友宝自动售货机软件技术规范 v3.1.1

北京友宝科斯科贸有限公司

友宝自动售货机软件技术规范

v3.1.1

北京友宝科斯科贸有限公司

联系方式

北京市朝阳区亮马桥路39号第一上海中心C座5层
Tel: +86 4001-528-528

http://www.ubox.cn

目 录

编写及评审记录

修订日期	修订者	评审者	修订内容
2012年7月11日	吕进华	黄荣辉，李新阳，邱乐文、宋纪正等	第一次评审
2012年7月16日	吕进华	李新阳，宋纪正等	第二次评审
2012年7月17日	吕进华	黄荣辉、邱乐文等	v3.0版发布
2012年7月30日	吕进华		v3.1草案
2012年8月25日	吕进华	黄荣辉，李新阳，宋纪正	v3.1.1

版本修订记录

v3.1.1

勘误：

- PAYOUT_IND 中，删除“注：只有PAYOUT_IND发起的扣款，才需要发送PAYOUT_RPT！”
- ACTION_RPT 中，cost 和 total_value、value 等金额相关字段改为2个字节长度
- INFO_RPT.type=16 中“前六种纸币币种相当于多少个纸币基数” 改为 “前八种纸币币种相当于多少个纸币基数”
- INFO_RPT.type=17 中“前六种纸币币种相当于多少个硬币基数” 改为 “前六种硬币币种相当于多少个硬币基数”
- COST_RPT 中“如果压钞失败，则需要先后发送PAYIN_RPT（DT=101）和 ACTION_RPT（TYPE=4）” 改为 “如果压钞失败，则需要先后发送PAYIN_RPT（DT=101）和 COST_RPT”
- HUODAO_SET_IND中“注意：对于支持货道商品售空检测的机型，PC一般不发送HUODAO_SET_IND 消息”改为“注意：不支持货道商品个数设置的机型，VMC 收到 HUODAO_SET_IND 后，无需处理该消息，直接回复ACK_RPT 即可”
- OFFLINE_DATA_RPT 中“每相邻的7字节表示一笔投币记录，字节1表示实际货道，字节2表示出货扣款”改为“每相邻的7字节表示一笔投币记录”
- OFFLINE_DATA_RPT 中“只有VMC无法再存储type=0的离线数据，才将离线数据统计到type=1”改为“只有最大离线数据容量，VMC无法再存储type=0的离线数据，才将离线数据统计到type=1”
- COST_IND 中“是否真正扣款成功，PC需要查看COST_RPT消息，正常情况下，10秒内就能收到 COST_RPT消息。扣款失败会收到ACTION_RPT.TYPE=4” 改为 “是否真正扣款成功，PC需要查看 COST_RPT消息，正常情况下，10秒内就能收到COST_RPT消息。”
- STATUS_RPT 中“注意：如果VMC自动检测到某个设备状态从“故障”到“正常”，或从“正常”到“故障”，需要主动发送STATUS_RPT一次 ”改为“注意：如果VMC自动检测到某个设备状态发生变化，需要主动发送STATUS_RPT一次”

新增和变更：

- 删除“系统初始化流程”
- 删除 STARTUP_RPT 和 INITIAL_OK 两条消息
- 删除 ACTION_RPT type=3
- 删除 ACTION_RPT type=4
- 变更：POLL 消息在“离线模式”延时5秒，在“在线模式”延时0.5 秒
- 变更：蜂鸣器“滴”一声为0.2秒，“滴滴”两声为0.4秒，以此类推
- 变更：“恢复出厂默认”，在v3.2(草案)中需要改变行为
- 变更：对《VMC“暂停服务”》下，vmc_st的取值详解
- 变更：INFO_RPT.type=15中，“VMC硬件型号”改为“VMC硬件唯一标识”

- 新增： GET_SETUP 消息
- 新增：“现金投入/支出统计”
- 新增：“售货机维护模式菜单->区间销售统计”中，新增“硬币投入、纸币投入、硬币出币”
- 新增：“售货机维护模式菜单”中，新增“每一次维护键盘按键，均快速‘滴’一声”
- 强调：“暂停服务”模式下，不处理展示位按键、小键盘、游戏按键，但需要处理退币按键

v3.1（草案）

- 所有金额相关字段确定为2字节
- CONTROL_IND的F7位变为1
- PAYIN_RPT的F7位变为0
- CONTROL_IND及INFO_RPT新增节能、工作、照明三个时间段
- VENDOUT_RPT新增huodao字段
- 不再使用ACTION_RPT.action=4
- 新增小液晶屏提示说明
- 新增消息响应说明
- 新增维护模式说明

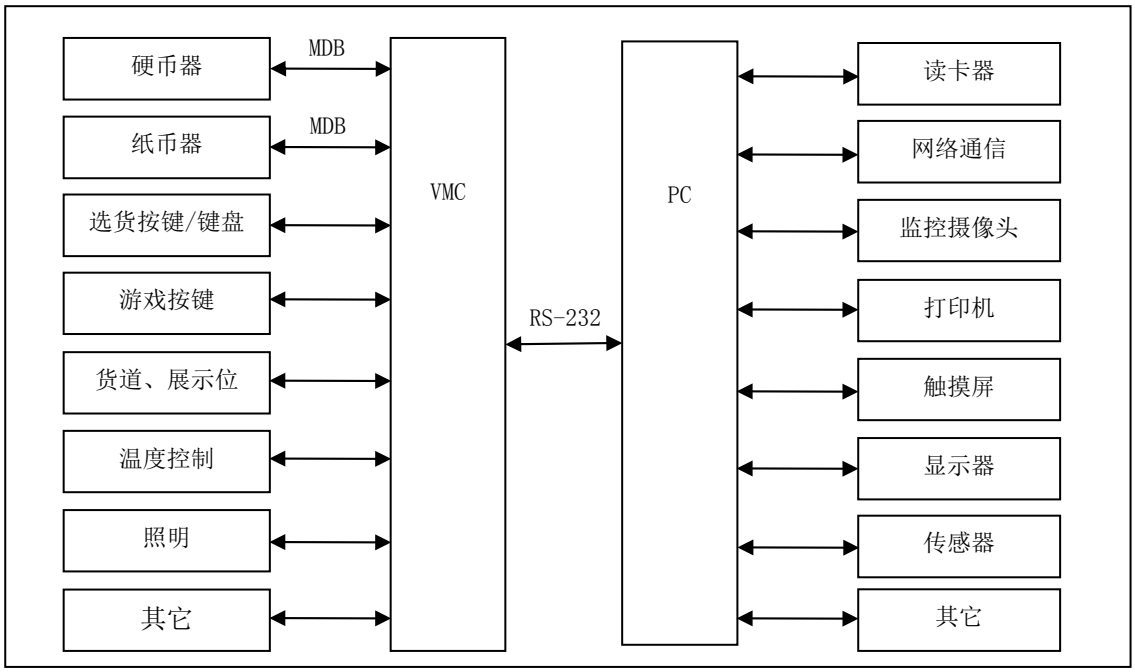
v3.0

- 离线售卖模式
- 使用新的传输层校验和算法
- VMC与PC失去连接，消息连续超时从10次改为3次
- CONTROL_IND新增type=2/6/7/8/9
- 多国货币支持
- 定义用户投币及找零规则
- VMC_SETUP新增4字节内容，用于描述机器支持特性
- 规定VENDOUT_IND.type=0为现金购物，同时规定type大于0，cost必须为0

概述

本文档描述友宝自动售货机工作原理，VMC与PC间的接口协议、默认行为及相关注意事项。

友宝自动售货机组成原理如下图所示：



友宝自动售货机模组图

第一部分 术语和定义

1. 自动售货机Vending Machine（VM）

一种自助设备，可以7x24小时方便快捷地提供各类饮料、食品及小商品的售卖服务。

2. 下位机Vending Machine Controller（VMC）

自动售货机中央控制单元，用于管理自动售货机各个物理模组，使各个模组之间可以相互配合，完成自动售货机定时制冷、自动加热、智能照明、现金收银、商品出货、销售统计等工作。

3. 上位机Programmable Computer（PC）

友宝自动售货机特有的模组，用于完成网络通信，多媒体回放，游戏，打印，读卡等功能复杂，占用CPU资源较多，实时性要求较低的功能。

4. 货道（Huodao）

又叫料道。是售货机中用于存放商品的通道，其放置于保温的货仓里面。

5. 展示位（Position）

包含展示商品/展示图片的摆放位置，和对应的选货按键。

透明箱体机型没有展示位。

第二部分 主要模组

1. 传统自动售货机模组

整机箱体、照明、出货检测、商品展示和储存、制冷/加热、现金收银和找零、小LED/液晶屏、维护键盘。

2. 友宝特有模组

大尺寸触屏（横屏或竖屏）、PC、游戏按键、网络通信、打印机、读卡器、监控摄像头、传感器。

第三部分 通信协议

1. 概述

VMC与PC直接采取主/从的方式进行通信。其中VMC主，PC从。VMC将以500ms的周期轮询PC。

本协议使用大端字节序，如：0x13546799，在串口上，先发送0x13，再发送0x54，0x67，最后发送0x99。

注意：

1. 如无特殊说明，本文所有整数都是“无符号整数”。

2. 本文“硬币器”特指包含硬币收币及硬币找零功能的设备。

3. VMC需要关注自己的非易失存储器的写次数限制。对于 HUODAO_IND、POSITION_IND、SALEPRICE_IND 等消息，VMC需要自动判断，如果无变化，无需写入非易失存储器。

2. 约定

2.1 金额换算

除非特殊说明，本文中所有涉及金额，均进行一定倍数转换。详见《VMC_SETUP》公式1和公式2。

2.2 协议版本

如下约定：

- 遵循v3规范的机型，必须完整实现版本Ver=3的所有消息及默认行为、附录A内容
- VMC支持的最高Ver版本，第一条消息Ver确定
- VMC如果收到比自己支持的Ver版本更高的消息，则丢弃
- VMC发出的所有消息，Ver须完全相同
- PC需要根据VMC的Ver版本判断如何处理消息

3. 物理接口

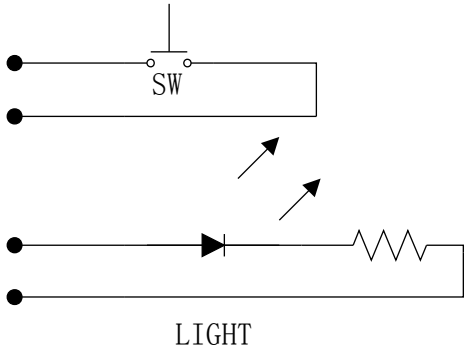
3.1 VMC与PC的连接

本协议不限定VMC与PC之间具体连接方式：

- 如果使用RS-232，有如下约定：
  - 只需要连接 GND、发送、接收 三根线
  - 波特率：9600
  - 数据位：8位
  - 停止位：1位
  - 奇偶位：无

- bit位顺序：｜ST｜b0·····b7｜SP｜

3.2 游戏按键和游戏按键灯



游戏按键内部示意图

游戏按键内部示意图如上图所示：

- 其中开关（SW）处于常开状态，按下闭合，松手自动打开
- 按键指示灯（LIGHT）已经自带限流电阻，正常工作电压直流9~12V

4. 传输层帧格式

传输层帧格式

VMC-PC使用二进制协议，为了实现接口数据的传输，定义如上图所示传输层帧格式；对于每一次的消息传输，双方均根据该帧格式进行消息编解码，各字段详解如下：

“**开始标记（SF）**”，Start Flag，SF使用十六进制数0xE7表示。

“**消息长度（LEN）**”，Length，用于表示其后所有内容的字节数，LEN占用1个字节，最小值为5（数据域长度为0），最大值为255（数据域长度为250）。

“**版本号（Ver）**”占最低3个bit位，故最高可表示0-7共8个协议版本。

- 消息的版本，VMC与PC需要根据版本号判断如何解析消息数据域内容
- v3协议支持两个版本的协议：
    - Ver=0、1、2：由于历史原因，0/1/2代表同一个协议版本（后续用Ver<3表示）
    - Ver=3：本协议最新版（后续用Ver=3表示）

“**确认（F7）**”占最高1个bit位

- 如果F7值为0，表示消息接收方不需要响应发送方，是否收到并正确处理该消息。
- 如果F7值为1，表示消息接收方需要响应发送方，是否正确处理该消息。
- 提示：简单理解，如果F7为0，相当于TCP/IP中的UDP消息；如果F7为1，相当于TCP/IP中的TCP消息。

“**消息类型（MT）**”，Message Type，表示该帧实际代表那条消息。

“**序列号（SN）**”，Serial Num，主要用于标识是否是相同的消息；无符号整数，超过255后归0。

“**数据域（datas）**”，每个MT包含一定的数据内容，长度从0-250字节不等。

“**校验和（CHK）**”，接收方用于验证接收到的数据是否正确，MT占用2个字节，CHK的值来自“头部”和datas，不包括CHK本身；

Ver = 3版本生成校验和方式：

```
unsigned short calc_crc(unsigned char *msg, unsigned short len)
{
   unsigned short i, j;
   unsigned short crc = 0;
   unsigned short current;
   for (i = 0; i < len; i++)
   {
      current = msg[i] << 8;
      for (j = 0; j < 8; j++)
      {
         if ((short)(crc ^ current) < 0)
            crc = (crc << 1) ^ 0x1021;
         else
            crc <<= 1;
         current <<= 1;
      }
   }
   return crc;
}
```

- 举例：

SF	LEN	F7+Ver	MT	SN	datas	CHK
E7	0A	83	06	00	00 0064 0064	a319

- 举例：

SF	LEN	F7+Ver	MT	SN	datas	CHK
E7	05	03	01	02	none	aabe

注意：如果接收端数据，发现SF/LEN/VER等字段不正确，则直接丢弃消息，不给发送端任何响应

5. 消息列表

默认约定：

- 以“_RPT”结尾的消息，均来自VMC
- 以“_IND”结尾的消息，均来自PC
- 以“GET_”开头的消息，均来自PC
- VMC是否响应ACK_RPT/NAK_RPT的唯一依据是F7位，任何消息（包括HUODAO_IND等），如果F7=0，则VMC不需要响应
- PC是否响应ACK/NAK的原则与VMC相同

5.1 VMC->PC

注： Ver=3版本无需实现 TEXT_RPT/STARTUP_RPT/CARD_RPT/HUODAOCONF_RPT消息

〈MT〉	〈F7〉	〈Message〉	〈简介〉
~~0x00(0)~~	~~0~~	~~TEXT_RPT~~	~~(保留)~~
0x01(1)	0	ACK_RPT	VMC 确认成功
0x02(2)	0	NAK_RPT	VMC 确认失败
0x03(3)	1	POLL	轮询
~~0x04(4)~~	~~1~~	~~STARTUP_RPT~~	~~VMC 发起初始化~~
0x05(5)	1	VMC_SETUP	VMC 系统参数
0x06(6)	0	PAYIN_RPT	投币报告
0x07(7)	1	PAYOUT_RPT (*)	出币报告
0x08(8)	1	VENDOUT_RPT	出货报告
0x09(9)	1	REQUEST	VMC 向 PC 请求数据
0x0A(10)	1	ADMIN_RPT	维护菜单功能
0x0B(11)	0	ACTION_RPT	动作报告
0x0C(12)	0	BUTTON_RPT	游戏按键报告，或商品选择报告等
0x0D(13)	0	STATUS_RPT	VMC 整机状态报告
0x0E(14)	0	HUODAO_RPT	VMC 货道状态报告
~~0x0F(15)~~	~~0~~	~~CARD_RPT~~	~~(保留)~~
0x10(16)	1	COST_RPT	扣款报告
0x11(17)	0	INFO_RPT	数据报告
0x12(18)	1	VENDOUT_REQ	VMC 向 PC 出货请求
0x13(19)	1	OFFLNE_DATA_RPT	VMC 上报离线售卖数据
~~0x78(120)~~	~~1~~	~~HUODAOCONF_RPT~~	~~(保留)~~

*：如果是PAYOUT_IND发起的出币，则 F7=1；如果是VMC发起的出币，则 F7=0。

注意： VMC在等待PC响应的过程中，不能给PC再发送其他消息。

提示：除了VENDOUT_REQ和ADMIN_RPT，其他消息PC只能回复ACK。

5.2 PC->VMC

注：Ver=3版本无需实现 INITIAL_OK 消息

<MT>	<F7>	<Message>	<简介>
0x80(128)	0	ACK	PC 确认成功
0x81(129)	0	NAK	PC 确认失败
~~0x82(130)~~	~~1~~	~~INITIAL_OK~~	~~PC 确认系统初始化流程成功完成~~
0x83(131)	1	VENDOUT_IND	PC 出货指示
0x84(132)	0	RESET_IND	PC 发出指示，要求 VMC 重新启动
0x85(133)	1	CONTROL_IND	PC 控制 VMC
0x86(134)	0	GET_STATUS	PC 通知 VMC 发送 STATUS_RPT
0x87(135)	1	HUODAO_IND	PC 发送货道配置数据
0x88(136)	1	POSITION_IND	PC 发送展示位配置数据
0x89(137)	1	PAYOUT_IND	PC 退币指示
0x8A(138)	0	GET_HUODAO	PC 通知 VMC 发送 HUODAO_RPT
0x8B(139)	1	COST_IND	PC 扣款指示
0x8C(140)	0	GET_INFO	PC 通知 VMC 发送 INFO_RPT
0x8D(141)	1	SALETIME_IND	盒饭机专用：PC 发送售卖时间段指示
0x8E(142)	1	SALEPRICE_IND	商品售价指示
0x8F(143)	1	HUODAO_SET_IND	货道商品余量
0x90(144)	0	GET_SETUP	PC 通知 VMC 发送 VMC_SETUP
0x91(145)	0	GET_OFFLINE_DATA	PC 通知 VMC 发送 OFFLINE_DATA_RPT

