Copyright Todos los derechos reservados ThinkChip

Contenido

Manual de referencia para comunicación RS232

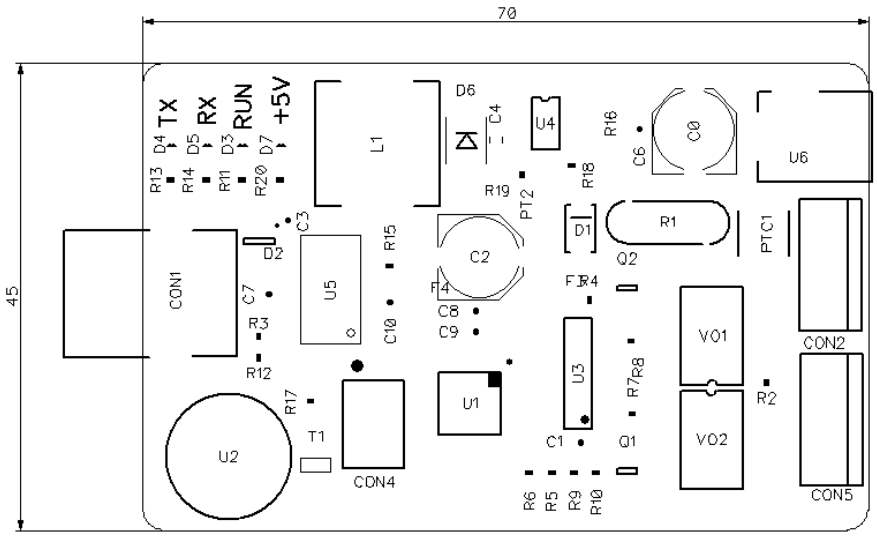
Especificaciones técnicas

A continuación, especificamos las características técnicas del convertidor USB RS232 MDB

Especificaciones eléctricas

- Alimentación eléctrica de 24VDC hasta 36VDC 2amp
- 2 puertos independientes MDB
- Conexión RS232
- Conexión USB
- Botón de reset
- Led indicador de alimentación eléctrica
- 3 led indicadores de funcionamiento

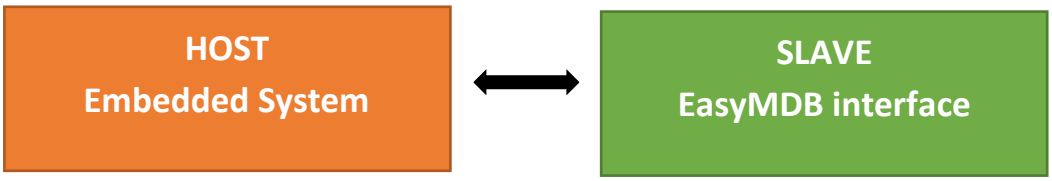
Especificaciones físicas



CON1 = USB
CON5 = MDB Master
CON2 = MDB Slave

Introducción protocolo de comunicación

La comunicación entre el dispositivo y el maestro (PC o sistema embebido) es mediante comunicación asíncrona. Siendo el maestro el que establezca siempre la comunicación con la interfaz.



Comandos Coin Changer

La interfaz tiene los siguientes comandos disponibles al maestro

- Init
- Setup
- Poll
- Tubes
- PayOut
- Coin Accepted
- Errors
- Dispense Coin
- Identification

Los pasos para iniciar la comunicación con el Coin Changer son:

1. Init – Inicializa el coin changer y la secuencia, responde si se ha establecido correctamente la comunicación con el dispositivo.
2. Setup – Se recepciona información importante acerca del dispositivo.
3. Identification – Información acerca de número de serie, modelo, etc.
4. Tubes – Estado de tubos de moneda
5. Coin accepted – Habilita o deshabilita aceptación de monedas
6. Poll – Estado actual de dispositivo, con este comando se tiene que realizar cada 1 a 4 segundos

Comandos para aceptador de billetes y/o reciclador

Las funciones disponibles para el aceptador de billetes y reciclador de billetes son

- Init
- Setup
- Identification
- Poll
- Bill Accepted
- Bill Stacker
- Diagnostic
- Security
- Escrow
- Recycler Payout
- Recycler Setup
- Recycler enable
- Recycler Dispense Status

Los pasos para iniciar la comunicación con el aceptador de billetes y/o reciclador son:

1. Init – Inicializa el dispositivo y la secuencia, responde si se ha establecido correctamente la comunicación con el dispositivo.
2. Setup – Se recepciona información importante acerca del dispositivo.
3. Identification – Información acerca de número de serie, modelo, etc.
4. Bill accepted – Habilita o deshabilita aceptación de monedas
5. Security
6. Poll – Estado actual de dispositivo, con este comando se tiene que realizar cada 1 a 4 segundos

Copyright Todos los derechos reservados ThinkChip

Formato de comunicación

Parameter	Description
Baud Rate	115200 bauds
Data bits	8 bits
Start Bits	1 bit
Stop Bits	1 bit
Parity	None

Link Layer

El protocol de comunicación es un paquete de información orientado, toda la información intercambiada entre los dos dispositivos será basada en este formato de paquete.

El paquete de información comienza con un carácter de control, existen 4 tipos de control: COIN, BILL, CASH y SYSTEM (0x02, 0x03, 0x04 y 0x01 valores hexadecimales respectivamente) y terminan con un carácter que indica final de carrera 0xff, el cual viene seguido de un valor de verficación de bytes llamado checksum 8 bit.

Formato de datos

Existen dos tipos de paquetes de datos, mensaje comando es el paquete que se envía del maestro al esclavo y el mensaje respuesta es la respuesta del esclavo ante la petición de información del maestro.

