Serial Protocol Specification for the Communication between Programmable Logical Controller PLC and MDB Vending Peripherals via SPS2MDB Interface.

This protocol specification was initially defined by BonusData AG on February 15th, 2006 and published as SPS2MDB document, Version 1.00

Copyright © 2006-2008 by BonusData AG, Tiergartenweg 1, CH-4710 Balsthal.

http://www.bonusdata.net
Mailto: team@bonusdata.net
Phone: +41 62 3919600
Fax. +41 62 3919601
Contents:

CHAPTER 1: BASICS .......................................................................................................................... 4
  Introduction ................................................................................................................................. 4
  Basic conditions ........................................................................................................................ 4
  Electrical connection .................................................................................................................. 4
  Operating states ......................................................................................................................... 5
  Start Condition .......................................................................................................................... 5
  Stop Condition ........................................................................................................................... 5
  Transmitting one byte ............................................................................................................... 5
  Transmitting more than one byte ............................................................................................ 6
  Synchronization ......................................................................................................................... 6
  Timeout ..................................................................................................................................... 6

CHAPTER 2: VEND SEQUENCE ................................................................................................. 8
  Introduction ................................................................................................................................. 8
  Simplified vend procedure ......................................................................................................... 8

CHAPTER 3: COMMANDS ......................................................................................................... 9
  Command: Status Request (0xA0) ............................................................................................ 9
  Command: Enable (0xA1) .......................................................................................................... 9
  Command: Disable (0xA2) ......................................................................................................... 10
  Command: Begin Session (0xA3) ............................................................................................ 10
  Command: Vend Request (0xA4) ............................................................................................. 10
  Command: Vend Approved (0xA5) .......................................................................................... 10
  Command: Vend Denied (0xA6) ............................................................................................... 11
  Command: Vend Success (0xA7) ............................................................................................. 11
  Command: Vend Failure (0xA8) .............................................................................................. 11
  Command: End Session (0xA9) ............................................................................................... 11
  Command: Cancel Session (0xAA) ......................................................................................... 11
  Command: Device Reset (0xAB) ............................................................................................. 12

CHAPTER 4: APPLICATION SPECIFIC FUNCTIONALITY .................................................... 13
  Introduction: ............................................................................................................................... 13
  Setup Of Prices .......................................................................................................................... 13
  Multi Vend .................................................................................................................................. 13

CHAPTER 5: HARDWARE DESCRIPTION .............................................................................. 14
  SPS2MDB Technical Data ......................................................................................................... 14
  SPS2MDB to PLC Interconnection ............................................................................................ 14

CHAPTER 6: SOFTWARE DOWNLOAD ................................................................................ 15
  Connecting Hyper Terminal ....................................................................................................... 15
  Monitor ...................................................................................................................................... 16
  Download of a new software release ....................................................................................... 16

CHAPTER 7: SETUP OF A TEST APPLICATION .................................................................... 17

Figure: 4 ......................................................................................................................................... 17

TEST SEQUENCES FOR SPS/PLC COMMUNICATION COMMANDS ..................................... 17
  Status ......................................................................................................................................... 17
  Enable / Disable ......................................................................................................................... 17
  Enter session .............................................................................................................................. 17
  Enter vend / Vend denied ........................................................................................................... 18
  Enter vend / Vend failed ............................................................................................................ 18
  Enter vend / Vend success ......................................................................................................... 18
  Enter vend / Vend approved (No further coins accepted) ......................................................... 18
  Escrow (Acceptor only) ............................................................................................................ 18
  Remove card (Reader only) ........................................................................................................ 18

ALLOWED COMMANDS WITHIN CERTAIN STATES ......................................................... 18
  Command / State ....................................................................................................................... 18

CHAPTER 8: APPENDIXES ....................................................................................................... 20
  A: Figures .................................................................................................................................. 20
Chapter 1: Basics

Introduction

The project name for 'Connect MDB Vending Peripherals to a Programmable Logical Controller (PLC)' is SPS2MDB. Where SPS is the shortcut for 'Speicher Programmierbare Steuerungen' like PLC stands for 'Programmable Logical Controller'.