6. 普通消息

6.1 ACK / NAK

功能：PC确认VMC的消息，本消息无消息体，即datas长度等于0。

Ver <= 3:

字段名	
字节数	0

其中：

- ACK用于通知VMC，这是一个成功的响应
- NAK用于通知VMC，这是一个失败的响应

注意：除了如下两条消息，其它消息PC肯定回复ACK，PC和VMC可以据此优化协议实现：

- VENDOUT_REQ（如果PC拒绝售卖，则回复NAK）
- ADMIN_RPT（如果配置数据出错、或全系统同步失败等，则回复NAK）

提示：VMC需要对ACK和NAK的序列号SN进行严格判断，如下：

<SN>	<Message>	<说明>
5	VENDOUT_RPT	

4	ACK	ACK 的 SN=4，不等于 VENDOUT_RPT 的 5，VMC 抛弃该消息，并继续等待，直到收到 SN=5 的 ACK 或超时
5	ACK	VMC 在有效时间内，收到了正确的 ACK 消息

6.2 ACK_RPT / NAK_RPT

功能：通知PC，这是一个成功/失败的响应

Ver <= 3:

字段名	
字节数	0

6.3 GET_SETUP

功能：通知VMC上发VMC_SETUP消息。本消息无消息体，即datas长度等于0。

Ver = 3:

字段名	
字节数	0

6.4 VMC_SETUP

功能： VMC向PC报告自己的系统参数。

Ver=3:

字段名	hd_num	pos_num	magic1	scale factor	decimal places	feature
字节数	1	1	2	1	1	4

hd_num：

- VM支持的货道数量
- 某些VM可能会将多个物理货道合并为一个，如果仅仅是钣金上的改动，则hd_num不受影响

pos_num：

- VM支持的展示位数量
- 需要上报实际展示位数量
- 某些机型没有展示位，则pos_num=0

注意：如果VMC上报的 HD_NUM、POS_HUM、MAGIC1 与PC的数据不一致， PC不回复VMC任何消息，此时VMC将等待超时；

magic1：

- 售货机机型标识
- 本标识根据不同厂商VM种类的不同而不同，由友宝统一分配

scale factor：

- VMC与PC之间金额换算倍数（The multiplier used to scale all monetary values transferred between the VMC and the PC.）
- 如果VM连接硬币器，则scale factor就是MDB协议中的Coin Scaling Factor；（一般，人民币scale factor是5，美元也是5）
- 如果VM没有连接硬币器，但连接了纸币器或其他支付设备，则scale factor为这些对应设备的scale factor
- 如果VM没有连接任何支付设备，则 scale factor默认为50

decimal places：

- 金额小数点位置
- 如果VM连接硬币器，则decimal places就是MDB协议中的Coin decimal places；（一般，人民币decimal places 是1，美元是2）
- 如果VM没有连接硬币器，但连接了纸币器或其他支付设备，则decimal places为这些对应设备的decimal places
- 如果VM没有连接任何支付设备，则decimal places默认为2；

公式1：$ActualValue = (ScaledValue * ScaleFactor) * 10^{-(DecimalPlaces)}$

公式1举例：假如VMC上报PC的投币额ScaledValue=9，则如果是人民币，ActualValue=9*5*0.1=4.5元；如果是美元，ActualValue=9*5*0.01=0.45美金

公式2：$ScaledValue = ActualValue / (ScaleFactor * 10^{-(DecimalPlaces)})$

公式2举例：假如PC的商品售价ActualValue=4.5，则如果是人民币，ScaledValue=4.5/(5*0.1)=9；如果是美元，ScaledValue =4.5/(5*0.01) =90

feature：

- VM支持特性列表（最大支持32个特性）,bitx=1表示支持，bitx=0表示不支持：
- bit0：是否支持PAYOUT_IND
- bit1：是否支持COST_IND
- bit 2：是否支持CONTROL_IND（type=2、6）和BUTTON_RPT（type=4）【这些功能需要同时支持】
- bit 3：是否支持CONTROL_IND（type=7）
- bit4：是否支持CONTROL_IND（type=8）
- bit 5：是否支持CONTROL_IND（type=9）
- bit 6：是否支持CONTROL_IND（type=17）和GET_INFO（type=5）
- bit 7：是否支持离线售卖
- bit 8：是否支持出货检测（注意：即使出货检测功能被手工禁用或VMC自动禁用，也认为支持）
- bit 9：是否支持total_value和GET_INFO（type=3）
- bit 10：是否支持GET_INFO（type=6）
- bit 11：是否是盒饭机

- bit 12：是否在1拖2状态
- bit 13：箱体1（主箱体），是否支持自动检测货道商品售空
- bit 14：箱体2（从箱体），是否支持自动检测货道商品售空（1拖2状态有效）
- bit 15：是否支持制冷
- bit16 ：是否支持加热
- 注意：未定义BIT固定为0。

6.5 POLL

功能：VMC对PC发起轮询。本消息无消息体，即datas长度等于0。

Ver <= 3:

字段名	
字节数	0

注意：VMC发起的轮询，时间间隔需要在500毫秒左右【离线模式时间间隔为5秒】

注意：如果POLL前后存在其他消息，则从那条消息开始记500MS

提示：PC需对POLL做好时间标记，超过1500ms的POLL消息直接丢弃

6.6 RESET_IND

功能：PC复位VMC

Ver <= 3:

字段名	dt
字节数	1

dt：device type

- 该字段永远为0

注意：VMC收到该消息后，需要自动对VM系统进行全系统复位（与初始上电相同的效果）

提示：VMC在收到RESET_IND消息需要复位，或其他原因VMC需要复位时，如果用户有现金投币，VMC需要主动进行退币，并记录退币情况。

7. 配置消息

7.1 HUODAO_IND

功能：PC通知VMC，当前货道对应商品ID

Ver = 3:

字段名	device	sp_id	…	sp_id
字节数	1	1	…	1

device：

- 固定为0

sp_id：货道商品ID

- sp_id取值范围为（1 .. n）
- sp_id字段的总个数等于VMC_SETUP消息中hd_num！

注1： 如果VMC收到的HUODAO_IND消息中，sp_id字段的总个数与自身物理货道数量不相同，则VMC抛弃该消息，并回复NAK_RPT。

注意：这里全部的SP_ID可以看做一个数组，数组的索引即为PC逻辑货道；VMC收到数据后，需要自动将PC的逻辑货道映射到自己的物理货道。参见《VMC物理货道和PC逻辑货道》

7.2 HUODAO_SET_IND

功能：PC通知VMC，当前货道对应商品的数量等信息。

Ver = 3:

字段名	device	huodao	…	huodao
字节数	1	1	…	1

device：

- 当前固定为0

huodao： zyxxxxxx

- “z”固定填0
- “y”固定填0
- “xxxxxx”，表示商品余量，如果商品余量大于63，则统一为63

注1： 如果VMC收到的HUODAO_SET_IND消息中，huodao 字段的总个数与自身物理货道数量不相同，则VMC抛弃该消息，并回复NAK_RPT。

注意：类似饮料机不支持货道商品个数设置的机型，VMC 收到 HUODAO_SET_IND 后，无需实际处理该消息，直接回复ACK_RPT

7.3 POSITION_IND

功能：PC通知VMC，当前展示位对应商品ID。

Ver = 3：

字段名	device	sp_id	…	sp_id
字节数	1	1	…	1

device：展示位对应商品的售价

- 固定为0

sp_id：货道商品ID

- 如果售货机包含有可配置的展示位，则sp_id字段的总个数等于VMC_SETUP消息中pos_num！

注1： 如果VMC收到的POSITION_IND消息中，sp_id 字段的总个数与自身物理展示位数量不相同，则VMC抛弃该消息，并回复NAK_RPT。

7.4 SALEPRICE_IND

功能：PC通知VMC，当前商品售价。

Ver = 3:

字段名	type	sp_price	…	sp_price
字节数	1	2	…	2

变长消息，最大商品个数不超过124个；

数组索引（1…n）代表商品id，直接对应HUODAO_IND和POSITION_IND中的sp_id。

type：

- 固定填0

sp_price：

- 商品售价（通过公式1和公式2计算）
- 关于商品售价等于0和 0xffff 的特性情况，参见《商品售价》

7.5 CONTROL_IND

功能：PC控制售货机完成对应的动作。

提示：对于VMC不支持的type类型，VMC直接丢弃，回复NAK_RPT。

7.5.1 type=2：现金收银模组（纸币器、硬币器）开关

Ver = 3:

字段名	type	value
字节数	1	1

- 如果value=0，表示关闭设备
- 如果value非0表示开启设备
- VMC初次启动或复位结束后，现金收银模组自动设置为默认的“开启”状态
- 在VMC“暂停服务”模式，可以通过该命令，强制开启和关闭纸硬币器的收币功能

7.5.2 type=6：找零（与手工拨弄物理找零开关相同）

Ver = 3:

字段名	type	value
字节数	1	1

- value固定为0

7.5.3 type=7：节能工作时间段设置

如果配置节能时间段，则只在配置的时间段内自动进行加热或制冷；否者全天都自动进行加热或制冷。

Ver = 3:

字段名	type	flag	Group1	Group2	Group3
字节数	1	2	28	28	28

- Group1-Group3：第一组到第三组，每组有7个工作时间段

字段名	ts1	ts2	ts3	ts4	ts5	ts6	ts7
字节数	4	4	4	4	4	4	4

ts1-ts7：每个时间段的“开始”和“结束”时间：

字段名	开始小时	开始分钟	结束小时	结束分钟
字节数	1	1	1	1

“小时”的范围：00-23；“分钟”的范围：00-59。

举例：配置时间“11：10至12：20”，那么对应的字段值：“0x0B 0x0A 0x0C 0x14”

注意：1、时间段不能跨天，即开始时间与结束时间一定要在00:00 – 23:59之间，如时间段“22:00 – 03:00”不正确，VMC回复NAK_RPT拒绝该条消息。

注意：2、如果两个时间段完全相同，表示无效时间段，如“22:00 – 22:00”，或“00:00 – 00:00”。如果FLAG指定的GROUP全是无效时间段，表示*全天都是非节能*状态。

注意：3、时间段可以重叠，如时间段1为“07:00 – 09:00”，时间段2为“05:00 – 15:00”。

注意：4、时间段不分先后，如时间段1为“07:00 – 09:00”，时间段2为“02:00 – 05:00”。

- flag

字段名	<保留>	六	五	四	三	二	一	日
bit 数	15 14	13 12	11 10	9 8	7 6	5 4	3 2	1 0

- flag表示一周7天，如何对应到Group1~Group3
  - * flag=00 没有对应任何Group
  - * flag=01 对应Group1
  - * flag=10 对应Group2
  - * flag=11 对应Group3
  - *<保留>字段固定为00

注意：如果FLAG没有对应任何GROUP，则表示所有GROUP均为节能工作时间段。

- VM收到有效时间段数据后，需要自动保存，掉电不丢失，离线售卖有效
- 如果VM收到的时间段数据格式有误，直接回复NAK_RPT

7.5.4 type=8：售卖时间段设置

如果配置售卖时间段，则只在指定的时间段内进行商品售卖，其他时间段暂停服务；如果没有配置，则全天都进行售卖。

注意：

1. type=8配置的售卖时间段之外的时间，不需要自动关闭制冷/制热和照明

2. 进入暂停服务，VMC自动关闭一次纸硬币器（非售卖时间，PC可以通过type=2来打开和关闭纸硬币器）

3. 如果flag配置的某个Group全是无效时间段，则表示*全天售卖*

Ver = 3:

字段名	type	flag	Group1	Group2	Group3
字节数	1	2	28	28	28

各个字段含义，同 type=7。

7.5.5 type=9：照明时间段设置

如果配置照明时间段，则只在配置的时间段内开启照明；否者全天都开启照明。

注意：

1. 如果flag配置的某个Group全是无效时间段，则表示*全天照明*

Ver = 3:

字段名	type	flag	Group1	Group2	Group3
字节数	1	2	28	28	28

各个字段含义，同 type=7。

7.5.6 type=16：游戏按键灯开关

Ver <= 3:

字段名	type	value
字节数	1	1

- 如果value=0，表示关闭设备
- 如果value非0表示开启设备
- 系统一旦重启/复位完成，游戏按键灯自动设置为默认的“关闭”状态

7.5.7 type=17：下发时间给VMC

PC每3分钟自动下发一次。

Ver <= 3:

字段名	type	年	月	日	时	分	秒	星期
字节数	1	2	1	1	1	1	1	1

- 星期：0是星期日，1是星期一，…，6是星期六

- 举例：2012年2月7日，10点41分13秒，星期二表示为：

2012	2	7	10	41	13	2
0x07DC	0x02	0x07	0x0A	0x29	0x0D	0x02

7.6 SALETIME_IND

功能：PC控制销售时间段。（盒饭机专用）

Ver <=3：

字段名	时间段 1	时间段 2	时间段 3	时间段 4	时间段 5
字节数	4	4	4	4	4

时间段x：

- 第一个字节：开始小时；第二个字节，开始分钟； 第三个字节：结束小时；第四个字节，结束分钟；

  举例：配置销售时间“11：10至12：20”，那么对应的字段值：“0x0B 0x0A 0x0C 0x14”

- 跨天时间段支持

  举例：配置销售时间“23：00至02：00”，那么对应的字段值：“0x17 0x00 0x02 0x00”

- 如果某时间段不进行工作，则默认全f（0xffffffff）

注1：5个销售时间段不能出现相互重叠，如果出现重叠，则直接回复NAK_RPT。

注2：如果时间段取值不正确（如小时字段出现25），则直接回复NAK_RPT。

注3：VMC收到该消息后，需要自动更新内部销售时间段配置信息。

8. 售货机行为

8.1 REQUEST

功能：VMC请求PC响应的数据。

Ver = 3:

字段名	type
字节数	1

type：指明请求的数据

- type=1：请求货道、展示位和商品售价数据（PC将配合POLL 自动发送HUODAO_IND 和 POSITION_IND、SALEPRICE_IND、HUODAO_SET_IND）
- type=2：请求PC发送当前时间（PC发送CONTROL_IND的type=17）
- type=3：请求PC发送销售时间段（PC发送SALETIME_IND）
- type=4：请求HUODAO_IND
- type=5：请求POSITION_IND
- type=6：请求SALEPRICE_IND

- type=7：请求HUODAO_SET_IND

注意：一般情况下，并不需要VMC请求这些数据，PC会周期性地发起同步。

8.2 OFFLINE_DATA_RPT

功能：VMC发送离线销售数据给PC。

type=0：

Ver = 3:

字段名	type	seconds	hd_id	cost	…
字节数	1	4	1	2	7n

datas:

- 每相邻的7字节表示一笔投币记录，字节1表示实际货道，字节2表示出货扣款
- 每条销售记录由seconds、hd_id、cost七个字节组成；
- 如果datas的字节长度不能被 7 整除或其他错误，则认为这是一条错误消息；PC不处理，也不响应ACK或NAK！
- VM一次最多发送20条记录

seconds:

- 销售记录产生的时间，根据“算法1”生成

hd_id:

- 商品出货的货道ID

cost:

- 商品出货扣款

type = 1：（只有最大离线数据容量，VMC无法再存储type=0的离线数据，才将离线数据统计到type=1）

Ver = 3:

字段名	type	value all	vendout all
字节数	1	2	2

value all:

- 总出货金额

vendout all:

- 总出货个数

注意：如果PC回复ACK，则VMC需要将RPT中携带的相关数据全部删除。

注意：VMC需要优先发送较早的数据。

8.3 PAYIN_RPT

功能：VMC发送现金投币报告给PC。

Ver = 3:

字段名	dt	value	total_value
字节数	1	2	2

dt：投币类型

- dt=0：硬币投币
- dt=1：纸币投币。纸币成功压钞到钱箱（暂存状态压钞成功也需要上报dt=1）。
- dt=100：纸币暂存入。纸币进入暂存状态
- dt=101：纸币暂存出。暂存的纸币被退出、或暂存的纸币压钞失败

value：

- 现金投币的面额

total_value：

- 用户现金投币余额
- 该值包含纸币+硬币，其中纸币包括“暂存”状态的纸币（简单理解，total_value就是小LED/LCD上显示的用户投币值）
- total_value的值包含当前投入的这笔现金金额

8.4 PAYOUT_IND

功能：PC指示VMC出币。

Ver = 3:

字段名	device	value	type
字节数	1	2	1

device：出币设备

- device=0：硬币出币（如果无对应设备，回复NAK_RPT）
- device=1：纸币出币（如果无对应设备，回复NAK_RPT）
- 其他device暂不支持

value：本次出币总金额

type：出币类型

- VMC无需理解type的含义，只需要在出币完成后的PAYOUT_RPT中将该type值回传给PC即可

VMC能判断无法出币，直接回复NAK_RPT的情况：

- value值不能通过“算法2”组合出来

- 无找零设备、或找零设备故障

注意：收到PAYOUT_IND均需要根据VALUE指示进行出币，该出币*不会*减少用户余额（退币过程中，小LED/LCD显示的用户金额不发生变化），如下所示：

<SN>	<Message>	<说明>
1	POLL	当前用户投币余额 5 元
1	PAYOUT_IND	PC 需要出币 3 元，指示 VMC
1	ACK_RPT	VMC 确认系统找零足够出币
…	<其他无关消息>	出币过程可能时间很长，这期间可能出现其他无关消息
5	PAYOUT_RPT	VMC 终于出币完成，报告结果给 PC
5	ACK	PC 确认收到该消息，此时用户投币余额依然是 5 元