Packet format for Command Message (HOST to SLAVE)

Device	Command	Data	FC (Final character)	Checksum BCC 8 bits
SYSTEM=0x01 COIN=0x02 BILL=0x03 CASHLESS=0x04	These describe the action that will be realized. For example COIN_INIT=0x01	The Data Field is a stream of data with variable length, which depends on the Command word. There are also some COMMANDs have zero length of data field.	FC=0xff	Checksum is the last character to be sent, is calculated by adding all character, the checksum is not included in the summation.

Packet format for Reply Message (SLAVE to HOST)

Device	Command	Data	FC (Final character)	Checksum BCC 8 bits
SYSTEM=0x01 COIN=0x02 BILL=0x03 CASHLESS=0x04	These describe the action order	The Data Field is a stream of data with variable length, which depends on the Command word. There are also some	FC= 0xff	Checksum is the last character to be sent, is calculated by adding all character, the checksum is not

		COMMANDs have zero length of data field.		included in the summation.

Comandos para dispositivos de cobro

Coin changer comandos

Init

Inicia la comunicación con el aceptador cambiador de monedas, y la secuencia sugerida por NAMA para inicializar dispositivos MDB.

Send data

Buffer Output	Packet Format	Description
Buffer Output[0]=0x02	**Device**	**Coin changer**
Buffer Output[1]=0x01	**Command**	**Init coin changer**
Buffer Output[2]=FC	**FC**	**FC=0xff**
Buffer Output[3]=CHK	**CHECKSUM**	

Reception data

Buffer Input	Packet Format	Description
Buffer Input[0]=0x02	**Device**	**Coin changer**
Buffer Input[1]=0x01	**Command**	**Init coin changer**
Buffer Input[2]	**Data**	**Response** 0x01 OK 0x00 FAIL

Buffer Input[3]=FC	**FC**	**FC=0xff**
Buffer Input[4]=CHK	**CHECKSUM**	

Setup

Send data

Buffer Output	Packet Format	Description
Buffer Output[0]=0x02	**Device**	**Coin Changer**
Buffer Output[1]=0x0c	**Command**	**Setup command**
Buffer Output[2]=FC	**FC**	**FC=0xff**
Buffer Output[3]=CHK	**CHECKSUM**	

Reception data

Buffer Input	Packet Format	Description
Buffer Input[0]=0x02	**Device**	**Coin Changer**
Buffer Input[1]=0x0c	**Command**	**Setup**
Buffer Input[2]	**Data**	**Level** Changer Feature Level - 1 byte Indicates the feature level of the changer. This will distinguish the changers feature level to the VMC. Current defined levels: Level 2: Supports “core” command set. These are: RESET, STATUS, TUBE STATUS, POLL, COIN TYPE, and DISPENSE.

		Level 3: Supports level two and the EXPANSION command addition changer model number, manufacturer code, turning revision, etc.
Buffer Input[3] to **Buffer Input[4]**	**Data**	Country / Currency Code - 2 bytes The packed BCD country / currency code of the changer can be sent in two different forms depending on the value of the left most BCD digit. If the left most digit is a 0, the International Telephone Code is used to indicate the country that the changer is set-up for. For example, the USA code is 00 01H (Z2 = 00 and Z3 = 01). If the left most digit is a 1, the latest version of the ISO 4217 numeric currency code is used (see Appendix A1). For example, the code for the US dollar is 18 40H (Z2 = 18 and Z3 = 40) and for the Euro is 1978 (Z2 = 19 and Z3 = 78).
Buffer Input[5]	**Data**	Coin Scaling Factor - 1 byte All accepted coin values must be evenly divisible by this number. For example, this could be set to 05H for the USA nickel.
Buffer Input[6]	**Data**	Decimal Places - 1 byte Indicates the number of decimal places on a credit display. For example, this could be set to 02H in the USA.
Buffer Input[7] to [8]	**Data**	Coin Type Routing - 2 bytes Indicates what coin types can be routed to the Changer's tubes. b15 b14 b13 b12 b11 b10 b9 b8 | b7 b6 b5 b4 b3 b2 b1 b0

		Z6 Z7 Bit is set to indicate a coin type can be routed to the tube. Valid coin types are 0 to 15.
Buffer Input[9] to [24]	**Data**	Coin Type Credit - 16 bytes Indicates the value of coin types 0 to 15. Values must be sent in ascending order. This number is the coin's monetary value divided by the coin scaling factor. Unused coin types are sent as 00H. Unsent coin types are assumed to be zero. It is not necessary to send all coin types. Coin type credits sent as FFH are assumed to be vend tokens. That is, their value is assumed to worth one vend. The bytes position in the 16 byte string indicates the coin type(s). For example, the first byte sent would indicate the value of coin type 0, the second byte sent would indicate the value of coin type 1, and so on. For example, the USA coin types may be; Coin type 0 = nickel, Coin type 1 = dime, Coin type 2 = quarter, Coin type 3 = dollar.
Buffer Input[25]=FC	**FC**	**FC=0xff**
Buffer Input[26]=CHK	**CHECKSUM**	

Coin Poll

Send data

Buffer Output	Packet Format	Description
Buffer Output[0]=0x02	**Device**	**Coin Changer**
Buffer Output[1]=0x03	**Command**	**Poll Coin Changer**
Buffer Output[2]=FC	**FC**	**FC=0xff**

Buffer Output[3]=CHK	**CHECKSUM**	

Reception data

Buffer Input	**Packet Format**	**Description**
Buffer Input[0]=0x02	**Device**	**Coin Changer**
Buffer Input[1]=0x03	**Command**	**Poll Coin Changer**
Buffer Input[2] to [9]	**Data**	**Value Coin Inserted (8 bytes)** **4 pair bytes** **Coins Deposited:** Z1 Z2 (01yyxxxx) (zzzzzzzz) yy = Coin routing. 00: CASH BOX 01: TUBES 10: NOT USED 11: REJECT xxxx = Coin type deposited (0 to 15). zzzzzzzz = The number of coins in the tube for the coin type accepted.
Buffer Input[10]=FC	**FC**	**FC=0xff**
Buffer Input[11]=CHK	**CHECKSUM**	