Because PLC devices often do not have a RS-232 interface, or if there is one, it is quite difficult or even impossible to implement the MDB protocol programmatically. Even then a electrical interface from RS-232 to MDB (Multi Drop Protocol) of the vending peripherals. So it was obviously to design a direct interface from standard digital inputs and outputs to MDB. This way the interface can be used with any PLC with a few spare inputs and outputs. Thus allows to transfer small amounts of data at low baud rate in both direction.

MDB is a 'Multi Drop Protocol' mainly used in vending machines for the internal communication among the vending machine controller VMC and the peripherals like cash less payment system, coin acceptors, bill validator, credit card reader etc.

It is not part of this document to define all the needed error handling but to specify layer one and two of the OSI model.

Basic conditions

- The PLC needs to have two available inputs and two outputs to connect a SPS2MDB device.
- The standard voltage is 24VDC.
- Passive is defined as near 0V equal a logical ‘O’.
- Active is define as near 24VDC equal l logical ‘1’.
- In each direction there is a data line and a clock line.
- The device starting the communication with a start condition is master for this data transfer session and must end the communication by a stop condition.
- The devices synchronize each other with the clock and acknowledge.
- Numbers in parentheses (X) are states in diagram #1.

Electrical connection

![Figure 1](image-url)
**Operating states**

a) No device connected (1):
- Inputs ‘Passive’ (not connected).
- Outputs ‘Active’.

b) Device connected and idle (2):
- Inputs ‘Active’ by connected device.
- Outputs ‘Active’.

c) Device connected and communication:
- Clock ‘Passive’ -> Invalid data.
- Clock ‘Active’ -> Valid data must be present.

**Start Condition**

See diagram #1:

The start condition consists of following sequence:
- The device set its data line ‘Passive’ (3).
- The opposite device set its data line ‘Passive’ -> Acknowledge to master (4).
- The opposite device set its clock line ‘Passive’ (4).
- The device set its clock line ‘Passive’ (5).
- The device is now master and the opposite device is now slave (5).
- The opposite device sets clock line ‘Active’ (6).

**Stop Condition**

See diagram #1:

The stop condition consists of following sequence:
- The master set its clock line ‘Active’ (12).
- The slave set its clock line ‘Passive’ -> Acknowledge to master (13).
- The master set its data line ‘Active’ (14).
- The slave set its clock line ‘Active’ -> The devices are now idle (16).

Both devices are now idle and ready for a new start condition.

**Transmitting one byte**

See diagram #1:

1. The master starting the communication with a start condition:
   - The device set its data line ‘Passive’ (3).
   - The opposite device set its data line ‘Passive’ -> Acknowledge (4).
   - The opposite device set its clock line ‘Passive’ (4).
   - The device set its clock line ‘Passive’ (5).
   - The device is now master and the opposite device is now slave (5).
   - The opposite device sets clock line ‘Active’ (6).

2. The master transmits 8 bits with following sequence in a loop:
   - According the value of the most significant data bit, the master set its data line ‘Active | Passive’ (6).
   - The master set its clock line ‘Active’ (7).
   - The slave reads the value of the data line ‘Active | Passive’.
   - The slave set its clock line ‘Passive’ -> Acknowledge to master (8).
   - The master set its clock line ‘Passive’ -> Acknowledge to slave (9).
   - The slave set its clock line ‘Active’ -> Acknowledge to master (10).

3. The above sequence is repeated for the following data bits.

4. The master ends the communication with a stop condition:
   - The master set its clock line ‘Active’ (12).
   - The slave set its clock line ‘Passive’ -> Acknowledge to master (13).
- The master set its data line ‘Active’ (14).
- The slave set its data line ‘Active’ -> Acknowledge to master (15).
- The slave set its clock line ‘Active’ -> The devices are now idle (16).
- Both devices are now idle and ready for a new start condition.