注意：VMC如果判断出当前系统找零量小于PC出币需求，则需要返回NAK_RPT，如下所示：

<SN>	<Message>	<说明>
1	POLL	
1	PAYOUT_IND	当前系统找零量是 5 元，PC 要求出币 8 元
1	NAK_RPT	VMC 发现系统找零量 5 元小于出币需求 8 元，通知 PC 无法出币

注意：在PAYOUT_IND退币过程中，发生异常导致无法退出足额的VALUE金额，不需要特别处理，只需要在对应的PAYOUT_RPT中，上报实际出币金额即可。

提示：PAYOUT_IND与 CONTROL_IND中type=6的区别是：

**CONTROL_IND功能是退出用户余额，退币结束后用户余额变为0，作用与退币按钮退币相同；*

**PAYOUT_IND不会减少用户余额！*

8.5 PAYOUT_RPT

功能：VMC出币报告。

Ver = 3:

字段名	device	value	total_value	type
字节数	1	2	2	1

device：

- device=0：硬币出币
- device=1：纸币出币

value：

- 实际退币金额
- 注意：如果应退币金额大于0，但由于故障，导致实际退币金额为0，也需要发送PAYOUT_RPT！

total_value：

- 退币结束后，用户投币余额总额

type：

- 如果是用户手工退币（包括CONTROL_IND中type=6的情况），则type=0

- 如果是PAYOUT_IND发起的出币，则type=PAYOUT_IND中的type

8.6 VENDOUT_REQ

功能：VMC请求出货。售货机在每次出货前需要向pc进行出货请求，若是pc返回ACK，售货机继续出货；若是pc返回NAK，售货机停止出货流程。（盒饭机专用）

Ver = 3:

字段名	device	method	sp_id/hd_id	total_money
字节数	1	1	1	2

device：

- 同VENDOUT_IND.device

method：

- 同VENDOUT_IND.method

hd_id：

- 逻辑货道id

sp_id：

- 商品id

total_money：

- 同“PAYIN_RPT”total_money

注意：盒饭售货机在收到VENDOUT_IND消息时，不需要发送此出货请求

8.7 VENDOUT_IND

功能：PC出货指示。

Ver = 3:

字段名	device	method	sp_id/hd_id	type	cost
字节数	1	1	1	1	2

method：

- **method** =1：VMC通过商品ID指示出货，如果商品ID不存在，回复NAK_RPT
- **method** =2：VMC通过货道ID指示VMC出货，如果货道ID不存在，回复NAK_RPT
- 其它值直接回复NAK_RPT
- VMC两种出货方式都需要支持
- 注意：如果是METHOD=2，则VMC需要强制出货（也即即使VMC认为商品个数为0，也回复ACK_RPT，并发起出货动作）

device：

- 固定为0

sp_id：通过商品ID指示VMC出货

- VMC需要根据sp_id到货道表中查询对应的商品，由于货道表中多个货道可能配置相同的sp_id，则：
  - VMC需要对多个含有相同商品的货道轮流出货，不能总从一个货道出
  - VMC需要自己判断货道已经售空，从而自动跳过该货道，自动查找任然有存货的货道；如果所有货道都售空，则回复NAK_RPT

hd_id：通过货道ID指示VMC出货

- VMC需要根据sp_id到货道表中查询对应的商品，由于货道表中多个货道可能配置相同的sp_id，则：
  - VMC需要对多个含有相同商品的货道轮流出货，不能总从一个货道出
  - VMC需要自己判断货道已经售空，从而自动跳过该货道，自动查找任然有存货的货道；如果所有货道都售空，则回复NAK_RPT

type：

- VMC需在对应的VENDOUT_RPT中回填

cost：

- 如果type=0，cost 代表本次出货扣款金额，VMC需要先在用户投币余额中，扣除cost。
- **注意：如果TYPE不为0，则COST必须为0，如果违反该约定，则VMC需要回复NAK_RPT**
- **注意：如果TYPE=0且COST大于0，此VENDOUT_IND视为现金购物（如触屏购物）**

VMC能判断无法出货，直接回复NAK_RPT的情况：

- 整机“暂停服务”
- sp_id对应商品已经售完
- sp_id对应的货道全部故障
- 对于VENDOUT_IND中type不为0，但cost不为0
- 对于VENDOUT_IND中type=0且cost大于0，用户投币不足，或扣款后无法找零（参见《用户投币及找零》）

8.8 VENDOUT_RPT

功能：VMC出货报告。

Ver = 3:

字段名	device	status	hd_id	type	cost	total_value	huodao
字节数	1	1	1	1	2	2	1

device：

- VENDOUT_IND发起购物的情况，device=VENDOUT_IND.device

status：

- status=0，出货成功完成

- status=2，出货失败；出货失败的情况，参见《出货失败和货道故障》

hd_id：

- 实际出货货道
- 该hd_id指PC逻辑货道，VMC需要自动将物理货道映射到逻辑货道，参见《VMC物理货道和PC逻辑货道》

type：

- 如果是现金购物，则type=0
- 如果是VENDOUT_IND发起的出货，则type=VENDOUT_IND.type
- type的具体含义，参见《INFO_RPT（type=6）》

total_value：

- 出货完成后，用户投币余额

huodao：

- hd_id对应的货道中，剩余商品个数，与HUODAO_RPT中huodao字段含义相同

cost：

- 如果是非现金购物，cost=0
- 如果是现金购物，且status=0，cost代表用户购买商品的花费
- 注意：对于STATUS=2的出货失败情况，如果是现金购物（即TYPE=0），则 COST 字段表示*正确返还*给用户的金额，正常情况下COST应该与商品售价相同，如果COST等于0，说明本次出货失败没有给用户返还金额（比如找零量为0的情况下使用纸币进行的购买）；参见《用户投币及找零》。

8.9 COST_IND

功能：PC扣款指示。

Ver = 3:

字段名	device	value	type
字节数	1	2	1

device：

- device=0，从用户投币总额中扣款；优先从用户非暂存金额中扣除（纸币尽量滞后压钞），参见《现金扣款顺序》

value：

- 扣款金额

type：

- VMC不用理解type的含义，只需上报对应的COST_RPT时回传即可

VMC能判断无法扣款，直接回复NAK_RPT的情况：

- 用户投币余额小于value
- 参见《用户投币及找零》

注意：COST_IND实际扣款成功后，VMC需要实时自动更新展示位指示灯。

注意：是否真正扣款成功，PC需要查看COST_RPT消息，正常情况下，10秒内就能收到COST_RPT消息。

8.10 COST_RPT

功能：VMC扣款报告。

注：只有COST_IND发起的扣款，才需要发送COST_RPT！

Ver = 3:

字段名	device	value	total_value	type
字节数	1	2	2	1

device：

- 实际扣款方式，与COST_IND中的device相同。

value：

- 实际成功扣款金额；
- 注意：如果由于故障，导致实际扣款金额为0，也需要发送COST_RPT

total_value：

- 与PAYIN_RPT中total_value含义相同

type：

- 与COST_IND中的type相同。

注意：如果 COST_IND 扣款需要暂存状态的纸币压钞：

*如果压钞成功，则需要先后发送PAYIN_RPT（DT=1）和 COST_RPT；

*如果压钞失败，则需要先后发送PAYIN_RPT（DT=101）和 COST_RPT

8.11 ADMIN_RPT

功能：VMC按键报告。

Ver <= 3:

字段名	type	datas
字节数	1	0 … n

type：

8.11.1 type=1，加满全部货道

* datas=null

* PC正常执行加满后，会给VMC直接回复HUODAO_SET_IND，否者回复NAK

8.11.2 type=2，货道存货数量修改

* datas[0]=货道ID，datas[1]=输入的数量

* 注意：VMC在收到PC的ACK响应后，需要自己主动自动修改VMC对应货道的数量；如果收到NAK响应，则无需修改。

8.11.3 type=3，补硬币完成

* datas=null

8.11.4 type=4，全系统同步

* datas=null

8.11.5 type=5，VMC开门

* datas=null

* 注意：开门不作为进入维护模式的判断依据

8.11.6 type=6，VMC关门

* datas=null

8.11.7 type=8，按层加满

* datas[0]=层号

* PC正常执行加满后，会给VMC直接回复HUODAO_SET_IND，否者回复NAK

8.11.8 type=9，取纸币完成

* datas=null

注意：当TYPE=4时，如果PC回NAK相应，则VMC继续自动向PC发相同消息（ADMIN_RPT.TYPE=4），直到PC响应ACK。全过程提示“正在全系统同步…”

注意：在整个TYPE=4的全系统同步过程中，VMC不能给PC发送其他消息，也不再处理其他PC消息。除非同步成功，或VMC重新启动！

注意：小LED/LCD要根据ACK/NAK，做好对应的正确和错误提示

8.12 ACTION_RPT

功能：售货机行为报告

8.12.1 action=0：心跳

Ver = 3:

字段名	action
字节数	1

提示：在正常售卖模式，VMC需要周期性，每10秒发送一次心跳。维护模式无需发送。

8.12.2 action=1：出货开始

Ver = 3:

字段名	action	seconds	hd_id	type	cost	total_value
字节数	1	1	1	1	2	2

hd_id/type/cost/total_value 与 VENDOUT_RPT 相同。

seconds：

- VMC预估出货完成需要的最长时间（秒）
- 注意：对于最长出货时间在1秒以内的，无需通知 “出货开始”

8.12.3 action=2：出币开始

Ver = 3:

字段名	action	seconds	value	total_value	type
字节数	1	1	2	2	1

所有出币开始之前发送，包括自动退币，手工退币，PAYOUT_IND出币。

seconds：

- VMC预估退币完成需要的最长时间（秒）
- 注意：对于最长出币时间在1秒以内的，无需通知 “出币开始”

value：

- 同PAYOUT_RPT.value

total_value：

- 代表出币开始前，用户总余额（比如，用户投入9元钱，出币5元，则：value=5元，total_value=9元）

value：

- 同PAYOUT_RPT.type

8.12.4 action=5：维护模式

通知PC，进入或退出维护模式（只在维护模式发送action=5消息，每500ms发送一次）

Ver = 3:

字段名	action	value
字节数	1	1

value：

- 0：VMC退出维护模式
- 非0：VMC在维护模式中

注：action=5，只在没有其他ADMIN_RPT消息的时候才需要发送

8.13 BUTTON_RPT

功能：VMC按键报告。

8.13.1 type=0：游戏按键

Ver <= 3:

字段名	type	value
字节数	1	1

value：固定为0（代表具体哪个游戏按键，当前只有一个游戏按键）

8.13.2 type=1：选择货道（适用使用小键盘或其他方式，直接选择货道的机型）

Ver = 3:

字段名	type	device	value
字节数	1	1	1

value：代表用户选择的是那个货道（逻辑货道ID）

device：同 HUODAO_IND.device

8.13.3 type=2：选择商品（适用展示位选货的机型）

Ver = 3:

字段名	type	device	value
字节数	1	1	1

value：代表用户选择的是那个商品ID

device：同 HUODAO_IND.device

8.13.5 type=4：退币/找零按键

Ver = 3:

字段名	type	value
字节数	1	1

value：固定为0，代表用户手工拨动退币按钮

注意：对于TYPE=0、1、2、4参见《游戏按键和购物按键》

9. VM数据获取

9.1 GET_STATUS

功能：PC通知VMC上报STATUS_RPT。

Ver <= 3:

字段名	
字节数	0

PC需要周期性（3分钟）发送GET_STATUS。

VMC收到该消息后，需要在5秒之内上报STATUS_RPT。

9.2 STATUS_RPT

功能：VMC状态报告。

Ver = 3:

字段名	check_st	bv_st	cc_st	vmc_st	pos_st	change
字节数	7 6	5 4	3 2	1 0	7 6 5 4 3 2 1 0	2

tem1	tem2	tem3	tem4	tem_st	自动退币
1	1	1	1	1	1

change：

- 机器中，可用的找零量总金额（包括硬币和纸币）

pos_st：

- 展示位状态，无展示位的机型，post_st=0xff
- 两两bit位组合，最多可用表示4组展示位：bit1 bit0表示第一组，…，bit7 bit6表示第4组
- 对于每一组，0=正常，1=被软件禁用，2=故障，3=不存在

cc_st：

- 硬币器状态（包含找零功能）：0=正常，1=被软件临时禁用，2=故障，3=设备不存在
- “被软件临时禁用”指设备运行过程中的临时禁用。
- 可能导致报“故障”的情况：
  - 卡币
  - 收币功能故障
  - 找零功能故障
  - 硬币管被取下（或没有正确安装到位）
  - 传感器故障
  - 其他任何导致硬币器不收币或不找零的情况（被软件临时禁用除外）
- 可能导致报“设备不存在”的情况：
  - 设备电源故障
  - 设备信号线开路
  - 设备上电前就已经被拆掉
  - 设备上电前就已经完全损坏
  - 维护菜单“停用”硬币器

bv_st：

- 纸币器状态：0=正常，1=被软件临时禁用，2=故障，3=设备不存在
- “被软件禁用”指设备运行过程中的临时禁用。
- 可能导致报“故障”的情况：
  - 卡币
  - 钱箱满
  - 钱箱被取下（或没有正确安装到位）
  - 货币识别传感器故障
  - 纸币器堵塞
  - 纸币器马达故障
  - 其他任何导致纸币器不收币的情况（被软件临时禁用除外）
- 可能导致报“设备不存在”的情况：
  - 同cc_st【可能导致报“设备不存在”的情况】

vmc_st：

- VMC状态（代表整个售货机）：0=正常，1= 正常货道商品全部售空，或非售卖时间（盒饭机在SALETIME_IND之外，其他机型CONTROL_IND.type=8之外），2=故障，3=维护模式
- vmc_st 的值，具体参见《VMC“暂停服务”》

check_st：

- “出货检测”设备：0=正常，1=被软件禁用，2=故障，3=不支持出货检测功能
- “被软件禁用”指手工通过维护菜单“关闭”出货检测功能。如果是由于VMC自动判断出货检测模块故障而自动禁用出货检测，属于“故障”

tem_st：

- 货仓状态设置值，共支持4个货仓：

字段名	货仓4	货仓3	货仓2	货仓1
bit位	7 6	5 4	3 2	1 0

状态值：

0 0	常温
0 1	制冷
1 0	加热
1 1	无此货仓

注：即使货仓在节能工作时间段或暂停服务时间段，不影响对应的状态值。

tem1：

- 货仓1温度，8位有符号数，该温度通过货仓内传感器获取，单位：℃
- 0xff（11111111）表示温度控制装置（所有和加热/制冷相关器件、温度传感器、电源）故障，导致不能进行规定的加热或制冷

- 0xfe（11111110）表示不存在此货仓
- 0xfd（11111101）表示该温度无意义（无制冷/制热模组）
- 其它情况表示当前货仓温度。如：tem1=0x00表示0℃，tem1=0x08表示8℃，tem1=0x82表示-2℃

tem2、3、4：

- 货仓2、3、4温度，其他含义同tem1

自动退币：

- 自动退币时间。
- 0：表示商品出货后，立即自动退币
- 255：表示永不自动退币
- 1-254：表示商品出货后，自动退币时间（单位：秒）

注意：如果VMC自动检测到某个设备状态发生变化，需要主动发送STATUS_RPT一次。

9.3 GET_HUODAO

功能：PC通知VMC上报HUODAO_RPT。

Ver = 3:

字段名	device
字节数	1

device：

- 同 HUODAO_IND.device

PC需要周期性（3分钟）发送GET_HUODAO。

VMC收到该消息后，需要在5秒之内上报HUODAO_RPT。

9.4 HUODAO_RPT

功能：VMC货道报告。

Ver = 3:

字段名	device	huodao	…	huodao
字节数	1	1	…	1

device：

- 同 HUODAO_IND.device

huodao： zyxxxxxx

- “z”固定填0
- “y”=0，表示货道正常
- “y”=1，表示货道故障（根据机型的不同，可能是电机、电磁阀、电磁铁等组件故障）

- “xxxxxx”，表示商品余量，如果商品余量大于63，则统一为63
- huodao字段的个数应与VMC_SETUP中hd_num相同，如果PC发现不相同，则忽略该消息，并记录和上报错误日志

注意：对于含有自动检测商品是否售空的机型（如饮料机），当VMC检测到货道从“有货”到“无货”转换时，需要第一时间主动通知PC一次HUODAO_RPT。

注意：当VMC检测到正常货道发生故障后，需要第一时间主动通知PC一次HUODAO_RPT。

9.5 GET_INFO

功能：PC通知VMC上报INFO_RPT。

Ver <= 3:

字段名	type
字节数	1

type：

- type=3：用户投币余额
- type=4：销售时间段配置（盒饭机专用）
- type=5：VMC时间
- type=6：出货统计
- type=7：节能工作时间段
- type=8：售卖时间段
- type=9：照明时间段
- type=10：货道关系
- type=11：展示位关系
- type=12：售价关系
- type=13：VMC能力
- type=14：VMC软件版本
- type=15：VMC硬件唯一标识
- type=16：纸币器信息（如果纸币器不存在，则VMC无需回复INFO_RPT）
- type=17：硬币器信息（如果硬币器不存在，则VMC无需回复INFO_RPT）
- type=18：压缩机信息（如果压缩机不存在，或无法获取压缩机信息，则VMC无需回复INFO_RPT）
- type=19：售货机产品型号
- type=23：售货机机型相关错误码

根据实际需要，PC需要周期性（3分钟）针对某些type，发送GET_INFO。

VMC收到该消息后，需要在5秒之内上报INFO_RPT；如果PC 5秒内没有收到INFO_RPT，则PC超时。

注意：对于VMC不支持的TYPE类型，忽略该消息，不回复对应的INFO_RPT。

9.6 INFO_RPT

功能：VMC信息报告。

type=3：用户投币余额

Ver = 3:

