input
4	Power	Gnd	Logic ground
5	Output	Benable	Bill acceptor enable
6	Power	+5V	Power output.
7	Input	Billerr	Bill acceptor error signal
8	Power	+15V	Power output.

No isolation is provided on this connector

J23 "Aux Controls"

Pin #	Type	Name	Description
1	Power	Gnd	Logic ground
2	Input	MD8	'Mode switch' bit 3
3	Input	MD4	'Mode switch' bit 2
4	Input	MD2	'Mode switch' bit 1
5	Input	MD1	'Mode switch' bit 0
6	Input	Switch 9	Triggers SwitchPressed event when connected to ground.
7	Input	Switch 10	Triggers SwitchPressed event when connected to ground.
8	Input	Switch 11	Triggers SwitchPressed event when connected to ground.

P1 "Option Jumpers"

Position	Type	Name	Description
A	Input	Switch 1	Triggers SwitchPressed event when closed.
B	Input	Switch 2	Triggers SwitchPressed event when closed.
C	Input	Switch 3	Triggers SwitchPressed event when closed.
D	Input	Switch 4	Triggers SwitchPressed event when closed.
E	Input	Switch 9	Triggers SwitchPressed event when closed.

Push buttons

ID	Label	Description
S1	Reset	Hard reset
S2	A	Switch 12 SwitchPressed event.
S3	B	Manual test feed. Only works if serial connection is open.

I/O Connectors on the multi column controller

Connector J4

Pin #	Type	Name	Description
1	Output	Lamp 1	Sinks up to 500mA @ 50V*.
2	Output	Lamp 2	Sinks up to 500mA @ 50V*.
3	Output	Lamp 3	Sinks up to 500mA @ 50V*.
4	Output	Lamp 4	Sinks up to 500mA @ 50V*.
5		Not used	Reserved
6	Power	+12V	Power Output

* Total power of all outputs on this connector should be limited to 1.0W

Connector JB4

Pin #	Type	Name	Description
1	Output	Lamp 5	Sinks up to 500mA @ 50V*.
2	Output	Lamp 6	Sinks up to 500mA @ 50V*.
3	Output	Lamp 7	Sinks up to 500mA @ 50V*.
4	Output	Lamp 8	Sinks up to 500mA @ 50V*.
5		Not used	
6	Power	+12V	Power Output

* Total power of all outputs on this connector should be limited to 1.0W

Connector J6

Pin #	Type	Name	Description
1	Input	Switch 3	Triggers SwitchPressed event when connected to ground.

2	Input	Switch 4	Triggers SwitchPressed event when connected to ground.
3	Input	Switch 1	Triggers SwitchPressed event when connected to ground.
4	Input	Switch 2	Triggers SwitchPressed event when connected to ground.
5		Not used	Reserved
6	Power	Gnd	Ground return

Connector JB6

Pin #	Type	Name	Description
1	Input	Switch 7	Triggers SwitchPressed event when connected to ground.
2	Input	Switch 8	Triggers SwitchPressed event when connected to ground.
3	Input	Switch 5	Triggers SwitchPressed event when connected to ground.
4	Input	Switch 6	Triggers SwitchPressed event when connected to ground.
5		Not used	Reserved
6	Power	Gnd	Ground return

Connector JB8

Pin #	Type	Name	Description
1	Power	Gnd	Ground return
2	Input	MD8	'Mode switch' bit 3
3	Input	MD4	'Mode switch' bit 2
4	Input	MD2	'Mode switch' bit 1
5	Input	MD1	'Mode switch' bit 0
6		Not used	Ground return
7		Not used	
8		Not used	Reserved

Connector J7

Pin #	Type	Name	Description
1	Input	Billin	Bill acceptor bill pulse input
2	Power	Gnd	Logic ground
3	Input	Coinin	Coin acceptor pulse input
4	Power	Enable -	Connect to ground at bill acceptor
5	Output	Enable +	Bill acceptor enable
6	Power	Error +	Connect to voltage at bill acceptor
7	Input	Error -	Bill acceptor error signal
8	Power	+15V	Power output.

Pins 4&5 and 6&7 are optically isolated