Fórmula para sacar valor real

$$ValorMoneda\ Real = Monto\ Total\ Insertado * ValorEscalaMonedas * 10^{-decimalplaces}$$

Coin Tubes Status

Send data

Buffer Output	Packet Format	Description
Buffer Output[0]=0x02	**Device**	**Coin Changer**
Buffer Output[1]=0x05	**Command**	**Tube Status Coin Changer**
Buffer Output[2]=FC	**FC**	**FC=0xff**
Buffer Output[3]=CHK	**CHECKSUM**	

Reception data

Buffer Input	Packet Format	Description
Buffer Input[0]=0x02	**Device**	**Coin Changer**
Buffer Input[1]=0x05	**Command**	**Tube Status Coin Changer**
Buffer Input[2] to [19]	**Data**	Z1 - Z2 = Tube Full Status - 2 bytes Indicates status of coin tube for coin types 0 to 15. b15 b14 b13 b12 b11 b10 b9 b8 | b7 b6 b5 b4 b3 b2 b1 b0 Z1 Z2 A bit is set to indicate a full tube. For example, bit 7 = set would indicate the tube for coin type 7 is full.

		Z3 - Z18 = Tube Status - 16 bytes Indicates the greatest number of coins that the changer “knows” definitely are present in the coin tubes. A bytes position in the 16 byte string indicates the number of coins in a tube for a particular coin type. For example, the first byte sent indicates the number of coins in a tube for coin type 0. Unsent bytes are assumed to be zero. For tube counts greater than 255, counts should remain at 255. **NOTE:** If a changer can detect a tube jam, defective tube sensor, or other malfunction, it will indicate the tube is "bad" by sending a tube full status and a count of zero for the malfunctioning coin type.
Buffer Input[20]=FC	**FC**	**FC=0xff**
Buffer Input[21]=CHK	**CHECKSUM**	

Coin PayOut

Send data

Buffer Output	Packet Format	Description
Buffer Output[0]=0x02	**Device**	**Coin Changer**
Buffer Output[1]=0x07	**Command**	**PayOut Coin Changer**
Buffer Output[2]= data	**Data**	**Amount PayOut Coin Changer (1 byte)** This value is expressed as the number of coin scaling factors that would sum to the value. For example, in a USA system using a scaling factor of 05, if the change to be paid out is 75 cents, then Y1 will equal fifteen. That is, the sum of fifteen nickels equal 75 cents. The coin changer will

		determine which actual denominations of coins will be paid out. In the 75 cent example, the coins may be 3 quarters; or, 7 dimes & 1 nickel; or, 2 quarters & 2 dimes & 1 nickel, etc.
Buffer Output[4]=FC	**FC**	**FC=0xff**
Buffer Output[5]=CHK	**CHECKSUM**	

Reception data

Buffer Input	Packet Format	Description
Buffer Input[0]=0x02	**Device**	**Coin Changer**
Buffer Input[1]=0x07	**Command**	**PayOut Coin Changer**
Buffer Output[2] to [17]	**Data**	Z1 = Total amount dispensed Z2 - Z17 = Number of each coin type paid out - 16 bytes This is the changer's response to the last VMC Alternative PAYOUT command (0FH-02H). Bytes are sent in ascending order of coin types. A bytes position in the string indicates the coin type. That is, byte one is the number of coins for coin type 1, byte two is the number of coins for coin type two, and so on. Unsent bytes are assumed to be zero. The changer clears payout data after an ACK response from the VMC. There is no defined limit on how long the actual payout takes.
Buffer Input[18]=FC	**FC**	**FC=0xff**

Buffer Input[19]=CHK	**CHECKSUM**	

Coin Accepted

Send data

Buffer Output	Packet Format	Description
Buffer Output[0]=0x02	**Device**	**Coin Changer**
Buffer Output[1]=0x08	**Command**	**Coin Accepted Coin Changer**
Buffer Output[3]=msb **Buffer OutPut[4]=lsb**	**Data**	**Coin Enable(2 bytes)** b15 b14 b13 b12 b11 b10 b9 b8 | b7 b6 b5 b4 b3 b2 b1 b0 Y1 Y2 A bit is set to indicate a coin type is accepted. For example, bit 6 is set to indicate coin type 6, bit 15 is set to indicate coin type 15, and so on. To disable the changer, disable all coin types by sending a data block containing 0000H. All coins are automatically disabled upon reset.
Buffer Output[5]=msb **Buffer Output[6]=lsb**	**Data**	**Manually Coin Enable (2 bytes)** b15 b14 b13 b12 b11 b10 b9 b8 | b7 b6 b5 b4 b3 b2 b1 b0 Y1 Y2 A bit is set to indicate dispense enable. For example, bit 2 is set to enable dispensing of coin type 2. This command enables/disables manual dispensing using optional inventory switches. All manual dispensing switches are automatically enabled upon reset
Buffer Output[7]=FC	**FC**	**FC=0xff**

Buffer Output[8]=CHK	**CHECKSUM**	

Reception data

Buffer Input	Packet Format	Description
Buffer Input[0]=0x02	**Device**	**Coin Changer**
Buffer Input[1]=0x08	**Command**	**Coin Accepted Coin Changer**
Buffer Input[2]	**Data**	**Confirmation PayOut Coin Changer (1 byte)** OK 0x01 FAIL 0x00
Buffer Input[3]=FC	**FC**	**FC=0xff**
Buffer Input[4]=CHK	**CHECKSUM**	