字段名	type	total_value
字节数	1	2

total_value：

- 同“PAYIN_RPT”total_value

type=4：销售时间段配置（盒饭机专用）

Ver <= 3:

字段名	type	时间段 1	时间段 2	时间段 3	时间段 4	时间段 5
字节数	1	4	4	4	4	4

时间段x：

- 同“SALETIME_IND”

type=5：VMC时间

Ver <= 3:

字段名	type	年	月	日	时	分	秒	星期
字节数	1	2	1	1	1	1	1	1

同《CONTROL_IND》type=17

type=6：出货统计

Ver = 3：

字段名	type	全部出货量	全部收入	现金出货量	现金收入	游戏出货量	游戏收入
字节数	1	4	4	4	4	4	4

刷卡出货量	刷卡收入	在线出货量	在线收入	硬币投入	纸币投入	硬币出币
4	4	4	4	4	4	4

- 各统计项详情，参见《商品出货统计》、《现金投入/支出统计》

---

type=7：节能工作时间段

Ver = 3：

参考“CONTROL_IND.type=7”

字段名	type	flag	Group 1	Group 2	Group 3
字节数	1	2	28	28	28

---

type=8：售卖时间段

Ver = 3：

参考“CONTROL_IND.type=8”

字段名	type	flag	Group 1	Group 2	Group 3
字节数	1	2	28	28	28

---

type=9：照明时间段

Ver = 3：

参考“CONTROL_IND.type=9”

字段名	type	flag	Group 1	Group 2	Group 3
字节数	1	2	28	28	28

---

type=10：货道关系

Ver = 3：

参考“HUODAO_IND”

字段名	type	device	sp_id	…	sp_id
字节数	1	1	1	1	1

---

type=11：展示位关系

Ver = 3：

参考“POSITION_IND”

字段名	type	device	sp_id	…	sp_id
字节数	1	1	1	1	1

type=12：售价关系

Ver = 3：

参考“SALEPRICE_IND”

字段名	type	device	sp_price	…	sp_price
字节数	1	1	2	n	2

type=13：VMC能力

Ver = 3：

字段名	type	离线售卖数据存储容量
字节数	1	2

离线售卖数据存储容量：

- VMC分配给离线售卖的数据存储空间，最大能存储多少条离线销售记录
- 如果机器没有存储空间，则“存储容量=0”

type=14：VMC软件版本

Ver = 3：

字段名	type	soft_ver
字节数	1	n

soft_ver：

- 软件版本编号，ASCII码表示，不能出现不可见字符和汉字
- 编号方式：“厂商名英文或拼音”+“厂商自有编号”+“(yyyy-MM-dd)”

type=15：VMC硬件唯一标识（每台售货机均不相同）

Ver = 3：

字段名	type	hard_ver
字节数	1	n

hard_ver：

- ASCII码表示，可以使用CPU唯一编号等代替，不能出现不可见字符和汉字

type=16：纸币器信息

Ver = 3：

字段名	type	level	CC	Bill Scaling Factor	Decimal Places	钞箱容量	安全等级	暂存功能	纸币1#	…	纸币8#	Z
字节数	1	1	2	2	1	2	1	1	1	6	1	33

level:

- 纸币器级别：level=1或2

CC:

- 货币代码

Bill ScalingFactor:

- 纸币基数值

Decimal Places:

- 小数点位数

安全等级:

- bit0-bit7，分别指代纸币1#到纸币8#的安全等级

暂存功能:

- 如果等于0x00，则纸币器无暂存功能
- 如果大于0x00，则纸币器有暂存功能

钞箱容量:

- 纸币器钞箱能容纳多少张纸币

纸币 n#:

- 前六种纸币币种相当于多少个纸币基数

Z:

- 纸币器最高level的IDENTIFICATION

type=17：硬币器信息

Ver = 3：

字段名	type	level	CC	Coin Scaling Factor	Decimal Places	硬币1#	…	硬币6#	找零1#	…	找零6#	Z
字节数	1	1	2	1	1	1	4	1	1	4	1	33

level:

- 硬币器级别：level=2或3

CC:

- 货币代码

Coin ScalingFactor:

- 硬币基数值

Decimal Places:

- 小数点位数

硬币 n#:

- 前六种纸币币种相当于多少个硬币基数

找零 n#:

- 前六种找零币种相当于多少个硬币基数

Z:

- 硬币器最高level的IDENTIFICATION

---

type=18：压缩机信息

Ver = 3：

字段名	type	压缩机信息
字节数	1	n

ASCII码表示的压缩机型号、厂商等信息。

---

type=19：售货机产品型号（同一系列的售货机，产品型号相同）

Ver = 3：

字段名	type	售货机产品型号
字节数	1	n

ASCII码表示的售货机产品型号。（同一系列的售货机，产品型号相同）

---

type=23：售货机机型相关状态和错误码

Ver = 3：

字段名	type	machine_st
字节数	1	n

由于每种售货机机型都有自己特定的错误码，故无法统一，详见各个机型相关说明文档。

9.7 GET_SETUP

功能：PC通知VMC，上报VMC_SETUP 消息。

Ver = 3:

字段名	
字节数	0

9.8 GET_OFFLINE_DATA

功能：PC通知VMC，上报OFFLINE_DATA_RPT 消息。

Ver = 3:

字段名	
字节数	0

注1：VMC需要自动判断，如果存在type=0 的离线数据，需要先上报！ 最后再上报type = 1的离线数据。如果不存在 type = 0 的离线数据，VMC直接上报 type = 1的离线数据。

注2：VMC每收到一条 GET_OFFLINE_DATA，可以只上报一条 OFFLINE_DATA_RPT，也可以上报多条 OFFLINE_DATA_RPT，当必须在前一条收到ACK后，再发送后一条。

例：VMC连续上报多条 OFFLINE_DATA_RPT 的情况：

〈SN〉	〈Message〉	〈说明〉
	〈延时 500ms〉	
18	POLL	
18	GET_OFFLINE_DATA	
	〈VMC 准备数据〉	
19	OFFLINE_DATA_RPT.type=0	
19	ACK	
	〈VMC 准备数据〉	
20	OFFLINE_DATA_RPT.type=0	
20	ACK	
	〈VMC 准备数据〉	
21	OFFLINE_DATA_RPT.type=1	VMC 发送 type = 1 的离线数据，标志着离线数据发送完毕
21	ACK	
	〈延时 500ms〉	
22	POLL	
22	ACK	

第四部分 默认行为

概述

默认行为对友宝自动售货机各种状态下的动作进行标准化定义。

传统机型各种状态都定义了自己的处理方式，没有统一标准，运维和物流配送人员需要学习和记忆各个机型之间的差异点，最终导致学习和培训成本高，易出错。

这里的默认行为，均针对Ver=3版本。

1. 消息超时及通信中断

1.1 VMC等待超时

如果VMC发出的消息F7位为1，则强制PC_ERR=0；消息发出后，在1.5秒以内没有收到PC的响应， VMC将进行2次消息重传。每超时一次，将PC错误消息数（PC_ERR）加1。

如果PC_ERR累加到3或者3以上，则认为VMC与PC通信中断， VMC自动进入“离线售卖”模式。

提示：VMC在F7=1的消息发送和重传期间，没有收到PC的响应之前，不能再向PC发送任何F7=1的消息。

消息流举例：

<SN>	<Message>	<说明>
18	COST_RPT	（PC_ERR=0）
	<等待 1.5s>	这 1.5 秒内，VMC 收到任何 SN!=18 的 PC 消息，均丢弃（PC_ERR=1）
18	COST_RPT	重传第 1 次
	<等待 1.5s>	这 1.5 秒内，VMC 收到任何 SN!=18 的 PC 消息，均丢弃（PC_ERR=2）
18	COST_RPT	重传第 2 次
	<等待 1.5s>	这 1.5 秒内，VMC 收到任何 SN!=18 的 PC 消息，均丢弃（PC_ERR=3）
	<VMC 进入离线售卖>	连续 3 次超时，自动进入离线售卖模式，VMC 不再发送 POLL 外任何其它消息
	<额外等待 5s>	
19	POLL	（PC_ERR=0）
19	POLL	每 5 秒重发一次（Retry each 5s）（PC_ERR=0）
19	POLL	每 5 秒重发一次（Retry each 5s）（PC_ERR=0）
19	POLL	每 5 秒重发一次（Retry each 5s）（PC_ERR=0）
19	POLL	每 5 秒重发一次（Retry each 5s）（PC_ERR=0）
19	POLL	每 5 秒重发一次（Retry each 5s）（PC_ERR=0）

任何时候，只要VMC与PC通信恢复正确，则将PC_ERR置0，比如：

<SN>	<Message>	<说明>

18	VENDOUT_RPT	
	〈等待 1.5s〉	这 1.5 秒内，VMC 收到任何 sn!=18 的 PC 消息，均丢弃（PC_ERR=1）
18	VENDOUT_RPT	重传第 1 次
	〈等待 1.5s〉	这 1.5 秒内，VMC 收到任何 sn!=18 的 PC 消息，均丢弃（PC_ERR=2）
18	VENDOUT_RPT	重传第 2 次
18	ACK	VMC 正确收到 ACK （PC_ERR=0）
	〈等待 500ms〉	
19	POLL	

注意：如果VMC发送的最后一条消息是VENDOUT_RPT，则VMC需要把这个出货记录保存到离线销售数据中，比如：

〈SN〉	〈Message〉	〈说明〉
18	VENDOUT_RPT	（PC_ERR=0）
	〈等待 1.5s〉	这 1.5 秒内，VMC 收到任何 sn!=18 的 PC 消息，均丢弃（PC_ERR=1）
18	VENDOUT_RPT	重传第 1 次
	〈等待 1.5s〉	这 1.5 秒内，VMC 收到任何 sn!=18 的 PC 消息，均丢弃（PC_ERR=2）
18	VENDOUT_RPT	重传第 2 次
	〈等待 1.5s〉	这 1.5 秒内，VMC 收到任何 sn!=18 的 PC 消息，均丢弃（PC_ERR=3）
	〈VMC 进入离线售卖〉	连续 3 次超时，自动进入离线售卖模式，VMC 不再发送 POLL 外任何其它消息。 VMC 需要将 VENDOUT_RPT 保存到离线销售数据中。
	〈额外等待 5s〉	
19	POLL	（PC_ERR=0）
19	POLL	每 5 秒重发一次（Retry each 5s）（PC_ERR=0）

2. VMC“暂停服务”

VMC“暂停服务”情况如下：

- 非故障货道中的商品全部售空
- 新机器（或进行过“恢复出厂默认参数”），机器中没有货道、商品、价格等数据
- 售货机所有货道均故障（如果仅单个货道故障，不影响整机售卖，只需要停止该故障货道售卖，不能进入“暂停服务”模式，也即只要还有一个货道能正常工作，就不能整机暂停服务），参见《出货失败和货道故障》
- 使用机械手臂的机型，如果机械手臂故障，整机“暂停服务”
- 取货口卡货（清理后VM需要自动恢复售卖）
- VM在“在线售卖”或“离线售卖”模式，VMC均需要根据（CONTROL_IND消息type=8）配置，自动

根据系统售卖时间，判断是否需要进入“暂停服务”

- VMC或PC认为其他影响整机继续售卖的情况

特殊说明：

- 盒饭机：非售卖时间段需要“暂停服务”，参见《盒饭机》
- 制冷、制热故障，整机继续工作，不能“暂停服务”
- 纸/硬币器，部分或全部故障，不“暂停服务”，参见《现金购物和非现金购物》
- 对含有“出货检测”的机型，出货检测模组故障，一般不“暂停服务”，除非特殊说明，参见《出货检测》
- 如果是《消息超时及通信中断》的情况，则VM自动进入“离线售卖”模式
- VM在“离线售卖”模式，参见《离线售卖和在线售卖》
- 其他情况

一旦出现整机“暂停服务”，VMC需要：

- 刚进入暂停服务，主动发送一次STATUS_RPT 消息，其中vmc_st 的取值：

  如下情况vmc_st = 1：

  - CONTROL_IND.type = 8 规定时间之外（非盒饭机）
  - SALETIME_IND 规定的时间之外（盒饭机）
  - 售货机商品售空

  其他情况vmc_st = 2。
- 同时本机小LCD显示“暂停服务”，LED显示“-----”
- 强制自动退币（用户余额）（该退币需要统计入《现金投入/支出统计》中）
- 自动关闭现金收银设备（刚进入暂停服务模式时，自动关闭一次）
- 自动关闭照明
- 自动关闭制冷、制热
- 继续保持与PC正常通信
- 不处理PC的VENDOUT_IND消息，直接回复NAK_RPT
- 继续处理 PAYOUT_IND 和 COST_IND
- 如果在“出货过程”中出现“暂停服务”，VMC需尽量确保“出货过程”成功结束；如果确实无法保证，可以强制中断该出货过程，向PC上报VENDOUT_RPT出货失败；如果商品现金扣款，则VMC需要自动退款
- 暂停服务模式下，VMC不接受用户按键（展示位按键、小键盘、游戏按键），但退币按键需要正确处理，并正确上报 BUTTON_RPT
- 暂停服务模式下，PC可以通过 CONTROL_IND.type = 2 打开/关闭纸/硬币器

3. 整机非法断电或异常复位

这里特指VMC不可控和不可预知的断电或复位。

用户余额：直接吞掉，无需掉电保存

4. 商品出货

提示：如果手工通过展示位选择商品（无论投币与否），VMC判断已经售空，则主控板蜂鸣器在0.4秒钟内短鸣2声。如果是直接选择货道的机型，直接发起出货

“商品出货”，指VMC决定要控制货道动作，到VMC检测到商品成功出货（或商品出货失败）；这期间短则几百毫秒，长则几十秒。

对于现金购物，VMC在确定要出货之后，需要：

- 从用户的现金余额中扣除商品货款
- LED/LCD上显示的金额需要动态更新，减去商品扣款
- 上报ACTION_RPT.action=1

对于最长出货时间大于1秒的机型，在出货过程中，需要：

- VMC需要有超时机制，超过最大出货时间，自动终止出货流程
- 在“出货过程”中，VMC不给PC发送F7=1的消息，F7=0的消息可以发送
- 出货过程，不发送BUTTON_RPT.type=1/2/3/4
- 临时关闭纸币器、硬币器收币功能，直到出货结束

对于VMC而言，在正在出货的过程中，会出现如下几种冲突：

- 收到来自用户的按键/键盘选货请求（VMC直接拒绝该请求，不用上报PC）
- 收到PC的VENDOUT_IND、PAYOUT_IND、COST_IND消息，直接NAK_RPT拒绝

5. 出货失败和货道故障

5.1 出货失败

“出货失败”指售货机系统开始进行商品出货，在*规定时间*内，VMC自动判断，认为商品没有出现到“取货口”。

一旦出现出货失败的情况，VMC需要：

- 发送失败报告VENDOUT_RPT（status=2）
- 主控板蜂鸣器0.4秒钟内短鸣2声
- 如果是现金购买（包含VENDOUT_IND发起type=0且cost大于0的情况），则VMC进行退款，可能退币也可能不退，参见《自动找零时间设置》
- 在小LCD提示“出货失败，已退款”/LED显示“Error”，持续3秒，之后继续显示用户余额
- 如果是非现金购买，则VMC无需发起退款，只需上报VENDOUT_RPT（status=2）即可，具体的退款流程交由PC处理

出货失败原因列举：

- 货道中实际并没有货，属于配送人员人为配置错误
- 售货机软件故障，实际并没有真正触发出货模组进行出货
- 货道故障，参见《货道故障》
- 实际已经掉货，但“出货检测”模块没有检测到商品出货
- 其他原因

5.2 货道故障

“货道故障”指售货机空闲、自检、出货等过程中，自动判断出货道存在某种问题，会导致货道无法继续进行商品出货，需要运维人员到现场进行维修。

货道故障原因：

- 货道电机或相关模块故障
- 商品将货道卡死
- 超时了，货道仍没有成功转一圈
- 其他原因

当VMC判断出现*新的*货道故障，需记录该故障（要求掉电不丢失），并立即给PC发送一次HUODAO_RPT。

货道故障的查询和恢复：VM需要提供简单方便的方法，供运维人员现场快速查询本机有无货道发生故障；同时需要提供方法供运维人员检修完成后，快速清除货道故障状态，恢复运营，参见《维护菜单》。

6. 出货检测

“出货检测”指售货机提供的方法，用于判断商品是否成功出现到取货口。

如果VMC自动检测到出货检测硬件故障，VMC自动“关闭”出货检测（与通过菜单关闭出货检测相同）。

如果出货检测“关闭”，对弹簧机而言即只要VMC检测到电机转动一圈，则认为出货成功，其他货道结构的机型原理类似，参考实现。

注意：不论出货检测正常或故障，如果VMC检测到电机没有成功转一圈，则认为出货失败，参见《货道故障》

强调：对于某些严重依赖出货检测的机型，出货检测模组故障需要整机“暂停服务”，需要厂商在机型规格中特殊说明。

说明：配置菜单提供出货检测模组“开/关”，可以手动打开和关闭。出厂默认打开出货检测。

7. 商品售价

当商品售价为0或0xffff时，表明这是一件特殊商品。无论用户是否投币，用户通过售货机自生装备（选货按钮或小键盘）发起的这两种售价商品购买，小LCD显示“非现金售卖商品”/LED显示“0000”3秒钟，VMC不发起商品出货，仅向PC报告BUTTON_RPT消息。是否需要真正出货，由PC判定。

VMC需要有显示小数点的能力，如“0.5” 元、“22.50”元、“6.0”元；至于是否显示小数点，由

VMC_SETUP中的ScaleFactor和DecimalPlaces决定，如果 ScaleFactor*10 $^{-(DecimalPlaces)}$ 为整数，则不需要显示小数点，计算结果包含几位小数，则VMC需要显示几位小数！