**Transmitting more than one byte**

To transmit more than one byte the following sequence is used:
1. Start condition
2. Transmitting byte #1 (MSB)
3. Transmitting byte #2
4. Transmitting byte #3
5. Transmitting byte #4 (LSB)
6. Stop condition

**Important Note:**

See diagram #1:

The slave can detect whether the master will transmit a next byte at (14).
- If the master set its data line ‘Active’ -> It is stop condition.
- If the master set its clock line ‘Passive’ -> It is the first bit of the following data byte.

**Synchronization**

The synchronization is mainly given by the clock lines. Both devices are always waiting for the clock state transition -> Acknowledge of the opposite device. This results in a dynamic adapting of the communication speed. The start and stop conditions are synchronized by the data line transitions.

**Timeout**

It is strongly recommended to implement two timeouts to avoid deadlocks:

1. The synchronization timeout. Whenever the master issues a line state transition it should start the acknowledge timeout of approximately 100ms. If the expected acknowledge can not be detected within this timeout the device must go into idle state and issue an error condition. The same applies for the slave if an acknowledge is expected.

2. The command response timeout. Whenever the master sends a command to the slave with an expected response it should start a response timeout of approximately 1000ms. If the expected response is not received within this timeout the command should be treated as failed and a error condition is returned to the application.
One Byte Transfer

Byte value: 0xA5 (10100101) = 165
Chapter 2: Vend sequence

Introduction
This chapter explains the logical communication between PLC and SPS2MDB interface.

Following payment devices are implemented:
- MDB Change Giver or MDB Coin Acceptor
- MDB Card Reader
- MDB Bill Acceptor (Optional)

The SPS2MDB interface offers a unique logical interface against to PLC independent of the connected payment devices. Therefore a simplified transaction procedure has to be defined. The current simplified implementation allows just a single vend. But future releases should also consider to implement a multi vend implementation.

Simplified vend procedure

<table>
<thead>
<tr>
<th>PLC Action</th>
<th>SPS2MDB Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>(a) Wait for a coin or card insert...</td>
<td>Credit available because a coin accepted.</td>
</tr>
<tr>
<td></td>
<td>'&lt;Begin Session', Credit: 0-65535 units'</td>
</tr>
<tr>
<td>(b) Wait for a product selection...</td>
<td>Vend Request, Price: 0-65535 units</td>
</tr>
<tr>
<td></td>
<td>'&lt;Vend Request'</td>
</tr>
<tr>
<td>(c) Deduct price of the credit...</td>
<td>Deduction success, confirm with 'Vend Approved'.</td>
</tr>
<tr>
<td></td>
<td>'&lt;Vend Approved'</td>
</tr>
<tr>
<td>(d) or...</td>
<td>Deduction failed, confirm with 'Vend Denied'.</td>
</tr>
<tr>
<td></td>
<td>'&lt;Vend Denied'</td>
</tr>
<tr>
<td></td>
<td>'&lt;Cancel Session'</td>
</tr>
<tr>
<td>Terminate transaction, back to (a)...</td>
<td></td>
</tr>
<tr>
<td>(e) Wait for end of vend...</td>
<td>Vend Success</td>
</tr>
<tr>
<td></td>
<td>'&lt;Cancel Session'</td>
</tr>
<tr>
<td>Terminate transaction, back to (a)...</td>
<td></td>
</tr>
<tr>
<td>(f) or...</td>
<td>Vend Failure</td>
</tr>
<tr>
<td></td>
<td>'&lt;Cancel Session'</td>
</tr>
<tr>
<td>Terminate transaction, back to (a)...</td>
<td></td>
</tr>
</tbody>
</table>

Note: Units define the lowest currency value like 'Cents'.
Chapter 3: Commands

Each command can be sent either with the SPS interface (2 Inputs / 2 Outputs) or over the serial RS-232 monitor interface. The contents and functions are equal. If a command is sent over the serial interface it must be placed inside a frame.

The parameters for this Interface are: 9600 Baud, 8 Data bit, 1 Stop bit, No Parity, No handshake.

The command frame:

| STX (0x02) | Length (3/6) | Command [Data] |

See commands below for the commands and optional data to place inside the frame.