Coin Error

Send data

Buffer Output	Packet Format	Description
Buffer Output[0]=0x02	**Device**	**Coin Changer**
Buffer Output[1]=0x0a	**Command**	**Error Coin Changer**
Buffer Output[2]=FC	**FC**	**FC=0xff**
Buffer Output[3]=CHK	**CHECKSUM**	

Reception data

Buffer Input	Packet Format	Description
Buffer Input[0]=0x02	**Device**	**Coin Changer**
Buffer Input[1]=0x0a	**Command**	**Error Coin Changer**
Buffer Input[2] to [7]	**Data**	**Error Status Coin Changer (8 bytes)** This command requests the changer to report its current state of operation. The VMC should periodically transmit the command approximately every 1 to 10 seconds. The changer reports its current state of operation in a 2 byte code, msb is the main code and lsb is the sub-code. The code is reported as long as the condition exists and stops being reported as soon as the condition does not exist. Multiple 2 byte codes may be sent in response to a single command which could result in a maximum of eight 2 byte codes (8 bytes total).
Buffer Input[X]=FC	**FC**	**FC=0xff**
Buffer Input[X]=CHK	**CHECKSUM**	

Error table

Z1 / Z2	Status	Cause(s) of Status / Error
01 / 00	Powering up	Changer powering up / initialization
02 / 00	Powering down	Changer powering down
03 / 00	OK	Changer fully operational and ready to accept coins
04 / 00	Keypad shifted	MODE key pressed and held so that LED flashes indicating keypad in shifted state. Reverts to normal mode if no key pressed for 15 seconds
05 / 10	Manual Fill / Payout active	Manual Fill or Manual Payout mode of operation in progress (under control of the changer). This response must be reported at least once to allow the VMC to request a manual fill or manual payout report.
05 / 20	New Inventory Information Available	Changer not in Manual inventory mode, but new inventory information available.
06 / 00	Inhibited by VMC	All coin acceptance inhibited at request of VMC, possibly due to product dispenser jams, completely sold out, etc.
10 / Z2	General changer error	Z2 defined as: **00** Non specific error. **01** Check sum error #1. A check sum error over a particular data range of configuration field detected. **02** Check sum error #2. A check sum error over a secondary data range or configuration field detected. **03** Low line voltage detected. The changer has disabled acceptance or payout due to a low voltage condition.

Z1 / Z2	Status	Cause(s) of Status / Error
11 / Z2	Discriminator module error	Z2 defined as: **00** Non specific discriminator error. **10** Flight deck open. **11** Escrow Return stuck open. **30** Coin jam in sensor. **41** Discrimination below specified standard. **50** Validation sensor A out of range. The acceptor detects a problem with sensor A. **51** Validation sensor B out of range. The acceptor detects a problem with sensor B. **52** Validation sensor C out of range. The acceptor detects a problem with sensor C. **53** Operating temperature exceeded. The acceptor detects the ambient temperature has exceeded the changer's operating range, thus possibly affecting the acceptance rate. **54** Sizing optics failure. The acceptor detects an error in the sizing optics.
12 / Z2	Accept gate module error	Z2 defined as: **00** Non specific accept gate error. **30** Coins entered gate, but did not exit. **31** Accept gate alarm active. **40** Accept gate open, but no coin detected. **50** Post gate sensor covered before gate opened.
13 / Z2	Separator module error	Z2 defined as: **00** Non specific separator error **10** Sort sensor error. The acceptor detects an error in the sorting sensor.
14 / Z2	Dispenser module error	Z2 defined as: **00** Non specific dispenser error.
15 / Z2	Coin Cassette / tube module error	Z2 defined as: **00** Non specific cassette error. **02** Cassette removed. **03** Cash box sensor error. The changer detects an error in a cash box sensor. **04** Sunlight on tube sensors. The changer detects too much ambient light on one or more of the tube sensors.

Dispense Coin

Send data

Buffer Output	Packet Format	Description
Buffer Output[0]=0x02	**Device**	**Coin Changer**
Buffer Output[1]=0x0b	**Command**	**Dispense Coin**
Buffer Output[2]	**Data**	**Coin Type** Indicate coin type to be dispensed. Valid codes are 0H to FH to indicate coin types 0 to 15.
Buffer Output[3]	**Data**	**Numbers of coins** Indicate the number of coins to be dispensed. Range of 0 to 15
Buffer Output[4]=FC	**FC**	**FC=0xff**
Buffer Output[5]=CHK	**CHECKSUM**	

Reception data

Buffer Input	Packet Format	Description
Buffer Input[0]=0x02	**Device**	**Coin Changer**
Buffer Input[1]=0x0b	**Command**	**Dispense Coin**
Buffer Input[2]	**Data**	**Response** Data = 1 OK Data = 0 Fail
Buffer Input[3]=FC	**FC**	**FC=0xff**
Buffer Input[4]=CHK	**CHECKSUM**	

Identification ID Coin changer

Send data

Buffer Output	Packet Format	Description
Buffer Output[0]=0x02	**Device**	**Coin Changer**
Buffer Output[1]=0x0d	**Command**	**Coin Identification**
Buffer Output[2]=FC	**FC**	**FC=0xff**
Buffer Output[3]=CHK	**CHECKSUM**	

Reception data

Buffer Input	Packet Format	Description
Buffer Output[0]=0x02	**Device**	**Coin Changer**
Buffer Output[1]=0x0d	**Command**	**Coin Identification**
Buffer Input[2] to **Buffer Input[4]**	**Manufacturer** **3 bytes**	Identification code for the equipment supplier. Sent as ASCII characters. Currently defined codes are listed in the **EVA** document entitled "***European Vending Association Data Transfer Standard***" (**EVA-DTS**), the Audit Data Lists section, sub-section 2, "Manufacturer Codes".
Buffer Input[5] to **Buffer Input[16]**	**Serial Number** **12 bytes**	Factory assigned serial number. All bytes must be sent as ASCII characters, zeros (30H) and blanks (20H) are acceptable.
Buffer Input[17] to **Buffer Input[28]**	**Model** **12 bytes**	Manufacturer assigned model number and tuning number. All bytes must be sent as ASCII characters, zeros (30H) and blanks (20H) are acceptable. Each manufacturer should include information concerning the changer tuning revision.