需要注意：

- 对于VENDOUT_IND发起的非现金购买出货命令，不受该约束限制
- 在计算售货机商品最高售价时，需要忽略0xffff这个特殊值
- “离线售卖”模式，特殊售价 0 与0xffff的商品，VMC不发起商品出货

8. 货道和展示位映射

VMC收到映射关系后，需要保存到非易失存储器中，确保掉电不丢失。

注意：考虑存储器写入寿命有限，故VMC需要先进行判断，如果映射关系没有发生变化，则不需要覆盖存储器中的内容。

8.1 HUODAO_IND和POSITION_IND对应关系

- 展示位按钮购物：(展示位按钮选货的机器)
  - 如果展示位按钮(POSITION_IND)与货道(HUODAO_IND)存在映射关系，则用户的购买行为就应该被VMC正确处理，除非商品售价为0或0xffff，参见《商品售价》
  - 如果商品A只配置展示位按钮，没有配置货道。则用户按动了展示位按钮， VMC无需发起出货动作，只需直接向PC上报BUTTON_RPT消息（原因是展示位上可能配了一件非售货机中的物理商品，比如虚拟充值卡），VMC小LCD/LED无需任何特殊提示，主控板蜂鸣器响一声。
  - 如果商品A没有配置展示位按钮，但配置了货道。则当VENDOUT_IND发起了商品A的出货， VMC需要正确处理该VENDOUT_IND消息，参见《现金购物和非现金购物》
- 小键盘购物： (一般是对应透明门机型，用户使用键盘直接输入货道编号进行商品购买)
  - 如果用户选择的货道对应的商品ID，在SALEPRICE_IND中无法查询到，则在VMC小LCD提示“非现金售卖商品”/LED显示“Error”， VMC不发起商品出货，仅向PC报告该BUTTON_RPT消息；
  - 如果商品ID在SALEPRICE_IND中能查询到，则用户的购买行为就应该被VMC正确处理，除非商品售价为0或0xffff，参见《商品售价》

8.2 VMC物理货道和PC逻辑货道

VMC物理货道：按照各种机型特点规则进行编码；在设计上，需要考虑“1拖2”的情况（综合机与综合机、综合机与饮料机、饮料机与饮料机）。

具体编号上，采取“从左到右、从上到下”进行编号：

举例：如下是常见的综合机物理货道编号方式

11 12 … 17 18

21 22 … 27 28

……

61 62 … 67 68

建议：

1.对于单层货道超过9个的机型，采用11、12、…、1A、1B、… 1F 的方式编码

2.对于1拖2机型，主箱体使用11、12、…、1F的方式编码，从箱体使用A1、A2、…AF的方式编码

PC逻辑货道：采取自然数编号的方式，其总货道个数与VMC物理货道相同，采取按顺序一对一映射：

举例：如下是PC自然数货道编号与VMC物理货道映射关系

VMC物理货道： 11 12 … 17 18 21 22 … 61 62 … 67 68

PC逻辑货道： 1 2 … 7 8 9 10 … 41 42 … 47 48

VMC之需要把物理货道映射到逻辑货道。

9. 离线售卖和在线售卖

“离线售卖”，指VMC与PC在失去连接的情况下，VMC继续按照自己保存的配置进行商品售卖，但需要保存商品售卖数据，掉电不丢失。

注意：在离线售卖模式，不能改变售货机的商品价格、货道与展示位等关系，只能设置单个货道商品补货个数。

“在线售卖”，指VMC与PC正常通信的情况下，进行商品售卖；在线售卖时，VMC不需要保存商品售卖数据，只需要实时传递给PC即可。

离线售卖需要离线保存销售记录：最少保存2000条离线出货记录（旧机型改造可以适量减少）。当超过最大条数后，VMC不再保存详细离线销售记录，但仍需继续售卖，并保存一个汇总记录（详见《OFFLINE_DATA_RPT.type=1》）。

离线售卖其他说明：

- 离线售卖产生的销售数据，将在下一次与PC建立通信后，传递给PC，数据未传送完成，不发送POLL
- 离线销售数据，需要按时间先后顺序存储；也需要按同样的顺序传送给PC
- 在数据同步过程中，PC确认收到的数据，VMC需要同步删除
- 如果在数据传输过程中，又失去与PC连接，则VMC继续进入离线售卖模式，未传输完的离线数据继续保留在VMC中
- 离线销售记录，需要同步统计到《商品出货统计》
- 如果VMC在PC设定的“暂停服务”时间段（CONTROL_IND.type=8）内与PC失去连接，则VMC继续“暂停服务”直到时间段结束

何时进入“离线售卖”：

- 在“在线售卖”模式，与PC失去连接（参见《消息超时及通信中断》），VMC自动进入“离线售卖”
- VMC上电就与PC失去连接，VMC自动进入“离线售卖”

何时恢复“在线售卖“：

- 在“离线售卖”模式，VMC需要周期发送POLL，尝试与PC建立连接
- 当PC响应VMC的POLL消息，恢复在线售卖模式

说明：离线售卖模式，VMC需要强制开启已经被PC禁用的现金收币设备，否者离线售卖模式没有支付工具；
当恢复在线售卖后，PC可以重新关闭这些设备

离线售卖需要保存的数据：

- 商品出货离线数据：4个字节的出货时间+1字节货道id+1字节扣款金额
- 当离线数据满了后，只需要单独保存总个数和总金额
- 不保存的数据：纸币投币、硬币投币、找零

算法1：计算友宝元年开始，到给定时间内的秒数

```
#include<stdio.h>
/* 瑞年每年的秒数 = 3600*24*366 */
#define SECONDOFLEAPYEAR  31622400
/* 正常年每年的秒数  = 3600*24*365 */
#define SECONDOFYEAR  31536000
/* 每天的秒数  = 24 * 60 * 60 */
#define MAXSECONDOFDAY 86400
char const month_s[2][12] = {
                { 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31 },
                { 31, 29, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31 } };
/**
 * 计算从友宝元年（2011年1月1日0点0分0秒）开始，到给定时间内的秒数。<br/>
 * 注1：这里计算的秒数不考虑时区<br/>
 * @param wYear   年。有效值： 2011 - 2111
 * @param wMonth  月。有效值：1 .. 12，分别代表1月  .. 12月
 * @param wDay    每月第几天。有效值：1 .. 31，分别代表第1天 .. 第31天
 * @param wHour   时。24小时制，有效值：0-23
 * @param wMinute 分。有效值：0-59
 * @param wSecond 秒。有效值：0-59
 * @return
 */
unsigned int calcSecondsFrom2011(int wYear, int wMonth,
            int wDay, int wHour, int wMinute, int wSecond) {
        int tTemp = 0;
        int tSecond = 0;

        int nCount = 0;
        int i;
        int j;

        tSecond = wHour * 3600 + wMinute * 60 + wSecond;

        for (i = 2011; i < wYear; ++i) {
          if (isLeapYear(i)) {
              ++nCount;
          }
        }

        tTemp += (wYear - 2011 - nCount) * SECONDOFYEAR + nCount
              * SECONDOFLEAPYEAR;
```

```
        if (wMonth > 1) {
          if (isLeapYear(wYear)) {
              for (j = 0; j < wMonth - 1; ++j) {
                  tTemp += month_s[1][j] * MAXSECONDOFDAY;
              }
              tTemp += (wDay - 1) * MAXSECONDOFDAY + tSecond;
          } else {
              for (j = 0; j < wMonth - 1; ++j) {
                  tTemp += month_s[0][j] * MAXSECONDOFDAY;
              }
              tTemp += (wDay - 1) * MAXSECONDOFDAY + tSecond;
          }
        } else {
          tTemp += (wDay - 1) * MAXSECONDOFDAY + tSecond;
        }

        return tTemp;
}

int isLeapYear(int year) {
        /* is leap year? */
        if ((year % 100 != 0) && (year % 4 == 0)) {
          return 1;
        }
        if ((year % 100 == 0) && (year % 400 == 0)) {
          return 1;
        }
        return 0;
}

//test
int main() {
          unsigned int mySeconds = calcSecondsFrom2011(2012, 7,
              13, 17, 3, 12);
          printf("seconds from 2011=%u", mySeconds);
    return 0;
}
```

10. 现金购物和非现金购物

10.1 现金购物

用户从纸币器、硬币器投入现金，使用这些现金进行商品购买的行为，即为现金购物。有两种情况会视为现金购物：

- 用户直接通过售货机的物理选货按键或选货键盘购买
- 用户通过触屏或其他方式购买，PC下发VENDOUT_IND（type=0，cost大于0）

现金购物方式下，VMC发出的VENDOUT_RPT消息，type=0。

VMC收到VENDOUT_IND（type=0），需要进行如下判断：

- VENDOUT_IND指示扣款金额为cost，如果用户余额可以进行扣款（参见《用户投币及找零》），则VMC立即给PC回复ACK_RPT；否者立即回复NAK_RPT。

- 如果回复ACK_RPT，VMC从用户余额中扣款
  - 如果扣款成功，VMC发起出货
  - 如果扣款失败(如纸币压钞失败)，则流程结束！VMC需要发送ACTION_RPT.type=4消息！同时需要发送PAYIN_RPT（dt=101）。

注：现金购物出货失败，售货机主动退币。

10.2 非现金购物

“现金购物”以外的出货方式，均视为非现金购物，包括刷卡、游戏中奖、手机购物等。

注意： VENDOUT_IND关于type和cost的强制约定，参见《VENDOUT_IND》

11. 用户投币、找零及多国货币

目标：

1. 尽最大可能进行商品售卖；硬币器故障，不能停用纸币器。找零不足不能停用硬币器和纸币器

2. 支持多国货币（纸硬币必须是一个币种）（通过更换纸/硬币器自动完成，售货机零配置）

3. 找零金额自动计算

4. 自动计算，投币是否超过找零上限

5. 自动计算，购物后是否能正常找零

注意：整个退币过程中，VMC停止主动发送F7=1的消息，但可以发送F7=0的消息

提示：为使找零更安全，硬币器报给VMC的找零量，VMC可以自己将各种面额的硬币，**预留2个**。比如硬币器给上报5角硬币10枚，但VMC认为只有8个！

算法2： 计算找零

基于MakeChange算法，可以方便的计算是否可以找零及找零组合。

```
#include <stdio.h>
/**
 * 计算找零方案
 * @param in length   数组的长度
 * @param in coins    可找零币种
 * @param in lefts    每种币剩余量
 * @param out results 找零结果（返回结果"成功"时有效）
 * @param in money
 * @return           返回1：成功；返回0：失败
 */
int MakeChange(int length, unsigned char* coins,
        unsigned char* lefts, unsigned char* results, int money) {
    printf("总金额=%d\n", money);
    int i=0;

    for(i=0; i<length; ++i) {
        unsigned char coin = coins[i];
```

```
        unsigned char left = lefts[i];

        if(left <= 0 || coin > money) {
            continue;
        }
        int count = money / coin;

        if(count > left) {
            count = left;
        }
        lefts[i] = left - count;

        printf("%d=%d\n",coin,count);
        results[i] = count;

        money -= count*coin;
        if(money == 0) {
            break;
        }
    }

    if(money == 0) {
        return 1;
    } else {
        for(i=0; i<length; ++i) {
            results[i] = 0;
        }
        return 0;
    }
}

void testUSD() {
    //可找零硬币面额（美分）（面额从大到小）
    unsigned char coins[5] = {100, 50, 25, 10, 5};
    //每种面额硬币可找零个数（硬币只支持255个，超过255也只填255）
    unsigned char lefts[5] = {0, 3, 5, 1, 0};
    //找零结果（确保传入MakeChange前，results被清零）
    unsigned char results[5] = {0, 0, 0, 0, 0};
    if(MakeChange(5, coins, lefts, results, 335)) { //找零335美分
        printf("找零成功\n");
    } else {
        printf("找零失败\n");
    }
}

void testCNY() {
    //可找零硬币面额（人民币角）（面额从大到小）
    unsigned char coins[2] = {10, 5};
    //每种面额硬币剩余个数
    unsigned char lefts[2] = {0, 8};
    unsigned char results[2] = {0, 0};
    if(MakeChange(2, coins, lefts, results, 35)) { //找零3.5元
        printf("找零成功\n");
    } else {
        printf("找零失败\n");
    }

}

int main() {
```

```
    testUSD();
    testCNY();
    return 0;
}
```

11.1 投币上限

基本原则：

- 用户的所有投币（纸币或硬币），如果放弃购物，可以全部退出
- 用户购物后，剩余金额可以全部退出
- 用户投币额已经达到或超过最大商品售价，不再接受投币（纸币和硬币）

硬件条件：

- 硬币器包含找零功能，同时支持循环找零功能
- 纸币器包含暂存最后一张纸币功能
- 即使所有硬币找零量均为0，用户投入的硬币也可以正常退币（循环找零）

算法2应用举例（现金购物）：

注意：

1.这里举例基于人民币，真正实现时需要测试其它币种。

2.“√”表示可以购买。

3.现金购买扣款后肯定会造成无法找零的情况，一律拒绝出货。

条件：单件商品最高单价10元；1元找零量3元，5角找零量0元

单价 币种	5角硬	1元硬	1元硬	1元纸	5元纸(这张纸币暂存)	1元硬	1元硬（硬币器暂停收币）
	0.5元	1.5元	2.5元	3.5元	8.5元	9.5元	10.5元
2元			√	√	√	√	√
2.5元			√	√	√	√	√
3.5元				√	√	√	√
5元					√	√	√
7元					√	√	√
10元							√
PAYOUT_IND	**1.0/2.0/ 3.0**	**1.0/2.0/ 3.0**	**1.0/2.0/ 3.0**	**1.0/ 2.0**	**1.0/ 2.0**	**1.0/ 2.0**	**1.0/ 2.0**

条件：单件商品最高单价10元；1元找零量3元，5角找零量0元

单价 币种	1元硬	1元硬	1元硬	1元纸	5元纸(这张纸币暂存)	0.5元硬	1元硬（硬币器暂停收币）
	1.0元	2.0元	3.0元	4.0元	9.0元	9.5元	10.5元
2元		√	√	√	√	√	√
2.5元						√	√
3.5元						√	√
5元					√	√	√
7元					√	√	√
10元							√
PAYOUT_IND	**1.0/2.0/ 3.0**	**1.0/2.0/ 3.0**	**1.0/2.0/ 3.0**	**1.0/ 2.0**	**1.0/ 2.0**	**1.0/ 2.0**	**1.0/ 2.0**

条件：单件商品最高单价10元；1元找零量3元，5角找零量0元

单价 币种	1元硬	1元硬	1元硬	1元纸	10元纸(这张纸币暂存；硬币器暂停收币)	N/A
	1.0元	2.0元	3.0元	4.0元	14.0元	
2元		√	√	√	√	
2.5元						
3.5元						
5元						
7元						
10元					√	
PAYOUT_IND	**1.0/2.0/ 3.0**	**1.0/2.0/ 3.0**	**1.0/2.0/ 3.0**	**1.0/ 2.0**	**1.0/ 2.0**	

条件：单件商品最高单价10元；1元找零量3元，5角找零量0元

单价 币种	1元纸	1元纸	1元纸	一元纸(这张纸币暂存)	5角硬	1元硬
	1.0元	2.0元	3.0元	4.0元	4.5元	5.5元
2元		√	√	√	√	√
2.5元					√	√
3.5元					√	√
5元						√
7元						

10元						
PAYOUT_IND	**1.0/2.0**	**1.0**	**0.0**	**0.0**	**0.0**	**0.0**

条件：单件商品最高单价10元；1元找零量0元，5角找零量0元

单价 币种	1元硬	1元硬	1元硬	1元纸(这张纸币暂存)	0.5元硬	1元硬	1元硬
	1.0元	2.0元	3.0元	4.0元	4.5元	5.5元	6.5元
2元		√	√	√	√	√	√
2.5元					√	√	√
3.5元					√	√	√
5元						√	√
7元							
10元							
PAYOUT_IND	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**

条件：单件商品最高单价10元；1元找零量0元，5角找零量0元

单价 币种	1元纸(这张纸币暂存)	1元硬	1元硬	1元硬	1元硬	0.5元硬	0.5元硬
	1.0元	2.0元	3.0元	4.0元	5.0元	5.5元	6.0元
2元		√	√	√	√	√	√
2.5元						√	√
3.5元						√	√
5元					√	√	√
7元							
10元							
PAYOUT_IND	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**

条件：单件商品最高单价10元；1元找零量0元，5角找零量0元

单价 币种	5元纸(这张纸币暂存)	1元硬	1元硬	0.5元硬	1元硬	1元硬	1元硬(硬币器暂停收币)
	5.0元	6.0元	7.0元	7.5元	8.5元	9.5元	10.5元
2元			√	√	√	√	√
2.5元				√	√	√	√

3.5元					√	√	√
5元	√	√	√	√	√	√	√
7元			√	√	√	√	√
10元							√
PAYOUT_IND	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**

条件：单件商品最高单价10元；1元找零量15元，5角找零量0元

单价 币种	1元硬	1元硬	1元硬	1元纸	5元纸	0.5元硬	0.5元硬（硬币器暂停收币；纸币器暂停收币；）
	1.0元	2.0元	3.0元	4.0元	9.0元	9.5元	10.0元
2元		√	√	√	√	√	√
2.5元						√	√
3.5元						√	√
5元					√	√	√
7元					√	√	√
10元							√
PAYOUT_IND	**1 .. 15元**	**1 .. 15元**	**1 .. 15元**	**1 .. 14元**	**1 .. 9元**	**1 .. 9元**	**1 .. 9元**

