

Protocol Documentation

Vendron Default Machine Control Communication

Revision History

Notes: Latest changes in the document are highlighted in yellow.

Version	Release Date	Description
0.1	26 Oct 2014	First draft

Content

Acronyms.....	3
General.....	4
Communication Scenarios.....	6
Check Machine Status.....	6
Get current refrigeration temperature.....	6
Get Door Open Status.....	7
Dispense Product.....	7
Retry Dispense Product (reverse motor half cycle and turn half cycle).....	8
Check Dispense Status (Drop Sensor).....	8
Set refrigeration temperature.....	9
Turn refrigeration On/Off.....	10
Turn Specific Relay On/Off.....	10
Read Specific Sensor Status.....	11
Home Elevator.....	11
Move Elevator.....	12
Check Elevator Status.....	12
Contact Information.....	14

Acronyms

Acronyms	Definitions
RS232	A standard for serial communication transmission of data
TCP/IP	Transmission Control Protocol over Internet Protocol
VMC	Vending Machine Controller
XOR	Exclusive OR in bit operation

General

1. This protocol documentation defines the communication protocol between Vendron software and the vending machine controller (VMC) board over RS232 primarily but also for RS485, RS422, TCP/IP or any other string-based interface.
2. In the communication prospect, Vendron will also be the master while the VMC board as the slave, where the vending control board should only reply whenever received a command from the master.
3. For serial-based communication like RS232, the following connection parameters are the preferred settings. Other parameter settings can also be used if the VMC board cannot support:
 - a. Baud Rate: minimum 9600 or higher (the higher the better)
 - b. Data Bits: 8
 - c. Stop Bit: 1
 - d. Parity: None
 - e. Flow Control: None
4. General data packet format (all values are in HEX):

Start Byte 1	Start Byte 2	Controller ID	Command / Response	Data	Checksum Byte	End Byte 1	End Byte 2
AA	BB	01	[MM]	[Multi-bytes]	[SS]	CC	DD

Notes:

- a. All data packets always start with AA BB and ends with CC DD
- b. Controller ID is the VMC identification number starting from 01. For 1-to-1 communication channel like serial communication (RS232), this byte is always 01.
- c. Command byte is one byte and when the VMC responds to a command from Vendron, this byte should be the same command byte received from Vendron. If any hardware or module not supported by the VMC, such as the elevator for example, upon receiving the command the VMC should just reply with FF in the data byte.

Command / Response Byte [MM]	Description
01	Check machine status
02	Get current refrigeration temperature
03	Get door open status
11	Dispense product
12	Retry dispense product (reverse half cycle and turn half cycle)
13	Check dispense status (drop sensor)
21	Set refrigeration temperature
22	Turn refrigeration on / off
23	Turn specific relay on / off (such as lighting, etc)
24	Read specific sensor status (such as motion sensor, etc)
31	Home elevator
32	Move elevator

33	Check elevator status
FF	Data received error (Recipient should resend the last sent data packet)

- d. Data bytes can be zero byte to multiple number of bytes.
- e. Checksum byte is the exclusive OR (XOR) of all the bytes from Start Byte 1 to the last Data Byte (not including the Checksum byte itself, End Byte 1 and End Byte 2). If wrong checksum is received by either side (Vendron or VMC), it should immediately reply with the response FF like following:

Start Byte 1	Start Byte 2	Controller ID	Command / Response	Data	Checksum Byte	End Byte 1	End Byte 2
AA	BB	01	FF	-	EF	CC	DD

Upon received this error response, the recipient should immediately resend the last sent data packet.

Communication Scenarios

Check Machine Status

Command from Vendron:

Start Byte 1	Start Byte 2	Controller ID	Command / Response	Data	Checksum Byte	End Byte 1	End Byte 2
AA	BB	01	01	-	11	CC	DD

Response from VMC:

Start Byte 1	Start Byte 2	Controller ID	Command / Response	Data (Multi-bytes)	Checksum Byte	End Byte 1	End Byte 2
AA	BB	01	01	[RR] ...	[SS]	CC	DD

Data Byte [RR]	Description
00	Machine normal and ready
10 [X1] ... [Xn]	Machine error, subsequent bytes are the error codes