If we look at the status request below for example, the serial command would be:

```
---------- 0x02 - 0x01 - 0xA0 -------- (3 bytes)
```

The response is then:

```
<---------- 0x02 - 0x01 – 0x00 ---------- (3 bytes)
```

Note: The disadvantage of using the serial interface it that the monitor for debugging and configuration is not available any more.

Command: Status Request (0xA0)

The command ‘Status Request’ returns the current status of the SPS2MDB interface.

Transmit: 1 Byte

<table>
<thead>
<tr>
<th>PLC</th>
<th>Action</th>
<th>SPS2MDB</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>‘Status Request’ (0xA0) ---------&gt;</td>
<td></td>
</tr>
</tbody>
</table>

Receive: 1 Byte

<table>
<thead>
<tr>
<th>PLC</th>
<th>Action</th>
<th>SPS2MDB</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>‘Status’ (Status) ----------</td>
<td></td>
</tr>
</tbody>
</table>

| Status: | Bit 0: | ‘L’ = Disabled  
  |        | ‘H’ = Enable  |
|---------|--------|---------------|
| Bit 1:  | ‘L’ = No Session  
  |        | ‘H’ = Active Acceptor Session  |
| Bit 2:  | ‘L’ = No Session  
  |        | ‘H’ = Active Reader Session  |
| Bit 3:  | Reserved, Always ‘L’ |
| Bit 4:  | Reserved, Always ‘L’ |
| Bit 5:  | ‘L’ = No Vend  
  |        | ‘H’ = Vend in process  |
| Bit 6:  | Reserved, Alwasy ‘L’ |
| Bit 7:  | ‘L’ = Okay  
  |        | ‘H’ = Error, command could not be proccessed.  |

Command: Enable (0xA1)

The command ‘Enable’ enables the connected payment systems to accept coins or cards.

Transmit: 1 Byte

<table>
<thead>
<tr>
<th>PLC</th>
<th>Action</th>
<th>SPS2MDB</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>‘Enable’ (0xA1) ---------&gt;</td>
<td></td>
</tr>
</tbody>
</table>

Receive: 1 Byte

<table>
<thead>
<tr>
<th>PLC</th>
<th>Action</th>
<th>SPS2MDB</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>‘Status’ (Status) ----------</td>
<td></td>
</tr>
</tbody>
</table>
See ‘Status Request’.

**Command: Disable (0xA2)**
The command ‘Disable’ disables the connected payment systems to reject any coins or cards.

**Transmit: 1 Byte**

<table>
<thead>
<tr>
<th>PLC</th>
<th>Action</th>
<th>SPS2MDB</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>‘Enable’ (0xA2)</td>
<td></td>
</tr>
</tbody>
</table>

**Receive: 1 Byte**

<table>
<thead>
<tr>
<th>PLC</th>
<th>Action</th>
<th>SPS2MDB</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>‘Status’ (Status)</td>
<td></td>
</tr>
</tbody>
</table>

See ‘Status Request’.

**Command: Begin Session (0xA3)**
The command ‘Begin Session’ informs the PLC that there is some credit available and a vend sequence can be accomplished.

**Note:** If an Acceptor is connected a ‘Begin Session’ is send as soon as the first coin is accepted. If any further coins are inserted ‘Begin Session’ is send again with the new total amount available. The command ‘Disable’ can be used to reject any coins during the vend process.

**Transmit: 4 Byte**

<table>
<thead>
<tr>
<th>PLC</th>
<th>Action</th>
<th>SPS2MDB</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>‘Begin Session’ (0xA3, Price list, Credit)</td>
<td></td>
</tr>
</tbody>
</table>

**Price List:**

<table>
<thead>
<tr>
<th>Bit 0..3:</th>
<th>Price list No. 0..15 (Optional: With card reader only)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bit 4..7:</td>
<td>Not used, always ‘L’</td>
</tr>
</tbody>
</table>

**Credit:**

| Bit 0..15: | 0..65535 units (2 Bytes) |

**Receive: 1 Byte**

<table>
<thead>
<tr>
<th>PLC</th>
<th>Action</th>
<th>SPS2MDB</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>‘Status’ (Status)</td>
<td></td>
</tr>
</tbody>
</table>

See ‘Status Request’.