算法2应用举例（COST_IND扣款）：

COST_IND行为，与现金购物完全一样（只需要把商品“单价”，换成“扣款金额”即可）。也即，扣款后肯定会造成无法找零的情况，一律回复NAK_RPT。

注意：如果有用户投币，则如果扣款后肯定会造成无法找零的情况，一律回复NAK_RPT。

算法2应用举例（PAYOUT_IND出币）：

当没有用户投币，PAYOUT_IND的行为根据找零量判断即可。

当用户有投币，参见“算法2应用举例（现金购物）”PAYOUT_IND行！PAYOUT_IND行的数字表示“PAYOUT_IND”命令可出币金额的组合。

注意：当有用户投币的情况下，需要判断PAYOUT_IND出币后，是否会影响用户退币，只要会影响，一律回复NAK_RPT。

11.2 硬币找零顺序

见算法2。简单说，总是先从大面额找零，大面额不够，才考虑小面额。

11.3 现金扣款顺序

如果用户同时投入纸币和硬币，并且纸币存在暂存的情况下，则优先从非暂存金额中扣款。

举例：如用户投入纸币8元，其中3张1元纸币压钞，1张5元纸币暂存；此时用户购买价值2.5元的商品，成功出货后系统显示余额5.5元，此时退币，VMC需要退出1张5元暂存纸币和1枚5角硬币

11.4 纸币识别开关

通过“纸币器”提供的方法进行设置，打开/关闭某种面额纸币的识别。尽量使用简单的设定方式，比如MEI纸币器支持的“卡片”方式。

注：对于人民币，出厂是否默认关闭1元、20元纸币识别，根据具体需求定。

12. 自动找零时间设置

“自动找零时间”，指用户投币后，或发生购物后，用户没有再进行任何的投币或选货动作，用户投入的现金自动退出，这段时间即为自动找零时间。

举例：

假如运维设置的自动找零时间T0=3秒，自动找零时间分两个部分：

- 用户从余额0元开始投币，没有发生商品购买前，超时自动退币时间T=15+T0=18秒（15秒为初始投币固有等待时间）
- 用户进行了至少一件商品购买，超时自动退币时间T=T0=3秒

注意事项：

- 出货失败的情况下，对应商品的扣款，自动返还到用户余额
- 在自动找零时间段内，用户可以手工退币
- 时间设置范围：0-255
- 时间设置为0，表示购物后立即退币
- 时间设置为255，表示永不自动退币
- 如果计算的超时自动退币时间T大于255，则T=255（如 T=15+T0 = 15 + 254 = 269 大于255，则T=255）
- 小LCD显示方式：在小LCD上，同时显示用户余额和自动退币倒计时

13. 商品出货统计

出货类型	VENDOUT_RPT.type	出货量最大值(个)	收入最大值
全部出货	0~255	4294967295	4294967295
现金出货	0	4294967295	4294967295
游戏出货	1	4294967295	4294967295
刷卡出货	5、101~255	4294967295	4294967295
在线出货	2~4、6~100	4294967295	4294967295

“出货量”统计方式：

- 只统计VMC成功出货（VENDOUT_RPT.status=0）的情况，每出货一个，就增加1，当数量超过最大值4294967295后，从0开始计数。

  举例：原出货量统计是4294967295，当用户购买一瓶饮料后，当前出货量统计变为0

- 不区分具体货道、具体商品
- 机器出厂，“出货量”统计默认为0
- “恢复系统默认参数”不影响该值（也即出货量不会被清0）
- 出货量统计不提供*任何*清0方法，需要一直累加

“收入”统计方式：

- 只统计成功出货的情况，每出货一个，就取该商品的“售价*100”进行累加（这里的售价都是通过公式1算出的实际售价）：

  举例：原全部出货收入统计为300，现金出货收入统计为0。当用户购买一瓶3.5元的饮料后，当前全部出货收入统计根据公式“300+3.5*100=650”变为650；当前现金出货收入统计根据公式“0+3.5*100=350”变为350

  举例：原全部出货收入统计为650，游戏出货收入统计为0。当用户参与游戏中奖一瓶饮料后（假设饮料售价0.5元），当前全部出货收入统计根据公式“650+0.5*100=700”变为700；当前游戏出货收入统计根据公式“0+0.5*100=50”变为50

- 当收入超过最大值4294967295后，自动溢出

  举例：原全部出货收入统计为4294967293分，当用户购买一瓶3.5元的饮料后，当前全部出货收入统计根据公式“4294967293+3.5*100=347” 变为347

- 不区分货道、商品
- 机器出厂时，“收入”统计默认为0
- “恢复系统默认参数”不影响该值（也即收入不会被清0）
- 收入统计不提供*任何*清0方法，需要一直累加

本机显示“收入”和“出货量”：

- 本机中文LCD小液晶显示：

全部出货：12345678
全部收入：42949672
现金出货：xxx
现金收入：xxxx
游戏出货：xxxx
游戏收入：xxxx
刷卡出货：xxxx
刷卡收入：xxxx
在线出货：xxxx
在线收入：xxxx

注意：金额显示的时候，需要**先除以100**，再取整，故单位为各国货币的“元”

注意：显示较大金额时，需要能够自动换行；不能自动换行的，可以汉字和数字分两行显示。

- 本机LED数码管显示：
  - 对于只有数码管的机型，根据数码管位数确定出货量和金额显示的位数

举例（金额显示）：当前某收入统计为835299850，机器提供5位数码管，则通过LED查看数据，只显示52998【(835299850 / 100)%100000 = 52998】

举例（出货量显示）：当前某出货量统计为332533223，机器提供5位数码管，则通过LED查看数据，只显示33223【332533223%100000 = 33223】

- 对于如何通过键盘查看数据，根据具体机型再确定

“收入”和“出货量”通过INFO_RPT（type=6）上传到PC：

- 出货量数据，无需特殊处理，直接上传
- 金额数据，也无需特殊处理（无需对金额转换），直接上传

注意：“现金购买”统计类型中，包括VENDOUT_IND（TYPE=0）的现金购物情况，这种情况下，“收入”只需要累加VENDOUT_IND中的COST值

注意：“收入”统计，不包含COST_IND发起的扣款！

注意：如果出货的商品无法查询到售价，则认为商品售价为0

14. 现金投入/支出统计

现金类别	金额最大值
硬币投入	4294967295
纸币投入	4294967295
硬币出币	4294967295

现金投入：硬币：

- 只包含正常售卖模式下用户硬币投入金额
- 硬币器测试等维护目的的硬币投币，不计入该统计

现金投入：纸币：

- 只包含正常售卖模式下用户纸币投入金额
- 不计入“暂存”模式的纸币，必须是压钞成功的纸币才计入，压钞失败不计入该统计
- 纸币器测试等维护目的的纸币投币，不计入该统计

现金出币：硬币：

- 只包含正常售卖模式下硬币出币金额（含手工退币、PAYOUT_IND出币、VMC复位时的强制退币等）
- 硬币器测试等维护目的的硬币出币，不计入该统计

注1：本机显示现金投入、支出（参见“商品出货统计”、“附录A：售货机维护模式菜单”）

注2：数据上传到PC（参见“INFO_RPT.type=6”）

注3：现金出币（纸币）暂不考虑

注4：金额使用时机投币或出币面额乘以100存储

注5：现金统计不提供*任何*清0方法，需要一直累加

15. 游戏按键和购物按键

游戏和购物按键，均为用户按下按钮就上报消息（无需等待用户松手）；之后不再上报对应TYPE的BUTTON_RPT，直到用户松手，再按下去才上报新的 BUTTON_RPT！

注意：对于展示位选货按键的机型，当用户按动按键后，上报BUTTON_RPT（TYPE=2）

注意：对于键盘选货的机型，当用户通过键盘输入货道编号后，上报BUTTON_RPT（TYPE=1）

注意：游戏按键BUTTON_RPT消息，需要实现每秒连击2次而不丢失BUTTON_RPT！

16. 盒饭机

盒饭机需要保存最后同步的销售时间段：当盒饭机离线后，离线配置数据以这个时间段为准。

注意：当某一餐饭的售卖时间结束后，VM自己需要把自己货道中的商品余量自动清零。

盒饭机非售卖时间段，需要暂停服务，小LCD提示“非售卖时间”，并提示“售卖时间：xx:xx – yy:yy”

对于预定或不可售卖的商品，需要通过蜂鸣器和小LCD/LED提示用户（蜂鸣器0.4秒内响2声，小LED提示“盒饭已被预定”/ LED显示“booked”）。

- 机器所有商品只剩预定量，需要关闭纸硬币器
  - 由 PC 计算是否所有商品只剩预定量，并通过 CONTROL_IND（type=2）控制纸/硬币器关闭
- 盒饭机整机售卖停止的条件
  - 商品在有效售卖时间段内提前售空
  - 有效售卖时间结束
  - 其他参见《VMC“暂停服务”》
- 离线售卖/在线售卖
  - 在线售卖
    - VMC 需要发送 VENDOUT_REQ 请求，确定是否能出货
  - 离线售卖
    - 进入离线售卖模式，因为不再通过 PC 确认，用户发起购物请求，直接售卖（如果中途恢复在线售卖，此时 PC 应尽量确保被预定出去的盒饭被自动保留）
  - 注意：
    - 在离线售卖模式，售卖时间结束后，也需要停止盒饭机整机售卖（同时 VM 自己需要把自己货道中的商品余量自动清零）
- 盒饭机开始售卖
  - 盒饭机每一餐饭，开始售卖前，需要判断货道中有无商品余量，如果余量为 0，不启动本餐售卖。（在销售时间段内补货也有效）
  - 补货动作，可以在 VMC 与 PC 之间失去连接时，也可以进行，此时补货商品单最大数量由售货机自己判断（全部加满、按层加满、单货道配置）
- 盒饭机整机加热
  - 盒饭机需要根据售卖时间，自动提前 30 分钟开启加热
- 盒饭机投币上限
  - 参见《用户投币及找零》
- 盒饭机找零时机及自动找零时间
  - 参见《自动找零时间设置》
- 盒饭机投币上限
  - 参见《用户投币及找零》
- 现金购物与刷卡/在线购物冲突
  - 参见《现金购物和非现金购物》

- 强制购物如何处理
  - 参见《强制和非强制购物》
- 出货失败如何处理
  - 参见《出货失败和货道故障》
- 盒饭机需要支持的 CONTROL_IND 命令类型
  - type=2　纸硬币器开关
  - type=6　强制退币
  - type=16　游戏按键灯
  - type=17　系统时间

17. 恢复出厂默认参数

恢复出厂默认参数后：

- 售货机内部货道商品表默认全为 0xff，展示位商品表默认全为 0，商品售价表清空
- 整机处于“暂停服务”模式

附录 A:

A.1 售货机机型唯一码(magic1)

略

A.2 售货机维护模式菜单

说明：

>有检测开关或专用按钮的机型，直接进入维护模式；其他的机型，通过键盘输入 1->2->3->4 组合键进入维护模式

>不能将门开关作为*进入*维护模式的切换开关！

>平时需要将门开关作为退出维护模式的开关(但在“门开关测试”时需要特殊对待)

>20 秒内，用户没有任何动作（键盘操作或测试动作）自动返回售卖状态或暂停服务状态（“全系统同步”除外）

>维护模式下展示位灯全灭（除非特殊说明）

>根据机型特点，可以与友宝确认后，进行相应优化

>所有菜单内容，可以从上到下、或从下到上“循环”

>从子菜单返回后，需要自动定位到对应上级菜单

>按数字键，直接进入对应的项目

>任何时候，点击键盘“取消”，均返回上级

>对于“不支持”的提示，一秒钟后，自动返回上级

>每一次维护键盘按键，均快速“滴”一声

0. 销售统计	1. 总销售统计 （注:显示较大金额时，需要能够自动换行；不能自动换行的，可以汉字和数字分两行显示） （总销售统计不提供任何清零的方法，这些数据出厂默认为 0）	全部出货:12345678 全部收入:42949672 现金出货:xxx 现金收入:xxxx 游戏出货:xxxx 游戏收入:xxxx 刷卡出货:xxxx 刷卡收入:xxxx 在线出货:xxxx 在线收入:xxxx 1. 返回
	2. 总现金收支 （注：同“总销售统计”）	硬币投入:xxx 纸币投入:xxx 硬币出币:xxx
	3. 区间销售统计	硬币投入:xxx 纸币投入:xxx 硬币出币:xxx 总个数:xx 总金额:xx

<table>
<tr><td rowspan="4"></td><td rowspan="4"></td><td></td></tr>
<tr><td>1. 按货道统计>></td><td>输入货道号:xx
1. 货道 xx 个数:
2. 货道 xx 金额:
3. 下一货道
4. 上一货道
5. 返回（返回“按货道统计>>”）</td></tr>
<tr><td colspan="2">2. 清零区间销售统计</td></tr>
<tr><td colspan="2">3. 返回</td></tr>
<tr><td>4. 返回</td><td colspan="2"></td></tr>
<tr><td rowspan="11">1. 货道参数</td><td colspan="2">输入货道号:XX(货道编号)</td></tr>
<tr><td colspan="2">1. 货道编号:XX（不能编辑）</td></tr>
<tr><td colspan="2">2. 商品余量:X
（默认自动定位到该菜单项）
（默认显示当前余量，同时处于编辑模式）
（输入商品个数，最多不能超过两位）
（输入后，按确定，发送消息给 PC，如果 PC 回复 ACK，蜂鸣器“滴”一声，屏幕提示“成功”0.2 秒钟后返回；如果 PC 回复 NAK，蜂鸣器“滴滴”2 声，屏幕提示“失败”一秒钟后返回，处于非编辑模式；此时，如果按“确定”按钮，进入编辑模式；如果按“3”或“4”，进入下一个或上一个货道编辑模式）
(
举例：
正常情况：输入货道编号进入->编辑模式->输入数量->点击“确定”->发数据给 PC->收到 PC 响应，提示用户->退出编辑模式->按“3”->进入下一货道编辑模式（如果当前是最后一个货道，回到第一个货道）
提前退出编辑模式：输入货道编号进入->编辑模式->点击“取消”->退出编辑模式->按“3”->进入下一货道
)</td></tr>
<tr><td colspan="2">3. 下一货道</td></tr>
<tr><td colspan="2">4. 上一货道</td></tr>
<tr><td colspan="2">5. 商品编号:XX（不能编辑）</td></tr>
<tr><td colspan="2">6. 单价:XX.X（不能编辑）</td></tr>
<tr><td colspan="2">7. 货道状态:正常/故障（不能编辑）</td></tr>
<tr><td colspan="2">8. 返回（返回“输入货道号”）</td></tr>
<tr><td colspan="2">（提示:带“商品售空自动检测”的机型，“商品余量”不能编辑，且余量显示“0”或“1”）</td></tr>
<tr><td colspan="2">（提示:与该货道对应的展示位灯全亮；未对应的展示位灯全灭）</td></tr>
<tr><td rowspan="3">2. 货道状态查看</td><td colspan="2">故障货道 XX 个（一秒钟后进入子菜单）</td></tr>
<tr><td colspan="2">（显示格式如下，集中体现了货道商品余量和货道是否故障）

6 2 3 3 4 4 一 1 2 3 4
5 2 3 4 二 1 2 3 4
0 X 2 7 三 1 2 5 2 7
0 5 2 7 四 X 3 0 0
0 X 2 7 五 1 2 5 2 7
0 5 2 7 六 X 3 0 0

如上图第三层表示， 31 货道商品余量 0,32 货道故障，33 货道余量 2，34 货道余量 7，35 货道余量 12，36 货道余量 5，37 货道余量 2，38 货道余量 7。

注 1：中间汉字表示第几层，左边部分是每层的 1、2、3、4 货道，右边部分是 5、6、7、8 货道。
注 2：不同的机型可能的货道查看方式可以灵活处理，可协商。</td></tr>
<tr><td colspan="2">1. 返回</td></tr>
</table>

3.加满全部货道	确定加满吗？	
	1.确定	
	2.取消	
	（PC 回复 ACK，屏幕提示“成功”，蜂鸣器 0.2 秒内“滴”一声，一秒钟后返回上级）	
	（PC 回复 NAK，屏幕提示“失败”，蜂鸣器 0.4 秒内“滴滴”2 声，一秒钟后返回上级）	
4.补币完成	确定补币吗？	
	1.确定	
	2.取消	
	（PC 回复 ACK，屏幕提示“成功”，蜂鸣器 0.2 秒“滴”一声，一秒钟后返回上级）	
	（PC 回复 NAK，屏幕提示“失败”，蜂鸣器 0.4 秒内“滴滴”2 声，一秒钟后返回上级）	
5.全系统同步	确定同步吗？	
	1.确定	
	2.取消	
	（PC 回复 NAK，屏幕提示“正在同步中...”，VMC 继续发送同步请求）	
	（PC 回复 ACK，屏幕提示“成功”，蜂鸣器 0.2 秒“滴”一声，一秒钟后返回上级）	
6.按层加满	输入层号:X（只需输入一位，输入后自动进入）	
	1.确定	
	2.取消	
	（PC 回复 ACK，屏幕提示“成功”，蜂鸣器 0.2 秒“滴”一声，一秒钟后返回上级）	
	（PC 回复 NAK，屏幕提示“失败”，蜂鸣器 0.4 秒内“滴滴”2 声，一秒钟后返回上级）	
7.取币完成	确定取币吗？	
	1.确定	
	2.取消	
	（PC 回复 ACK，屏幕提示“成功”，蜂鸣器 0.2 秒“滴”一声，一秒钟后返回上级）	
	（PC 回复 NAK，屏幕提示“失败”，蜂鸣器 0.4 秒内“滴滴”2 声，一秒钟后返回上级）	
8.设备管理	1.整机故障查看 （检测整机的状态）	（文字方式列出当前机器所有故障，每行一个故障） （显示方式：比如故障代码为 222，故障内容为“压缩机故障”，显示方式如下） [222]压缩机故障 [223]xxx 故障
		1.返回
	2.电机测试 （仅针对类“弹簧机”，电机转一圈） （注:这里的测试结果需要反馈到“货道状态查看”）	1.单货道测试
		货道编号:XX（等待输入，输入两位后，自动开始）
		（显示过程）正在测试 xx
		（显示结果）货道 xx 正常/故障（1 秒钟后，自动返回上级）
		2.全货道循环测试（从第一个货道开始，到最后一个货道，每个货道轮流出货一次，无限循环）
		全货道测试
		正在测试 xx:正常/故障
		（按取消显示累计测试结果，单位“次”）
		总检测次数:xx（xx=所有货道转动的次数，包括故障次数）
		货道 xx 故障:n（xx 货道发生故障 n 次）
		……
		货道 yy 故障:n（yy 货道发生故障 n 次）
		1.返回
		3.返回
	3.出货测试 （完整出货过程，针对所有机型；如果出货确认模块没有被“禁用”，需包含出货确认）	1.单货道出货
		货道编号:XX（等待输入，输入两位后，自动开始）
		（显示过程）正在出货 xx
		（显示结果）货道 xx 正常/故障（1 秒钟后，自动返回上级）