Buffer Input[29] to **Buffer Input[32]**	**Optional features** **4 bytes**	Each of the 32 bits indicate an optional features availability. If the bit is set the feature is available. Bits should be sent in descending order, i.e. bit 31 is sent first and bit 0 is sent last. Currently defined options are: b0 - Alternative Payout method. This method allows changer designs that determine change payout. That is, the payout algorithm may reside in the changer instead of the VMC. b1 - Extended Diagnostic command supported. This command allows the VMC to request diagnostic status of the coin changer. b2 - Controlled Manual Fill and Payout commands supported. These commands allows the VMC to request the number of coin inserted or dispensed while the changer was in a controlled manual fill or payback mode. b3 - File Transport Layer (FTL) b4 - b31 Available for future use
Buffer Input[33]=FC	**FC**	**FC=0xff**
Buffer Input[34]=CHK	**CHECKSUM**	

Coin Errors

Send data

Buffer Output	Packet Format	Description
Buffer Output[0]=0x02	**Device**	**Coin Changer**
Buffer Output[1]=0x0a	**Command**	**Diagnostic error coin**
Buffer Output[2]=FC	**FC**	**FC=0xff**
Buffer Output[3]=CHK	**CHECKSUM**	

Reception data

Buffer Input	Packet Format	Description
Buffer Input[0]=0x02	Device	Coin Changer
Buffer Input[1]=0x0a	Command	Poll Coin Changer
Buffer Input[2] to [9]	Data	Errors Coin Inserted (8 bytes)
Buffer Input[10]=FC	FC	FC=0xff
Buffer Input[11]=CHK	CHECKSUM	

Z1 / Z2	Status	Cause(s) of Status / Error
01 / 00	Powering up	Changer powering up / initialization
02 / 00	Powering down	Changer powering down
03 / 00	OK	Changer fully operational and ready to accept coins
04 / 00	Keypad shifted	MODE key pressed and held so that LED flashes indicating keypad in shifted state. Reverts to normal mode if no key pressed for 15 seconds
05 / 10	Manual Fill / Payout active	Manual Fill or Manual Payout mode of operation in progress (under control of the changer). This response must be reported at least once to allow the VMC to request a manual fill or manual payout report.
05 / 20	New Inventory Information Available	Changer not in Manual inventory mode, but new inventory information available.
06 / 00	Inhibited by VMC	All coin acceptance inhibited at request of VMC, possibly due to product dispenser jams, completely sold out, etc.
10 / Z2	General changer error	Z2 defined as: **00** Non specific error. **01** Check sum error #1. A check sum error over a particular data range of configuration field detected. **02** Check sum error #2. A check sum error over a secondary data range or configuration field detected. **03** Low line voltage detected. The changer has disabled acceptance or payout due to a low voltage condition.

Z1 / Z2	Status	Cause(s) of Status / Error
11 / Z2	Discriminator module error	Z2 defined as: **00** Non specific discriminator error. **10** Flight deck open. **11** Escrow Return stuck open. **30** Coin jam in sensor. **41** Discrimination below specified standard. **50** Validation sensor A out of range. The acceptor detects a problem with sensor A. **51** Validation sensor B out of range. The acceptor detects a problem with sensor B. **52** Validation sensor C out of range. The acceptor detects a problem with sensor C. **53** Operating temperature exceeded. The acceptor detects the ambient temperature has exceeded the changer's operating range, thus possibly affecting the acceptance rate. **54** Sizing optics failure. The acceptor detects an error in the sizing optics.
12 / Z2	Accept gate module error	Z2 defined as: **00** Non specific accept gate error. **30** Coins entered gate, but did not exit. **31** Accept gate alarm active. **40** Accept gate open, but no coin detected. **50** Post gate sensor covered before gate opened.
13 / Z2	Separator module error	Z2 defined as: **00** Non specific separator error **10** Sort sensor error. The acceptor detects an error in the sorting sensor.
14 / Z2	Dispenser module error	Z2 defined as: **00** Non specific dispenser error.
15 / Z2	Coin Cassette / tube module error	Z2 defined as: **00** Non specific cassette error. **02** Cassette removed. **03** Cash box sensor error. The changer detects an error in a cash box sensor. **04** Sunlight on tube sensors. The changer detects too much ambient light on one or more of the tube sensors.

Bill Validator commands

Init

Send data

Buffer Output	Packet Format	Description
Buffer Output[0]=0x03	**Device**	**Bill Acceptor**
Buffer Output[1]=0x01	**Command**	**Init bill**
Buffer Output[2]=FC	**FC**	**FC=0xff**
Buffer Output[3]=CHK	**CHECKSUM**	

Reception data

Buffer Input	Packet Format	Description
Buffer Input[0]=0x03	**Device**	**Coin changer**
Buffer Input[1]=0x01	**Command**	**Init coin changer**
Buffer Input[2]	**Data**	**Response** 0x01 OK 0x00 FAIL
Buffer Input[3]=FC	**FC**	**FC=0xff**
Buffer Input[4]=CHK	**CHECKSUM**	

Bill Setup

Send data

Buffer Output	Packet Format	Description
Buffer Output[0]=0x03	**Device**	**Bill Validator**
Buffer Output[1]=0x05	**Command**	**Init Bill Validator**
Buffer Output[2]=FC	**FC**	**FC=0xff**
Buffer Output[3]=CHK	**CHECKSUM**	