**Command: Vend Request (0xA4)**
The consumer has pressed a product selection button and the price is known at this moment. The PLC request to deduct the price of the credit.

**Transmit: 4 Byte**

<table>
<thead>
<tr>
<th>PLC</th>
<th>Action</th>
<th>SPS2MDB</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>‘Vend Request’ (0xA4, Selection, Price)</td>
<td></td>
</tr>
</tbody>
</table>

**Selection:**

<table>
<thead>
<tr>
<th>Bit 0..7:</th>
<th>Selected product number: 0..255 (1 Byte)</th>
</tr>
</thead>
</table>

**Price:**

| Bit 0..15: | Price of the selected product: 0..65535 units (2 Bytes) |

**Receive: 1 Byte**

<table>
<thead>
<tr>
<th>PLC</th>
<th>Action</th>
<th>SPS2MDB</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>‘Status’ (Status)</td>
<td></td>
</tr>
</tbody>
</table>

See ‘Status Request’.

**Command: Vend Approved (0xA5)**
The requested price has been deducted of the credit and the PLC can start a vend sequence.
Transmit: 1 Byte
PLC  Action  SPS2MDB
<-------- 'Vend Approved' (0xA5) -------->

Command: Vend Denied (0xA6)
The requested price could not be deducted of the credit and a vend is not possible.
Transmit: 1 Byte
PLC  Action  SPS2MDB
<-------- 'Vend Denied' (0xA6) -------->

Command: Vend Success (0xA7)
The vend sequence successfully finished.
Note: Use ‘Cancel Session’ to payout the change or inform the consumer to remove the inserted card.
Or send another ‘Vend Request’ to start a new vend cycle.
Transmit: 1 Byte
PLC  Action  SPS2MDB
---------- 'Vend Success' (0xA7) -------->
Receive: 1 Byte
PLC  Action  SPS2MDB
<-------- 'Status' (Status) -------->
See ‘Status Request’.

Command: Vend Failure (0xA8)
The vend sequence failed. Undo the complete transaction as far as possible.
Transmit: 1 Byte
PLC  Action  SPS2MDB
---------- 'Vend Failure' (0xA8) -------->
Receive: 1 Byte
PLC  Action  SPS2MDB
<-------- 'Status' (Status) -------->
See ‘Status Request’.

Command: End Session (0xA9)
The current session has finished.
Transmit: 1 Byte
PLC  Action  SPS2MDB
<-------- 'End Session' (0xA9) -------->

Command: Cancel Session (0xAA)
Cancel the current session.
Note: If an Acceptor is connected, this command will payout any remaining amount. If a Reader is connected, its behavior depends on the specific reader implementation.
Transmit: 1 Byte

<table>
<thead>
<tr>
<th>PLC</th>
<th>Action</th>
<th>SPS2MDB</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>'Cancel Session' (0xAA)</td>
<td>--------</td>
</tr>
</tbody>
</table>

Receive: 1 Byte

<table>
<thead>
<tr>
<th>PLC</th>
<th>Action</th>
<th>SPS2MDB</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>'Status' (Status)</td>
<td>--------</td>
</tr>
</tbody>
</table>

See ‘Status Request’.

**Command: Device Reset (0xAB)**

Reset the active MDB engines. This is just a software reset and will reinitialize the MDB configuration.

Transmit: 1 Byte

<table>
<thead>
<tr>
<th>PLC</th>
<th>Action</th>
<th>SPS2MDB</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>'Device Reset' (0xAB)</td>
<td>--------</td>
</tr>
</tbody>
</table>

Receive: 1 Byte

<table>
<thead>
<tr>
<th>PLC</th>
<th>Action</th>
<th>SPS2MDB</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>'Status' (Status)</td>
<td>--------</td>
</tr>
</tbody>
</table>

See ‘Status Request’.
Chapter 4: Application Specific Functionality

Introduction:
Here are some hints for supporting the implementation of the custom specific application.