	（注:这里的电机等“货道故障”结果需要反馈到“货道状态查看”）	2.全货道出货（从第一个货道开始，到最后一个货道，每个货道轮流出货一次，不循环）
		全货道出货
		正在出货 xx:正常/故障
		（按取消/或完成一轮后，显示累计测试结果，单位“次”）
		总出货次数:xx
		货道 xx 失败:xx（注:这里的失败，包括“货道故障”和“出货确认”没检测到）
		……
		货道 yy 失败:xx
		1.返回
		3.返回
	4.纸币器测试	纸币器测试
		已投金额:XX.X
		1.返回
		（收一枚币响一下）
	5.硬币器测试	硬币器测试
		已投金额:XX.X
		1.返回
		（收一枚币响一下）
	6.选货按钮测试	选货按键测试
		按键值:xx[yy]（yy=商品 ID）
		1.返回
		（按一次响一下） （注：仅针对含展示位的机型；无展示位机型无动作）
	7.选货指示灯测试	1.全勾测试（有货指示灯亮）
		2.全叉测试（无货指示灯亮）
		3.返回 （注：仅针对含展示位的机型；无展示位机型无动作）
	8.游戏按钮测试	游戏按钮测试
		按键次数:xx
		1.返回
		（按一次响一下，并计数）
	9.退币按钮检测试	退币按钮测试
		按键次数:xx
		1.返回
		（按一次响一下，并计数）
	10.硬币出币测试	出币测试
		1.返回
		（每次每个币种轮流退币一枚）
	11.门开关测试 （不支持的机型，提示“不支持”）	门开关测试
		开关门次数:xx
		1.返回
		（开关门一次响一下，并计数）
	12.出货确认测试	出货确认检测
		检测次数:xx
		1.返回
		（检测到一次响一下，并计数）
	13.照明测试	1.打开照明测试
		2.关闭照明测试
		3.返回

	14.压缩机测试（如果没有压缩机，或压缩机不能通过 VMC 控制，则提示“不支持”）	压缩机测试（注:压缩机测试需要按正确方式启动压缩机和对应风机）
		1.开启测试
		2.关闭测试
		3.返回
	15.加热测试（如果没有加热，或加热不能通过 VMC 控制，则提示“不支持”）	加热测试（注:加热测试需要按正确方式启动加热和对应风机）
		1.开启测试
		2.关闭测试（关闭时，需要注意加保护）
		3.返回
	16.找零余量查看（注:以美元举例，单位:个）	找零 1# 0.05:xx
		找零 2# 0.25:xx
		找零 3# 0.50:xx
		找零 4# 1.00:xx
		找零 5# 0.00:0
		找零 6# 0.00:0
		1.返回
	17.硬币类型查看	硬币 1# 0.05
		硬币 2# 0.25
		硬币 3# 0.50
		硬币 4# 1.00
		硬币 5# 0.00
		硬币 6# 0.00
		1.返回
	18.纸币类型查看	纸币 1# 1.00
		纸币 2# 2.00
		纸币 3# 5.00
		纸币 4# 10.00
		纸币 5# 0.00
		纸币 6# 0.00
		纸币 7# 0.00
		纸币 8# 0.00
		1.返回
	19.返回	
9.系统参数	1.温度查看（根据具体机型的特点，显示各箱体当前温度；不支持温度检测的机型，提示“不支持”；箱体温度传感器坏，提示“传感器坏”）	箱体 1:xx 摄氏度
		箱体 2:xx 摄氏度
		1.返回
	2.制冷温度:11（没有制冷功能的机型提示“本机不支持制冷”）（出厂默认:11）	设定范围 3-21 度
	3.加热温度:46（没有加热功能的机型提示“本机不支持加热”）（出厂默认:46）	设定范围 40-53 度
	4.加热制冷状态设定	（每种机型根据自己特点与友宝确定）

	（设定每个箱体是:常温/加热/制冷；“加热”和“制冷”都不支持的机型，提示“本机只支持常温”） （出厂默认:制冷）		
	5.自动退币:立即/永不/xx 秒 （出厂默认:5 秒）	1.超时退币:xx 秒（可编辑，编辑完后，变为“自动退币:xx 秒”）	
		2.立即自动退币	
		3.永不自动退币	
		4.返回	
	6.出货确认:启用/自禁用/人禁用，代表模块正常工作、自动禁用、人工禁用）（出厂默认:启用）	1.人工启用	
		2.人工禁用	
		3.返回	
	7.售卖时间 （出厂默认:3 个 Group 为全 00:00，周日到周一全设置为:无）	0.查看组 1 >> （注:组数据只能查看）	（点击“组 1”，进入这里） （共 7 组数据） 00:00-06:30 09:00-12:30 07:00-08:30 00:00-00:00 00:00-00:00 13:34-19:00 00:00-00:00 1.返回
		1.查看组 2 >>（同组 1）	
		2.查看组 3 >>（同组 1）	
		3.周日:组 1	
		4.周六:组 1	
		5.周五:无	
		6.周四:无	
		7.周三:无	
		8.周二:无	
		9.周一:无	
		10.强制 7x24 工作（执行后，周一到周日变为全天售卖，即把 3 个 Group 全设置为 00:00 的无效时间段）	
		11.返回	
	8.节能时间	与“售卖时间”相同	
	9.照明时间	与“售卖时间”相同	
	10.时间查看 （出厂默认:2011 年 1 月 1 日 0 点 0 分 0 秒）	2011 年 12 月 07 日	
		13:22:55 周三	
		1.返回	
	11.恢复出厂默认参数	恢复出厂默认吗?	
		1.确定 （点击确定后，整机配置项恢复出厂默认状态） （详见《恢复出厂默认参数》）	
		2.取消	
	12.关于本机	厂商:	
		型号:	
		软件版本:	
		硬件唯一编号:	
		离线数据量:	

		出厂日期:
		友宝规范:v3.1.1
		纸币器型号:
		硬币器型号:
		纸币货币代码:
		硬币货币代码:
		纸币基数:
		硬币基数:
		纸币小数位数:
		硬币小数位数:
		压缩机型号:
		加热器型号:
		返回
	13.其他设置 （需密码才能访问，密码:8253）	（不能出现“价格设定”、“货道”、“展示位”设定）
		厂商自己的其他设定 … （需要跟友宝确定）
	14.返回	

A.3 展示位按键购物和商品指示灯

注：这里的描述仅针对包含展示位的机型。

>所有情况下：

*售空的商品， “×”灯亮或“售空”灯亮。用户选择售空的商品，展示位灯无变化，小蜂鸣器0.4秒内“滴滴”2声；同时给PC发送对应按键报告

*用户按动未售空的商品按键，蜂鸣器均快速“滴”一声，同时给PC发送按键报告

>未投币的情况下：

*售空的商品，同“所有情况下->售空的商品”

*未售空的商品， “√”灯亮或“O”灯灭。用户按按键，按键对应展示位灯闪10秒钟（均匀亮灭20下）：

*在这10秒钟内，其他有货的商品展示位灯全灭，缺货的商品无变化；

*在这10秒钟内，如果用户选择了其他有货的商品，则原展示位灯停止闪烁，新展示位灯开始闪10秒

*在这10秒钟内，如果用户再次按同样的按键，则对应灯重新开始闪10秒

*在这10秒钟内，如果用户投币额大于展示位对应商品售价，VMC不能自动发起出货

*在这10秒钟内，如果用户选择了无货的商品，小蜂鸣器0.4秒内“滴滴”2声，其他无影响，也不给PC发送按键报告

*展示位商品未映射到货道，与“未售空的商品”相同

>已投币的情况下：

*售空的商品，同“所有情况下->售空的商品”

*未售空的商品，根据《用户投币及找零》对应的算法，可购买的商品“√”灯亮或“O”灯亮，不可购买的商品“√”灯灭或“O”灯灭；如果用户选择不可购买的商品，蜂鸣器0.4秒内“滴滴”2声

*展示位商品未映射到货道，与“未售空的商品”相同

*展示位灯的亮灭，需要根据用户投币快速动态调整

>维护模式：

*展示位灯全灭，按键蜂鸣器不发声，不发按键报告给PC

>暂停服务模式：

*展示位灯全灭，按键蜂鸣器不发声，不发按键报告给PC

A.4 小键盘购物

注1：这里的描述仅针对包含小键盘的机型。

注2：小键盘选货，需要提供用户输错反悔操作。

任何情况下，用户输入键盘按键，均需要“滴”一声。

>未投币：

用户未投币，直接输入货道 ID	货道：2 0.0 元	如果用户不再输入下一个字符，10 秒钟后自动消失。 如果用户输入的字符不在编号范围内，则 VMC“滴滴”2 声，10 秒钟后自动消失。
用户未投币，直接输入货道 ID	货道：23 3.5 元	输入货道第二个字符，VMC 自动显示货道商品对应的售价，10 秒钟后自动消失。 如果用户输入的货道不再编号范围，则 VMC“滴滴”2 声，10 秒钟后自动消失。

>已投币：

与“未投币”相似，当用户输入合法的货道，并且用户投币足够，则自动发起出货。

用户未投币，直接输入货道 ID	货道：2 0.0 元 余额:2.0 元 倒计时	输入第一个字符
用户未投币，直接输入货道 ID	货道：23 3.5 元 余额:2.0 元 倒计时	输入货道第二个字符。 如果商品售价大于用户余额，则 VMC“滴滴”2 声，10 秒钟后自动消失。 如果售价小于等于用户余额，则发起出货，进入“正在出货，请稍等”界面

>暂停服务：

小屏幕无特殊提示。

>维护模式：

小屏幕无特殊提示。

A.5 售货机工作模式提示信息

规范售货机在工作状态下，屏幕提示内容。

〈显示项目〉	〈LCD 显示内容〉	〈说明〉
上电启动中	〈显示 VMC 软件版本，固件版本〉	持续至少 2 秒钟
系统自检	系统自检中…	
离线数据传送	正在同步数据…	
离线数据同步完成	正在更新数据…	
系统启动完成	正常销售	
正常销售中，用户未投币	正常销售	用户未投币的情况下，“正常销售”与“xx℃”轮流显示：“正常销售”显示 6 秒，“xx℃”显示 3 秒
与“正常销售”轮流显示	2012-06-30 22:22 xx℃	箱体内温度。如果不支持温度或温度检测有问题，则不显示温度，只显示时间。 注：只有显示“正常销售”，才需要轮流显示该界面，其他情况不显示
用户未投币，直接选货	售价: xx.x 元	显示商品售价，10 秒钟消失。 注：键盘选货机型，提示方式见“小键盘购物”

用户投币	XX.X 元 倒计时	右下角有“自动找零时间”倒计时
用户选货但金额不足	投币金额不足	3 秒钟消失。 注：键盘选货机型，提示方式见“小键盘购物”
用户再投币	XX.X 元 倒计时	
用户选货且金额足够	正在出货，请稍等	如果有纸币暂存，只有压钞成功才显示“正在出货” 如果压钞失败，直接提示“压钞失败”，3 秒钟自动消失，并显示用户余额
出货中	正在出货，请稍等	
出货成功	请取商品	3 秒钟后消失
出货失败	未出货，请取退款	10 秒钟后消失
用户还有余额	XX.X 元 倒计时	右下角有“自动找零时间”倒计时
用户退币	正在出币，请稍等	
退币中	正在出币，请稍等	

退币完成	请取币	3 秒钟后消失
硬币器故障或被强制停用，纸币正常	XXXX 硬币停用	注意：如果是硬币器被拆除，不显示“硬币停用”
纸币器故障或被强制停用，硬币正常	XXXX 纸币停用	注意：如果是纸币器被拆除，不显示“纸币停用”
纸币器，硬币器均故障或均停用	XXXX 刷卡/在线支付	“停用”指纸币器硬币器工作的过程中发生故障，或被 PC 禁用
纸币器，硬币器拆除	XXXX	“拆除”指售货机没有安装纸硬币器，这时默认就是刷卡和在线购物
暂停服务模式	暂停销售 在线业务正常	注：在暂停服务模式，如果用户投币，需要正常提示投币金额，退币等信息
离线售卖模式	〈与正常售卖相同〉	离线售卖平时与正常售卖显示的内容相同。但在“纸币器，硬币器均故障或均停用”和“纸币器，硬币器均拆除”时，显示“暂停销售”

A.6 消息响应机制

此处模拟一台售货机从上电启动开始，中途经历一系列连贯动作的过程中，VMC与PC之间消息交互和等待方式。这些流程适用于综合机、饮料机、盒饭机。

-》售货机上电，硬件初始化

-》售卖了几件商品（离线售卖）

-》PC启动完成，同步数据给VMC

-》正常售卖（在线售卖）

-》模拟POLL消息超时2次

-》售卖（投币过程，出货过程，游戏中奖，退币过程）

-》正常售卖（空闲2分钟）

-》在线购物因商品无货被拒绝

-》用户投入10元纸币，触屏购物，压钞失败

-》用户重新投入10元纸币，压钞，出货成功，但出货过程与PC失去连接

-》进入离线模式（用户接着在离线模式，离线购买2件商品，退币）

-》离线模式，物流补货

-》离线模式另一用户投币购物，当购买第二个商品的时候，出货过程中正好赶上与PC握手成功

-》恢复到在线销售

-》正常售卖

-》无法识别消息的处理

-》扣款流程

-》退币流程

-》配置维护模式：包括进入、全部加满、单货道补货、补币、全系统同步、退出，中间的动作

-》退出维护模式，PC自动同步货道、售价等数据

-》节能、售卖、照明时间段同步

-》状态报告的主动上报

〈状态〉	〈SN〉	〈Message〉	〈说明〉
		〈220V 上电〉	VMC 上电启动、PC 上电启动
		〈N 秒〉	VMC 完成压缩机、纸硬币器、时钟、货道等初始化
			VMC 启动或复位完成，VMC 进入正常售卖，PC 仍在启动中。

	0	POLL	VMC 未与 PC 建立连接。 注：与 PC 失去连接，POLL 延时固定为 5 秒。 （假设：这段时间，用户购买了两件商品，均记录到离线销售记录中）
		<等待 1.5 秒>	
		<延时 5 秒>	
	0	POLL	
		<等待 1.5 秒>	
		<延时 5 秒>	
	0	POLL	
		<等待 1.5 秒>	
		<延时 5 秒>	
	0	POLL	
		<等待 1.5 秒>	
		<延时 5 秒>	
	0	POLL	
		<等待 1.5 秒>	
		<延时 5 秒>	
	0	POLL	
		<等待 1.5 秒>	
		<延时 5 秒>	
	0	POLL	
	0	GET_SETUP	PC 启动完成，首先获取 VMC 属性
		<VMC 收集数据>	
	1	VMC_SETUP	VMC_SETUP 不再自动上报，PC 根据需要使用 GET_SETUP 获取。
	1	ACK	
		<延时 500ms>	注：与 PC 建立连接后，POLL 延时改为 500 毫秒
第一条离线数据	2	POLL	PC发起离线数据同步，直到全部同步完成。 （VMC需要按照时间先后顺序上报离线数据，先上报type=0的离线数据，type=0的离线数据上报结束后，再上报type=1的离线数据）
	2	GET_OFFLINE_DATA	
	3	OFFLINE_DATA_RPT	
	3	ACK	
		<延时 500ms>	
第二条离线数据	4	POLL	
	4	GET_OFFLINE_DATA	
	5	OFFLINE_DATA_RPT	
	5	ACK	
	…(假设到 14)	…… （离线数据同步中）	
	15	POLL	
	15	GET_STATUS	
		<VMC 收集数据>	
	15	STATUS_RPT	
		<延时 500ms>	
	16	POLL	PC 自动配置售货机
	16	HUODAO_IND	
	16	ACK_RPT	
		<延时 500ms>	
	17	POLL	
无展示位的机型，无需发送	17	POSITION_IND	