Reception data

Buffer Input	Packet Format	Description
Buffer Input[0]=0x03	**Device**	**Bill Validator**
Buffer Input[1]=0x05	**Command**	**Init Bill Validator**
Buffer Input[2]	**Data**	**Bill Validator Feature Level** - 1 byte Indicates current feature level of the bill validator. Currently defined levels are: Level 1 - does not support option bits (Z1 = 01h) Level 2 - supports option bits (Z1 = 02h)
Buffer Input[3] to Buffer Input[4]	**Data**	Z2 - Z3 = Country / Currency Code - 2 bytes The packed BCD country / currency code of the bill validator can be sent in two different forms depending on the value of the left most BCD digit. If the left most digit is a 0, the International Telephone Code is used to indicate the country that the validator is set-up for. For example, the USA code is 00 01H (Z2 = 00 and Z3 = 01). If the left most digit is a 1, the latest version of the ISO 4217 numeric

		currency code is used (see Appendix A1). For example, the code for the US dollar is 18 40H (Z2 = 18 and Z3 = 40) and for the Euro is 1978 (Z2 = 19 and Z3 = 78).
Buffer Input[5] to Buffer Input[6]	**Data**	**Bill Scaling Factor** - 2 bytes All accepted bill values must be evenly divisible by this number. For example, this could be set to 0064H for the USA.
Buffer Input[7]	**Data**	**Decimal Places Bill Validator (1 byte)** Indicates the number of decimal places on a credit display. For example, this could be set to 02H in the USA.
Buffer Input[8] to Buffer Input[9]	**Data**	**Stacker Capacity Bill Validator (2 bytes)** Indicates the number of bills that the stacker will hold. For example, 400 bill capacity = 0190H.
Buffer Input[10] to Buffer Input[11]	**Data**	**Bill Security Levels** - 2 bytes Indicates the security level for bill types 0 to 15. Since not all validators support multiple security levels, validators that do not have this feature must report a “high” security level.
Buffer Input[12]	**Data**	**Escrow** Indicates the escrow capability of the bill validator. If Z11 = 00H, the bill validator does not have escrow capability. If Z11 = FFH, the bill validator has escrow capability
Buffer Input[13] to **Buffer Input[28]**	**Data**	**Bills Type Accepted (16 bytes)** Indicates the value of the bill types 0 to 15. Values must be sent in ascending order. This number is the bill's monetary value divided by the bill scaling factor. Unused bill types are sent as 00H. Unsent bill types are assumed to be zero. FFH bills are assumed to be vend tokens.
Buffer Input[29]=FC	**FC**	**FC=0xff**
Buffer Input[30]=CHK	**CHECKSUM**	

Bill Identification

Send data

Buffer Output	Packet Format	Description
Buffer Output[0]=0x03	**Device**	**Bill Validator**
Buffer Output[1]=0x06	**Command**	**Init Bill Validator**
Buffer Output[2]=FC	**FC**	**FC=0xff**
Buffer Output[3]=CHK	**CHECKSUM**	

Reception data

Buffer Input	Packet Format	Description
Buffer Input[0]=0x03	**Device**	**Bill Validator**
Buffer Input[1]=0x06	**Command**	**Init Bill Validator**
Buffer Input[2] to **Buffer Input[4]**	**Data**	**Manufacturer Code Bill Validator (3 bytes)** Identification code for the equipment supplier. Sent as ASCII characters.
Buffer Input[5] to **Buffer Input[16]**	**Data**	**Serial Number Bill Validator (12 bytes)** Factory assigned serial number. All bytes must be sent as ASCII characters, zeros (30H) and blanks (20H) are acceptable.
Buffer Input[17] to **Buffer Input[28]**	**Data**	**Model Bill Validator (12 bytes)** Manufacturer assigned model number and tuning number. All bytes must be sent as ASCII characters, zeros (30H) and blanks (20H) are acceptable. Each manufacturer should include

		information concerning the changer tuning revision.
Buffer Input[29] to Buffer Input[30]	**Data**	**Software version (2 byte)** Current software version. Must be send in packet BCD
Buffer Input[31]=FC	**FC**	**FC=0xff**
Buffer Input[32]=CHK	**CHECKSUM**	

Bill Poll

Send data

Buffer Output	Packet Format	Description
Buffer Output[0]=0x03	**Device**	**Bill Validator**
Buffer Output[1]=0x02	**Command**	**Poll Bill Validator**
Buffer Output[2]=FC	**FC**	**FC=0xff**
Buffer Output[3]=CHK	**CHECKSUM**	

Received data

Buffer Input	Packet Format	Description
Buffer Input[0]=0x03	**Device**	**Bill Validator**
Buffer Input[1]=0x02	**Command**	**Poll Bill Validator**
Buffer Input[2] to Buffer Input[9]	**Data**	**Value Bill Inserted (8 bytes)** Z1 (1yyyxxxx) yyy = Bill Routing; 000: BILL STACKED 001: ESCROW POSITION2

		010: BILL RETURNED 011: BILL TO RECYCLER1 100: DISABLED BILL REJECTED 101: BILL TO RECYCLER – MANUAL FILL1,3 110: MANUAL DISPENSE1 111: TRANSFERRED FROM RECYCLER TO CASHBOX1 xxxx = Bill Type (0 to 15)
Buffer Input[10]=FC	**FC**	**FC=0xff**
Buffer Input[11]=CHK	**CHECKSUM**	

Bill Stacker

Send data

Buffer Output	Packet Format	Description
Buffer Output[0]=0x03	**Device**	**Bill Validator**
Buffer Output[1]=0x03	**Command**	**Stacker Bill Validator**
Buffer Output[2]=FC	**FC**	**FC=0xff**
Buffer Output[3]=CHK	**CHECKSUM**	