Setup Of Prices
The prices for the different product selections must be managed by the PLC. If a closed system debit card reader is connected it is possible to implement a set of price lists. The price list control can be accomplished by the price list parameter of ‘Begin Session’.

Multi Vend
Multi vend functionality is not implemented in the current version. Multi vend allows several vend sequences within one session, e.g. inserting a big valued coin and make a few selections before ending the session. Here an additional action is required to end the session, like pressing the reject button of the change giver.
Chapter 5: Hardware description

SPS2MDB Technical Data

<table>
<thead>
<tr>
<th>Specification</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Power supply:</td>
<td>24VDC regulated +/-20%</td>
</tr>
<tr>
<td>Power consumption:</td>
<td>Average: 70 mA, Max. 120 mA</td>
</tr>
<tr>
<td>Temperature range:</td>
<td>-20°C to +50°C celsius</td>
</tr>
<tr>
<td>Dimensions:</td>
<td>W x L x H: 18 x 90 x 60 mm</td>
</tr>
<tr>
<td>Case material:</td>
<td>PC 30% GF UL 94-V1</td>
</tr>
<tr>
<td>Protection:</td>
<td>IP 20</td>
</tr>
<tr>
<td>Weight:</td>
<td>65 Grams</td>
</tr>
<tr>
<td>Mounting:</td>
<td>On standard 35mm DIN rail</td>
</tr>
<tr>
<td>Max. load per output:</td>
<td>200 mA</td>
</tr>
<tr>
<td>Input current:</td>
<td>Typical 20 mA</td>
</tr>
</tbody>
</table>

SPS2MDB to PLC Interconnection

Figure 2
Chapter 6: Software Download

Connecting Hyper Terminal

The SPS2MDB includes a RS-232 interface to connect Microsoft HyperTerminal. The parameters for this Interface are: **9600 Baud, 8 Data bit, 1 Stop bit, No Parity, No handshake**.

Hyper Terminal setup:

![Hyper Terminal setup screenshot](image)

RS-232 interface cable for Microsoft Hyper Terminal:

![RS-232 interface cable](image)
Monitor

The purpose of this monitor interface is:
- Gathering of information like version, settings etc.
- Monitoring errors.
- Debugging of the SPS2MDB application.
- Downloading new releases of the SPS2MDB application.

The monitor presents a screen as below:

To enter the monitor, press any key in the first 3 seconds after the SPS2MDB interface is powered.
After the 3 seconds has expired it will start the SPS2MDB application.
To see a list of monitor commands enter a `?`.

Enter monitor:
1. Connect a serial port of your PC to the SPS2MDB interface.
2. Start Microsoft Hyper Terminal. Make sure for a correct setup of the parameters.
3. Make sure Hyper Terminal is connected. (The clock in the left bottom corner is running.)
4. Power on the SPS2MDB interface.
5. Press any button within 3 seconds.

The Monitor should show up as in the print screen above.

Download of a new software release

To download a new software release the monitor must be active.
Select menu option `Transfer` -> `Send Text File...`, select the file for download and Open. This will automatically start the download. The contents of the file must be an Intel Hexadecimal Object file with extension `*.HEX`. 
Chapter 7: Setup of a Test Application

To setup a test application, two SPS2MDB interfaces can be used. One will run the SPS2MDB application to communicate with the MDB devices on one side and the SPS/PLC on the other side. The other will run the SPS simulator application and communicate through the 4 digital I/O lines. The resulting software and communication model is shown below.

Figure: 4

The test application allows it to simulate with real equipment like an MDB change giver or a MDB card reader the complete application. It helps to get the knowledge for implementing the required software on a real SPS/PLC.

The following test sequences can be used to test the communication model above and also the final implementation with a real SPS/PLC.

Test sequences for SPS/PLC communication commands

Status
- Status request.

Enable / Disable
- Device enable, coins / cards are accepted.
- Device disable, coins / cards are rejected.

Enter session
- Device enable.
- Insert a coin / card -> enter session.
- Cancel session -> payout of coin / remove card.
- Device disable.