Error Code [Xn]	Description
01	Machine error: motor not connected / malfunction
02	Machine error: power failure
03	Machine error: refrigerator error
04	Machine error: drop sensor malfunction
[Xn]	Further errors can be defined with subsequent numbers

Example: With refrigeration error and drop sensor malfunction

Vendron --> VMC	<u>AA</u> <u>BB</u> <u>01</u> <u>01</u> <u>11</u> <u>CC</u> <u>DD</u>
Vendron <-- VMC	<u>AA</u> <u>BB</u> <u>01</u> <u>01</u> <u>10</u> <u>03</u> <u>04</u> <u>06</u> <u>CC</u> <u>DD</u>

Get current refrigeration temperature

Command from Vendron:

Start Byte 1	Start Byte 2	Controller ID	Command / Response	Data	Checksum Byte	End Byte 1	End Byte 2
AA	BB	01	02	-	12	CC	DD

Response from VMC:

Start Byte 1	Start Byte 2	Controller ID	Command / Response	Data (Multi-bytes)	Checksum Byte	End Byte 1	End Byte 2
AA	BB	01	02	Values in ASCII string including decimal point and unit	[SS]	CC	DD

Example: Current refrigeration temperature reading is 8.5C (in ASCII string: 38 2E 35 43)

Vendron --> VMC	<u>AA</u> <u>BB</u> <u>01</u> <u>02</u> <u>12</u> <u>CC</u> <u>DD</u>
-----------------	---

Vendron <-- VMC AA BB 01 02 38 2E 35 43 72 CC DD

Get Door Open Status

Command from Vendron:

Start Byte 1	Start Byte 2	Controller ID	Command / Response	Data	Checksum Byte	End Byte 1	End Byte 2
AA	BB	01	03	-	13	CC	DD

Response from VMC:

Start Byte 1	Start Byte 2	Controller ID	Command / Response	Data	Checksum Byte	End Byte 1	End Byte 2
AA	BB	01	03	[RR]	[SS]	CC	DD

Data Byte [RR]	Description
00	Door closed
01	Door opened

Example: Door opened

Vendron --> VMC AA BB 01 03 13 CC DD
 Vendron <-- VMC AA BB 01 03 01 12 CC DD

Dispense Product

Command from Vendron:

Start Byte 1	Start Byte 2	Controller ID	Command / Response	Data (2 bytes)	Checksum Byte	End Byte 1	End Byte 2
AA	BB	01	11	[Row] [Col]	[SS]	CC	DD

Response from VMC:

Start Byte 1	Start Byte 2	Controller ID	Command / Response	Data	Checksum Byte	End Byte 1	End Byte 2
AA	BB	01	11	[RR]	[SS]	CC	DD

Data Byte [RR]	Description
00	Completed normally
10	Error: unknown condition
11	Error: motor not started
12	Error: motor not fully rotated
[1n]	Further errors can be defined with subsequent numbers

Note: VMC does not need to return the drop sensor status in this response, but upon receiving this command, it should clear the drop sensor status first.

Example: Dispense row 1 column 5

Vendron --> VMC AA BB 01 11 01 05 05 CC DD

Vendron <-- VMC AA BB 01 11 00 01 CC DD

Retry Dispense Product (reverse motor half cycle and turn half cycle)

Command from Vendron:

Start Byte 1	Start Byte 2	Controller ID	Command / Response	Data (2 bytes)	Checksum Byte	End Byte 1	End Byte 2
AA	BB	01	12	[Row] [Col]	[SS]	CC	DD

Response from VMC:

Start Byte 1	Start Byte 2	Controller ID	Command / Response	Data	Checksum Byte	End Byte 1	End Byte 2
AA	BB	01	12	[RR]	[SS]	CC	DD

Data Byte [RR]	Description
00	Completed normally
10	Error: unknown condition
11	Error: motor not started
12	Error: motor not fully rotated
[1n]	Further errors can be defined with subsequent numbers

Note: VMC does not need to return the drop sensor status in this response, but upon receiving this command, it should clear the drop sensor status first.