Received data

Buffer Input	Packet Format	Description
Buffer Input[0]=0x03	**Device**	**Bill Validator**
Buffer Input[1]=0x03	**Command**	**Stacker Bill Validator**
Buffer Input[2]= msb **Buffer Input[3]=lsb**	**Data**	**Stacker Capacity (2 bytes)** Indicates stacker full condition and the number of bills in the stacker.

		msb lsb (Fxxxxxxx) (xxxxxxxx) F = 1 if stacker is full, 0 if not. xxxxxxxxxxxxxxx = The number of bills in the stacker.
Buffer Input[4]=FC	**FC**	**FC=0xff**
Buffer Input[5]=CHK	**CHECKSUM**	

Bill Accepted

Send data

Buffer Output	Packet Format	Description
Buffer Output[0]=0x03	**Device**	**Bill Validator**
Buffer Output[1]=0x04	**Command**	**Bills Accepted**
Buffer Output[2] to Buffer Input[5]	**Data**	**Bill Enable (4 bytes)** Y1 - Y2 = Bill Enable - 2 bytes Indicates what type of bills are accepted. b15 b14 b13 b12 b11 b10 b9 b8 | b7 b6 b5 b4 b3 b2 b1 b0 Y1 Y2 Y3 - Y4 = Bill Escrow Enable: Bill types are 0 to 15. A bit is set to indicate enable of escrow for a bill type.
Buffer Output[6]=FC	**FC**	**FC=0xff**
Buffer Output[7]=CHK	**CHECKSUM**	

Received data

Buffer Input	Packet Format	Description
Buffer Input[0]=0x03	**Device**	**Bill Validator**
Buffer Input[1]=0x03	**Command**	**Stacker Bill Validator**
Buffer Input[2]	**Data**	**Bill Enable Confirmation** 0x00 FAIL 0x01 OK
Buffer Input[3]=FC	**FC**	**FC=0xff**

Bill Diagnostic

Send data

Buffer Output	Packet Format	Description
Buffer Output[0]=0x03	**Device**	**Bill Validator**
Buffer Output[1]=0x07	**Command**	**Diagnostic Bill Validator**
Buffer Output[2]=FC	**FC**	**FC=0xff**
Buffer Output[3]=CHK	**CHECKSUM**	

Received data

Buffer Input	Packet Format	Description
Buffer Input[0]=0x03	**Device**	**Bill Validator**
Buffer Input[1]=0x07	**Command**	**Diagnostic Bill Validator (8 bytes)**

Buffer Input[2] to Buffer Input[9]	**Data**	
Buffer Input[4]=FC	**FC**	**FC=0xff**
Buffer Input[5]=CHK	**CHECKSUM**	

Status: **Bill Validator (Only)**

(00000001) = Defective Motor[3] - One of the motors has failed to perform its expected assignment.

(00000010) = Sensor Problem[3] - One of the sensors has failed to provide its response.

(00000011) = Validator Busy[2] - The validator is busy and can not answer a detailed command right now.

(00000100) = ROM Checksum Error[3] - The validators internal checksum does not match the calculated checksum.

(00000101) = Validator Jammed[3] - A bill(s) has jammed in the acceptance path.

(00000110) = Validator was reset[1] - The validator has been reset since the last POLL.

(00000111) = Bill Removed[1] - A bill in the escrow position has been removed by an unknown means. A BILL RETURNED message should also be sent.

(00001000) = Cash Box out of position[3] - The validator has detected the cash box to be open or removed.

(00001001) = Validator Disabled[2] - The validator has been disabled, by the VMC or because of internal conditions.

(00001010) = Invalid Escrow request[1] - An ESCROW command was requested for a bill not in the escrow position.

(00001011) = Bill Rejected[1] - A bill was detected, but rejected because it could not be identified.

(00001100) = Possible Credited Bill Removal[1] – There has been an attempt to remove a credited (stacked) bill.

Note:

- validators must have a means to disable this code due to potential older VMC issues.
- virtually all VMCs designed prior to this code's introduction (10/16/02) will not support it.
- It is a vending machine system issue as to what is done when this code is received.

(010xxxxx) = Number of attempts to input a bill while validator is disabled.[1]

Status:	Bill Recycler (Only)
(00100001) =	Escrow request[1] - An escrow lever activation has been detected. If a button is present and activated.
(00100010) =	Dispenser Payout Busy[2] - The dispenser is busy activating payout devices.
(00100011) =	Dispenser Busy[2] - The dispenser is busy and can not answer a detailed command right now.
(00100100) =	Defective Dispenser Sensor[4] - The dispenser has detected one of the dispenser sensors behaving abnormally.
(00100101) =	Not Used
(00100110) =	Dispenser did not start / motor problem[4].
(00100111) =	Dispenser Jam[4] - A dispenser payout attempt has resulted in jammed condition.
(00101000) =	ROM checksum error[4] - The dispensers internal checksum does not match the calculated checksum. (If separate from validator microprocessor.)
(00101001) =	Dispenser disabled – dispenser disabled because of error or bill in escrow position.
(00101010) =	Bill waiting[2,5] – waiting for customer removal
(00101011) =	Not Used
(00101100) =	Not Used
(00101101) =	Not Used
(00101110) =	Not Used
(00101111) =	Filled key pressed[1] – The VMC should request a new DISPENSER STATUS.