**Enter vend / Vend denied**
- Device enable.
- Insert a coin / card, enter session.
- Issue vend request.
- Receive vend denied.
- Cancel session -> payout of coin / remove card.
- Device disable.

**Enter vend / Vend failed**
- Device enable.
- Insert a coin / card, enter session.
- Issue vend request.
- Receive vend approved.
- Vend failed.
- Cancel session -> payout of coin / remove card.
- Device disable.

**Enter vend / Vend success**
- Device enable.
- Insert a coin / card, enter session.
- Issue vend request.
- Receive vend approved.
- Vend success.
- Cancel session -> payout of change / remove card.
- Device disable.

**Enter vend / Vend approved (No further coins accepted)**
- Device enable.
- Insert a coin / card, enter session.
- Device disable.
- Issue vend request.
- Receive vend approved.
- Vend success.
- Cancel session -> payout of change / remove card.
- Device enable.
  (This sequence not be suitable for card reader.)

**Escrow (Acceptor only)**
- Device enable.
- Insert a coin -> enter session.
- Escrow -> payout of coin.
- Device disable.

**Remove card (Reader only)**
- Device enable.
- Insert a card -> enter session.
- Remove card -> exit session.
- Device disable.

**Allowed commands within certain states**

<table>
<thead>
<tr>
<th>Command / State</th>
<th>Reset</th>
<th>Idle</th>
<th>Enabled</th>
<th>Session</th>
<th>Vend</th>
</tr>
</thead>
<tbody>
<tr>
<td>A0 – Status</td>
<td>Error</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Command</td>
<td>Error</td>
<td></td>
<td></td>
<td></td>
<td>Error</td>
</tr>
<tr>
<td>-----------------</td>
<td>-------</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>-------</td>
</tr>
<tr>
<td>A1 – Enable</td>
<td>Error</td>
<td></td>
<td></td>
<td></td>
<td>Error</td>
</tr>
<tr>
<td>A2 – Disable</td>
<td>Error</td>
<td>Error</td>
<td></td>
<td></td>
<td>Error</td>
</tr>
<tr>
<td>A4 – Vend Request</td>
<td>Error</td>
<td>Error</td>
<td>Error</td>
<td></td>
<td>Error</td>
</tr>
<tr>
<td>A7 – Vend Success</td>
<td>Error</td>
<td>Error</td>
<td>Error</td>
<td>Error</td>
<td>Error</td>
</tr>
<tr>
<td>A8 – Vend Failure</td>
<td>Error</td>
<td>Error</td>
<td>Error</td>
<td>Error</td>
<td>Error</td>
</tr>
<tr>
<td>AA – Cancel Session</td>
<td>Error</td>
<td>Error</td>
<td>Error</td>
<td>Error</td>
<td>Error</td>
</tr>
<tr>
<td>AB – Reset</td>
<td>Error</td>
<td></td>
<td></td>
<td></td>
<td>Error</td>
</tr>
<tr>
<td>Escrow</td>
<td>Error</td>
<td>Error</td>
<td>Error</td>
<td>Error</td>
<td>Error</td>
</tr>
<tr>
<td>Remove card</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Chapter 8: Appendixes

A: Figures
Diagram #1: Diagram of a One Byte Transfer
Figure 1: Electrical Connection.
Figure 2: SPS2MDB to PLC Interconnection.
Figure 3: RS-232 interface cable for Microsoft Hyper Terminal.
Figure 4: SPS2MDB Software & Communication Model.

B: Document History
20.02.06 1.00 First implementation of SPS communication.
28.08.06 1.01 Final modifications of SPS communication.
14.10.06 1.02 - Implementation of coin acceptor.
- Implementation of debit card reader.
02.04.08 1.10 Added serial commands via RS-232 monitor interface.

C: Glossary
LSB Last Significant Byte.
MDB Multi Drop Protocol.
MSB Most Significant Byte.
OSI Open Systems Interconnection model.
PLC Programmable Logical Controller.
SPS Speicher Programmierbare Steuerung.
VMC Vending Machine Controller.