Example: Retry Dispense row 1 column 5

Vendron --> VMC AA BB 01 12 01 05 06 CC DD
 Vendron <-- VMC AA BB 01 12 00 02 CC DD

Check Dispense Status (Drop Sensor)

Command from Vendron:

Start Byte 1	Start Byte 2	Controller ID	Command / Response	Data	Checksum Byte	End Byte 1	End Byte 2
AA	BB	01	13	-	03	CC	DD

Response from VMC:

Start Byte 1	Start Byte 2	Controller ID	Command / Response	Data	Checksum Byte	End Byte 1	End Byte 2
AA	BB	01	13	[RR]	[SS]	CC	DD

Data Byte [RR]	Description
00	Product dispensed properly (Drop sensor detected)
10	Product dispensed failed (Drop sensor NOT detected)

Example: Product dispensed properly and detected

Vendron --> VMC AA BB 01 13 03 CC DD

Vendron <-- VMC AA BB 01 13 00 03 CC DD

Scenario: Full Product dispense and retry

Vendron		VMC
Requests dispense product from row 1 col 5	<p>-----> <u>AA BB 01 11 01 05 05 CC DD</u></p> <p><----- <u>AA BB 01 11 00 01 CC DD</u></p>	Clears drop sensor status and turns motor at row 1 col 5
Check dispense status if product dropped	<p>-----> <u>AA BB 01 13 03 CC DD</u></p> <p><----- <u>AA BB 01 13 10 13 CC DD</u></p>	Replies motor turned completely Product not detected by drop sensor
Requests retry dispense (half cycle)	<p>-----> <u>AA BB 01 12 01 05 06 CC DD</u></p> <p><----- <u>AA BB 01 12 00 02 CC DD</u></p>	Replies product dispense failed Reverse half-cycle for motor at row 1 col 5 and then turn half-cycle to retry dispense the product
Check dispense status if product dropped	<p>-----> <u>AA BB 01 13 03 CC DD</u></p> <p><----- <u>AA BB 01 13 00 03 CC DD</u></p>	Replies dispense retry completed Product now detected by drop sensor
		Replies product dispense success

Set refrigeration temperature

Command from Vendron:

Start Byte 1	Start Byte 2	Controller ID	Command / Response	Data	Checksum Byte	End Byte 1	End Byte 2
AA	BB	01	21	Values in ASCII string including decimal point and unit	[SS]	CC	DD

Response from VMC:

Start Byte 1	Start Byte 2	Controller ID	Command / Response	Data	Checksum Byte	End Byte 1	End Byte 2
AA	BB	01	21	[RR]	[SS]	CC	DD

Data Byte [RR]	Description
00	Temperature set successfully
10	Failed

Example: 6.5C (in ASCII string: 36 2E 35 43)

Vendron --> VMC AA BB 01 21 36 2E 35 43 5F CC DD
 Vendron <-- VMC AA BB 01 21 00 31 CC DD

Turn refrigeration On/Off

Command from Vendron:

Start Byte 1	Start Byte 2	Controller ID	Command / Response	Data	Checksum Byte	End Byte 1	End Byte 2
AA	BB	01	22	[PP]	[SS]	CC	DD

Data Byte [PP]	Description
00	Set refrigeration Off
01	Set refrigeration On

Response from VMC:

Start Byte 1	Start Byte 2	Controller ID	Command / Response	Data	Checksum Byte	End Byte 1	End Byte 2
AA	BB	01	22	[RR]	[SS]	CC	DD

Data Byte [RR]	Description
00	Refrigeration set On/Off successfully
10	Failed

Example: Set refrigeration off

Vendron --> VMC AA BB 01 22 00 32 CC DD

Vendron <-- VMC AA BB 01 22 00 32 CC DD

Turn Specific Relay On/Off

Command from Vendron:

Start Byte 1	Start Byte 2	Controller ID	Command / Response	Data	Checksum Byte	End Byte 1	End Byte 2
AA	BB	01	23	[Rn] [PP]	[SS]	CC	DD

Data Byte [Rn] [PP]	Description
[Rn] 00	Set relay [Rn] Off, such as 01 00 will set relay 1 off
[Rn] 01	Set relay [Rn] On, such as 01 01 will set relay 1 on

Response from VMC:

Start Byte 1	Start Byte 2	Controller ID	Command / Response	Data	Checksum Byte	End Byte 1	End Byte 2
AA	BB	01	23	[RR]	[SS]	CC	DD