Bill Security

Send data

Buffer Output	Packet Format	Description
Buffer Output[0]=0x03	**Device**	**Bill Validator**

Buffer Output[1]=0x08	**Command**	**Bills Accepted**
Buffer Output[2]= msb **Buffer Output[3]= lsb**	**Data**	**Bill Enable (2 bytes)** Indicates what type of bills are accepted. b15 b14 b13 b12 b11 b10 b9 b8 | b7 b6 b5 b4 b3 b2 b1 b0 Y1 Y2 A bit is set to indicate the type of bill(s) which are set to a “high” security level.
Buffer Output[4]=FC	**FC**	**FC=0xff**
Buffer Output[5]=CHK	**CHECKSUM**	

Received data

Buffer Input	Packet Format	Description
Buffer Input[0]=0x03	**Device**	**Bill Validator**
Buffer Input[1]=0x08	**Command**	**Security Bill Validator**
Buffer Input[2]	**Data**	**Bill Enable Confirmation** 0x00 FAIL 0x01 OK
Buffer Input[3]=FC	**FC**	**FC=0xff**

Bill Escrow

Send data

Buffer Output	Packet Format	Description
Buffer Output[0]=0x03	**Device**	**Bill Validator**
Buffer Output[1]=0x0a	**Command**	**Escrow**

Buffer Output[2]	**Data**	**Action (1 byte)** 1 = accepted bill 0 = rejected bill
Buffer Output[3]=FC	**FC**	**FC=0xff**
Buffer Output[4]=CHK	**CHECKSUM**	

Received data

Buffer Input	Packet Format	Description
Buffer Input[0]=0x03	**Device**	**Bill Validator**
Buffer Input[1]=0x0a	**Command**	**Security Bill Validator**
Buffer Input[2]	**Data**	**Bill Enable Confirmation** 0x00 FAIL 0x01 OK
Buffer Input[3]=FC	**FC**	**FC=0xff**
Buffer Output[4]=CHK	**CHECKSUM**	

Bill Recycler Setup

Send data

Buffer Output	Packet Format	Description
Buffer Output[0]=0x03	**Device**	**Bill Validator**
Buffer Output[1]=0x0b	**Command**	**Bill Recycler Setup**
Buffer Output[2]=FC	**FC**	**FC=0xff**
Buffer Output[3]=CHK	**CHECKSUM**	

Received data

Buffer Input	Packet Format	Description
Buffer Input[0]=0x03	**Device**	**Bill Validator**
Buffer Input[1]=0x0a	**Command**	**Security Bill Validator**
Buffer Input[2] to Buffer Input[3]	**Data**	Z1 – Z2 = Bill Type Routing - 2 bytes Indicates what bill types can be routed to the Recycler dispenser. b15 b14 b13 b12 b11 b10 b9 b8 | b7 b6 b5 b4 b3 b2 b1 b0 Z1 Z2 Bit is set to indicate a bill type can be routed to the dispenser. Valid bill types are 0 to 15.
Buffer Input[4]=FC	**FC**	**FC=0xff**
Buffer Output[5]=CHK	**CHECKSUM**	

Bill Recycler Enable

Send data

Buffer Output	Packet Format	Description
Buffer Output[0]=0x03	**Device**	**Bill Validator**
Buffer Output[1]=0x0c	**Command**	**Bill Recycler Setup**
Buffer Output[2] to Buffer Input[3]	**Data**	Y1 - Y2 = Manual Dispense Enable - 2 bytes b15 b14 b13 b12 b11 b10 b9 b8 | b7 b6 b5 b4 b3 b2 b1 b0 Z1 Z2 A bit is set to indicate manual dispense enable. For example, bit 2 is set to enable manual dispensing of bill type 2. This command enables/disables

		manual dispensing using optional inventory switches. All manual dispensing switches are automatically disabled upon reset.
Buffer Output[4]=FC	**FC**	**FC=0xff**
Buffer Output[5]=CHK	**CHECKSUM**	

Received data

Buffer Input	**Packet Format**	**Description**
Buffer Input[0]=0x03	**Device**	**Bill Validator**
Buffer Input[1]=0x0c	**Command**	**Security Bill Validator**
Buffer Input[2]	**Data**	0x00 FAIL 0x01 OK
Buffer Input[3]=FC	**FC**	**FC=0xff**
Buffer Output[4]=CHK	**CHECKSUM**	

Bill Recycler Payout

Send data

Buffer Output	**Packet Format**	**Description**
Buffer Output[0]=0x03	**Device**	**Bill Validator**
Buffer Output[1]=0x09	**Command**	**Bill Recycler Setup**
Buffer Output[2] to Buffer Input[3]	**Data**	Y1 – Y2 = Value of bills to be paid out. Y1 and Y2 are defined as the total value of bills to be paid out. This value is expressed as the actual credit value divided by the bill scaling factor.

		For example, in a USA system using a scaling factor of 100 (64H), if the change to be paid out is $15.00, then Y1 will equal 15. The bill dispenser will determine which actual denominations of bills will be paid out. In the $15.00 example, the bills may be 3 $5 bills; or, 1 $10 bill & 1 $5 bill; or, 2 $5 bills & 5 $1 bills, etc. . There is no defined limit on how long the actual dispense takes.
Buffer Output[4]=FC	**FC**	**FC=0xff**
Buffer Output[5]=CHK	**CHECKSUM**	

Received data

Buffer Input	**Packet Format**	**Description**
Buffer Input[0]=0x03	**Device**	**Bill Validator**
Buffer Input[1]=0x0c	**Command**	**Security Bill Validator**
Buffer Input[2] to Buffer Input[3]	**Data**	PayOut Value (2 bytes) Amount dispensed
Buffer Input[4]=FC	**FC**	**FC=0xff**
Buffer Output[5]=CHK	**CHECKSUM**	

History version manual

V1.0 Date 05/2014

- First revision manual

V1.0.1 Date 06/2014

- Added Dispense Coin command
- Add parameter Level in Init Coin command
- Eliminate the Baudrate command

V2.0 Date 08/2014

- Added Display command value
- Rev 2 of PCB updated

V3.0

- Add new function coin changer
- Rev 3 of PCB updated