Data Byte [RR]	Description
00	Relay turned On/Off successfully
10	Failed

Example: Set relay 1 on

Vendron --> VMC	<u>AA</u> <u>BB</u> <u>01</u> 23 01 01 <u>33</u> <u>CC</u> <u>DD</u>
Vendron <-- VMC	<u>AA</u> <u>BB</u> <u>01</u> 23 00 <u>33</u> <u>CC</u> <u>DD</u>

Read Specific Sensor Status

Command from Vendron:

Start Byte 1	Start Byte 2	Controller ID	Command / Response	Data	Checksum Byte	End Byte 1	End Byte 2
AA	BB	01	24	Sensor Number [Sn]	[SS]	CC	DD

Response from VMC:

Start Byte 1	Start Byte 2	Controller ID	Command / Response	Data	Checksum Byte	End Byte 1	End Byte 2
AA	BB	01	24	[RR]	[SS]	CC	DD

Data Byte [RR]	Description
00	Sensor Off
01	Sensor On
10	Failed

Example: Read sensor 1 status and the status is off

Vendron --> VMC	<u>AA</u> <u>BB</u> <u>01</u> 24 01 <u>35</u> <u>CC</u> <u>DD</u>
Vendron <-- VMC	<u>AA</u> <u>BB</u> <u>01</u> 24 00 <u>34</u> <u>CC</u> <u>DD</u>

Home Elevator

Command from Vendron:

Start Byte 1	Start Byte 2	Controller ID	Command / Response	Data	Checksum Byte	End Byte 1	End Byte 2
AA	BB	01	31	-	21	CC	DD

Response from VMC:

Start Byte 1	Start Byte 2	Controller ID	Command / Response	Data	Checksum Byte	End Byte 1	End Byte 2
AA	BB	01	31	[RR]	[SS]	CC	DD

Data Byte [RR]	Description
00	Homing started (Use command 33 to check elevator status to know if the homing completed)
10	Failed

Example: Home Elevator

Vendron --> VMC	<u>AA</u> <u>BB</u> <u>01</u> 31 <u>21</u> <u>CC</u> <u>DD</u>
Vendron <-- VMC	<u>AA</u> <u>BB</u> <u>01</u> 31 00 <u>21</u> <u>CC</u> <u>DD</u>

Move Elevator

Command from Vendron:

Start Byte 1	Start Byte 2	Controller ID	Command / Response	Data (Multi-bytes)	Checksum Byte	End Byte 1	End Byte 2
AA	BB	01	32	Elevator position to move to, in hex value [PP]...	21	CC	DD

Response from VMC:

Start Byte 1	Start Byte 2	Controller ID	Command / Response	Data	Checksum Byte	End Byte 1	End Byte 2
AA	BB	01	32	[RR]	[SS]	CC	DD

Data Byte [RR]	Description
00	Elevator started moving (Use command 33 to check elevator status to know if the moving completed)
10	Failed

Example: Move the elevator to position 312 (in hex 01 38)

Vendron --> VMC AA BB 01 32 01 38 1B CC DD

Vendron <-- VMC AA BB 01 32 00 22 CC DD

Check Elevator Status

Command from Vendron:

Start Byte 1	Start Byte 2	Controller ID	Command / Response	Data	Checksum Byte	End Byte 1	End Byte 2
AA	BB	01	33	-	23	CC	DD

Response from VMC:

Start Byte 1	Start Byte 2	Controller ID	Command / Response	Data	Checksum Byte	End Byte 1	End Byte 2
AA	BB	01	33	[RR] ...	[SS]	CC	DD

Data Byte [RR]	Description
00	Elevator ready / completed last action
01	Elevator busy
10 [X1] ... [Xn]	Error, subsequent bytes are the error codes. If no following error code, it means error unknown.

Error Code [Xn]	Description
01	Encoder error
02	Move timeout
03	Lower Limit error

04	Upper Limit error
[Xn]	Further errors can be defined with subsequent numbers

Example: Elevator reports encoder error and lower limit error

Vendron --> VMC	<u>AA BB 01</u> 33 <u>23 CC DD</u>
Vendron <-- VMC	<u>AA BB 01</u> 33 10 01 03 <u>31 CC DD</u>

Contact Information



Website: www.silkron.com

Email: support@silkron.com

Silkron Technology Sdn Bhd

1-2-3, Mayang Mall

11950 Bayan Baru,

Penang, Malaysia.

Tel: +604 - 6455 896