该消息			
	17	ACK_RPT	
		<延时 500ms>	
	18	POLL	
	18	SALEPRICE_IND	
	18	ACK_RPT	
		<延时 500ms>	
	19	POLL	
	19	ACK	
		<延时 500ms>	
	20	POLL	
	20	ACK	
	…(假设到 43)	…	平时，就这样每 500ms 轮询
	44	ACTION_RPT.action=0	<VMC 每 10 秒左右无条件发一个心跳>
	…(假设到 47)	…	
	48	POLL	PC 周期获取盒饭机售卖时间
	48	GET_INFO.type=4	
注：仅盒饭机	49	INFO_RPT.type=4	PC 获取到数据后判断，如果 VMC 的数据与 PC 不一致，则 PC 需要下发 SALETIME_IND
		<延时 500ms>	
	50	POLL	
	50	SALETIME_IND	
	50	ACK_RPT	
	51	POLL	<模拟连续超时 2 次的场景>
		<等待 1.5s>	
	51	POLL	VMC 等待超时后，立即发起消息重传！PC_ERR=1
		<等待 1.5s>	
	51	POLL	再超时，再重传！PC_ERR=2
	51	ACK	PC_ERR=0 终于在第 3 次收到 ACK。 假设这个 ACK 也丢失，那 POLL 累计 3 次没有收到响应，VMC 需要自动转入离线售卖模式。
		<延时 500ms >	
	52	POLL	
	52	ACK	
		<等待 200ms>	
	53	PAYIN_RPT.dt=0	在第 200ms 时，用户投币：投币 1 元，总金额 1 元
		<延时 500ms>	
	54	POLL	从 VMC 最近的一条消息开始计算 500ms
	54	ACK	
		<延时 500ms>	
	55	POLL	

	55	ACK	
		<等待 100ms>	
	56	PAYIN_RPT.dt=0	投币 1 元，总金额 2 元
		<延时 500ms>	
	57	POLL	
	57	ACK	
		<延时 500ms>	
	58	POLL	
	58	CONTROL_IND.type=17	周期性地，PC 每 3 分钟同步一下时间
	58	ACK_RPT	
		<等待 300ms>	
	59	BUTTON_RPT.type=2	<模拟用户选货> <用户尝试购买 3.0 元的商品，但余额不足>
		<等待 50ms>	
	60	ACTION_RPT.action=0	<VMC 每 10 秒左右无条件发一个心跳>
		<延时 500ms>	
	61	POLL	
	61	ACK	
		<等待 120ms>	
	62	PAYIN_RPT.dt=0	投币 0.5 元，总金额 2.5 元
		<等待 400ms>	
	63	PAYIN_RPT.dt=0	投币 1 元，总金额 3.5 元
		<延时 500ms>	
	64	POLL	
	64	ACK	
		<等待 400ms>	
	65	PAYIN_RPT.dt=1	投币 1 元并直接压钞，总金额 4.5 元。 注意：如果 VMC 决定直接压钞就不用发送 PAYIN_RPT.dt=100
		<延时 500ms>	
	66	POLL	<模拟 PC 请求总金额的场景>
	66	GET_INFO.type=3	
	67	INFO_RPT.type=3	报告总金额=4.5 元
		<延时 500ms>	
	68	POLL	
	68	ACK	
		<延时 500ms>	
	…(假设到 79)	…	<用户正在思考要买哪个商品>
	80	POLL	
	80	ACK	
		<等待 200ms>	
	81	ACTION_RPT.action=0	
		<延时 500ms>	
	82	POLL	
	82	ACK	
		<延时 500ms>	

	83	POLL	
	83	ACK	
		<等待 220ms>	
	84	BUTTON_RPT.type=2	<模拟用户选货>用户选择一个商品，售价 3.0 元
注：盒饭机专用	85	VENDOUT_REQ	非盒饭机不需要发送 VENDOUT_REQ
	85	ACK	
	86	ACTION_RPT.action=1	出货开始，total_value=1.5 元,大概 seconds=3 秒
<A>		<等待 1000ms>	<这期间自动禁用纸硬币器> <出货过程中，不主动发送任何 F7=1 的消息，如果出货过程中有出币，则 VMC 缓存该出币报告，直到出货完成才发送> <收到 VENDOUT_IND、PAYOUT_IND、COST_IND 消息，直接 NAK_RPT 拒绝> <出货过程，不发送 BUTTON_RPT.type=1/2/3/4>
	87	ACTION_RPT.action=0	
		<等待 2000ms>	
	88	VENDOUT_RPT.type=0	出货成功，total_value=1.5 元
	88	ACK	VMC 自动打开纸硬币器
		<延时 500ms>	
	89	POLL	
	89	ACK	
		<延时 500ms>	
	90	POLL	
	90	ACK	
		<等待 200ms>	
	91	BUTTON_RPT.type=0	出游戏了，用户参与游戏
		<等待 300ms>	
	91	BUTTON_RPT.type=0	用户又拍了一次
		<延时 500ms>	
	92	POLL	
	92	ACK	
		<延时 500ms>	
	93	POLL	
	93	GET_STATUS	PC 周期主动获取 VMC 状态
	94	STATUS_RPT	
		<延时 500ms>	
	95	POLL	
	95	GET_HUODAO	PC 周期主动获取货道状态
	96	HUODAO_RPT	
		<延时 500ms>	
	97	POLL	
	97	CONTROL_IND.type=17	周期性地，每 3 分钟同步一下时间
	97	ACK_RPT	
		<延时 500ms>	
	98	POLL	
	98	VENDOUT_IND.type=1	用户中奖了！！
	98	ACK_RPT	
	99	ACTION_RPT.action=1	奖品出货开始，total_value=1.5 元,大概 seconds=3 秒

		<等待 3000ms>	见<A>点解释
	100	VENDOUT_RPT.type=1	出货成功，total_value=1.5 元
	100	ACK	VMC 自动打开纸硬币器
	101	HUODAO_RPT	刚刚出货的那个货道，刚好商品售空，故 VMC 主动上报一条 HUODAO_RPT 通知 PC
		<延时 500ms>	
	102	POLL	
	102	ACK	
		<等待 200ms>	
	103	BUTTON_RPT.type=4	用户手工退币
	104	ACTION_RPT.action=2	出币开始，total_value=1.5 元，两个 1 元一个 5 角硬币，大概 seconds=1.5 秒
<B>		<等待 1700ms> 注：如果在这个过程中，售货机自动退币，则发送的 PAYOUT_RPT 消息中，F7=0	<出币过程中，这期间自动禁用纸硬币器> <出币过程中，不主动发送任何 F7=1 的消息> <收到 VENDOUT_IND、PAYOUT_IND、COST_IND 消息，直接 NAK_RPT 拒绝> <出币拖长中，用户通过键盘购物，需要拒绝> <出币过程，不发送 BUTTON_RPT.type=1/2/3/4>
	105	PAYOUT_RPT.type=0	出币完成，total_value=0
	105	ACK	
		<延时 500ms>	
	106	POLL	
	106	ACK	
		<延时 500ms>	
	107	POLL	
	107	ACK	
	…(假设到 110)	…	平时，就这样每 500ms 轮询
	111	ACTION_RPT.action=0	<VMC 每 10 秒左右无条件发一个心跳>
		…	
	112	POLL	
	112	VENDOUT_IND.type=2	用户发起在线购物
	112	NAK_RPT	VMC 发现在线购物的商品为 0 个，拒绝出货指示
		<延时 500ms>	
	113	POLL	
	113	ACK	
		<延时 500ms>	
	114	POLL	
	114	ACK	
		<延时 500ms>	
	115	POLL	
	115	ACK	
		<等待 100ms>	
	116	PAYIN_RPT.dt=100	用户投入 10 元纸币，暂存，total_value=10.0
		<延时 500ms>	
	117	POLL	

	117	ACK	
		<延时 500ms>	
	118	POLL	
	118	VENDOUT_IND.type=0	用户发起触屏购物，cost=3.5
	118	ACK_RPT	total_value=10.0，大于 3.5，VMC 确认可以出货
<C>		<等待压钞完成>	<VMC 压钞过程中,VMC 需要临时禁用纸硬币器> <压钞过程，VMC 不发起 F7=1 的消息>
	119	PAYIN_RPT.type=101	压钞失败，暂存的纸币退出，total_value=0 （注：由于扣款失败没有发起出货，不发送 VENDOUT_RPT 消息）
	…	…	
	123	POLL	
	123	ACK	
		<延时 500ms>	
	124	POLL	
	124	ACK	
		<延时 500ms>	
	125	POLL	
	125	ACK	
		<等待 110ms>	
	126	PAYIN_RPT.dt=100	用户再次投入 10 元纸币，暂存， total_value=10.0
		<延时 500ms>	
	127	POLL	
	127	ACK	
		<延时 500ms>	
	128	POLL	
	128	VENDOUT_IND.type=0	用户发起触屏购物，cost=3.5
	128	ACK_RPT	total_value=10.0，大于 3.5，VMC 确认可以出货
		<等待 1.5 秒钟左右>	参考<C>点
	129	PAYIN_RPT.type=1	纸币压钞成功，total_value=10.0
	130	ACTION_RPT.action=1	出货开始，total_value=6.5 元,大概 seconds=3 秒
		<等待 3000ms>	参考<A>点
	131	VENDOUT_RPT.type=0	出货成功，total_value=6.5 元
		<等待 1.5s>	PC_ERR=1
	131	VENDOUT_RPT.type=0	
		<等待 1.5s>	PC_ERR=2
	131	VENDOUT_RPT.type=0	
		<等待 1.5s>	PC_ERR=3
		<离线售卖模式>	VMC 进入离线售卖，该 VENDOUT_RPT.type=0 的消息，作为第一条离线销售数据保存起来。VMC 自动打开纸硬币器。 在离线售卖模式下，与 PC 之间只发送 POLL 消息，其他消息如

			PAYIN_RPT、VENDOUT_RPT、BUTTON_RPT 等，均不再发送，直到 VMC 恢复到在线售卖模式。
		<额外延时 5s>	<用户在这段时间购买 2.5 元饮料一听，余额 4 元，离线售卖记录 2 条>
	132	POLL	每 5 秒重发一次（Retry each 5s）(PC_ERR=0)
		<等待 1.5s>	超时
		<延时 5s>	<用户在这段时间购买 2.5 元饮料一听，余额 1.5 元，离线售卖记录 3 条>
	132	POLL	
		<等待 1.5s>	<用户在这段时间退币，余额 0 元>
		<延时 5s>	
		…	<周期尝试与 PC 建立连接>
		<物流人员补货>	离线模式可以补货。
		…	
	132	POLL	
		<等待 1.5s>	用户投入 1 元硬币两枚
		<延时 5s>	用户投入 1 元纸币压钞，5 元纸币暂存，余额共 8 元
	132	POLL	
		<等待 1.5s>	购买 2.5 元商品，发起出货，余额 5.5 元，离线销售记录 4 条
		<延时 5s>	
	132	POLL	购买 3.0 元商品，发起出货，余额 2.5 元
	132	ACK	模拟 VMC 在出货过程中，与 PC 恢复通信
			收到 PC 的 ACK 了，但还在出货中，VMC 需要确保这次出货成功结束，并保存为离线销售记录
		<VMC 等待，直到本次出货结束>	出货成功结束，离线销售记录 5 条
		<延时 500ms>	
同步离线销售数据	133	POLL	PC将周期性发起离线数据同步。
	133	GET_OFFLINE_DATA	
	134	OFFLINE_DATA_RPT	
	134	ACK	
	…(假设到 140)	…	
		<延时 500ms>	
	141	POLL	离线数据同步完成，VMC 开始正常售卖
	141	ACK	
		<延时 500ms>	
	142	POLL	POLL 恢复到 500ms 一次
	142	ACK	
		<延时 500ms>	

	…(假设到150)	…	
			模拟双方无法识别消息的情况
	151	XXX_RPT	PC 无法识别这条消息
	151	NAK	如果这条无法识别的消息 F7=1，则 PC 回复 NAK
	…(假设到159)	…	
	160	POLL	
	160	YYYY	VMC 无法识别这条消息，丢弃
		<延时 500ms>	
	161	POLL	
	161	ACK	
	162	ACTION_RPT.action=0	心跳
		<等待 150ms>	
	163	PAYIN_RPT.dt=1	用户投入 1 元纸币，直接压钞
		<延时 500ms>	
	164	POLL	
	164	ACK	
		<等待 200ms>	
	165	PAYIN_RPT.dt=1	用户投入 1 元纸币，直接压钞，total_value=2.0 元
		<等待 400ms>	
	166	PAYIN_RPT.dt=100	用户投入 5 元纸币，暂存，total_value=7.0 元
		<延时 500ms>	
	167	POLL	
	167	PAYOUT_IND.type=2	用户发起出币两元
	167	ACK_RPT	
	168	ACTION_RPT.action=2	开始出币，total_value=7.0 元
		<等待出币结束>	参见<B>
	169	PAYOUT_RPT.type=2	出币成功，total_value=7.0 元
	169	ACK	
		<延时 500ms>	
	170	POLL	
	170	ACK	
		<延时 500ms>	
	171	POLL	
	171	ACK	
	…(假设到181)	…	
		<延时 500ms>	
	182	POLL	
	182	COST_IND.type=1	用户发起扣款指示，扣款 3 元
	182	ACK_RPT	

		<等待压钞结束>	压钞过程 VMC 不发送其他消息
	183	PAYIN_RPT.type=1	纸币压钞成功，total_value=7.0
	184	COST_RPT.type=1	扣款成功，total_value=4.0 元
		<延时 500ms>	
	185	POLL	
	185	ACK	
		<延时 500ms>	
	186	POLL	
	186	ACK	
		<延时 500ms>	
	187	POLL	
	187	CONTROL_IND.type=6	触屏发起找零
	187	ACK_RPT	
	188	ACTION_RPT.type=2	出币开始，total_value=4 元，4 个 1 元 2 秒
		<等待出币结束>	**参见<B>**
	189	PAYOUT_RPT.type=0	出币完成，total_value=0
	189	ACK	
		<延时 500ms>	
	190	POLL	
	190	ACK	
	…(假设到 200)	…	
	201	ADMIN_RPT.type=5	VMC 检测到开门（v3 规范中，无“检测开关”的机型无需上报）
		<可能的等待>	
	202	ACTION_RPT(type=5,value>0)	进入维护模式
		<延时 500ms>	
	203	ACTION_RPT(type=5,value>0)	
		<延时 500ms>	
	204	ACTION_RPT(type=5,value>0)	
		<延时 500ms>	
	205	ACTION_RPT(type=5,value>0)	
		<等待 200ms>	
	206	ADMIN_RPT.type=1	加满全部货道
	206	HUODAO_SET_IND	收到 PC 正确响应，屏幕提示“成功”（如果 PC 无法全部加满，PC 回复“NAK”，屏幕提示“失败”；如果 VMC 等待超时，无需重传，直接屏幕提示“失败”）
	206	ACK_RPT	
		…	
模拟等待超时	207	ADMIN_RPT.type=8	按层加满
		<等待 1.5s>	VMC 等待超时，无需重传，直接屏幕提示“失败”
		<其他消息略>	
模拟无	207	ADMIN_RPT.type=8	
	207	NAK	PC 无法全部加满，PC 回复“NAK”，屏幕提示“失

法加满			败”
		<其他消息略>	
模拟正常响应	208	ADMIN_RPT.type=8	
	208	HUODAO_SET_IND	收到 PC 正确响应，屏幕提示“成功”
	208	ACK_RPT	
	…(假设到 210)	…	
	211	ADMIN_RPT.type=2	修改货道存量
	211	ACK	收到 ACK，VMC 提示“成功”，蜂鸣器快速“滴”一声；同时将该存量值自动存入 VMC 存储器
		<延时 500ms>	
	212	ACTION_RPT(type=5,value>0)	
		<等待 100ms>	
	213	ADMIN_RPT.type=2	修改货道存量
	213	NAK	收到 NAK，VMC 提示“失败”，蜂鸣器快速“滴滴”2 声；同时该存量不保存到 VMC 存储器
		<等待 400ms>	
	214	ADMIN_RPT.type=3	补硬币完成
	214	ACK	收到 ACK，VMC 提示“成功”
		<延时 500ms>	
	215	ACTION_RPT(type=5,value>0)	
		<延时 500ms>	
	216	ACTION_RPT(type=5,value>0)	
		<延时 500ms>	
	217	ACTION_RPT(type=5,value>0)	
		<等待 300ms>	
全系统同步	218	ADMIN_RPT.type=4	全系统同步
	218	NAK	收到 NAK，VMC 提示“正在同步中…”
		<延时 500ms>	
	219	ADMIN_RPT.type=4	
	219	NAK	
		<延时 500ms>	
	220	ADMIN_RPT.type=4	
		<等待 1.5 秒>	等待超时，依然提示“正在同步中…”
		<延时 500ms>	
	220	ADMIN_RPT.type=4	
		<等待 1.5 秒>	
		<延时 500ms>	
	220	ADMIN_RPT.type=4	
		<等待 1.5 秒>	
		<延时 500ms>	
	220	ADMIN_RPT.type=4	
		<等待 1.5 秒>	
		<延时 500ms>	

	220	ADMIN_RPT.type=4	
		<等待 1.5 秒>	
		…	无论 PC 回复 NAK，还是超时，VMC 都需要一直重试，一直提示“正在同步中…”，并且 VMC 不能自动进入“离线售卖”
	220	ADMIN_RPT.type=4	
		<等待 1.5 秒>	
		<等待 500ms>	
	220	ADMIN_RPT.type=4	
		<等待 1.5 秒>	
		<延时 500ms>	
	220	ADMIN_RPT.type=4	
	220	NAK	
		<延时 500ms>	
	221	ADMIN_RPT.type=4	
	221	NAK	
	…(假设到 253)	…	
	254	ADMIN_RPT.type=4	
全系统同步成功	254	ACK	终于收到 ACK 了，VMC 提示“成功”，蜂鸣器“滴”一声
	255	REQUEST.type=4	当 VMC 发现全系统同步成功完成后，需要主动发起 REQUEST 请求，获取 PC 中最新的数据。数据包括货道关系、展示位关系和商品售价。 提示：这里演示了请求 type=5 数据时，出现了超时，这是需要进行重传，直到收到为止。
	255	HUODAO_IND	
	255	ACK_RPT	
	0	REQUEST.type=5	
		<等待 1.5s>	
	0	REQUEST.type=5	
	0	POSITION_IND	
	0	ACK_RPT	
	1	REQUEST.type=6	
	1	SALEPRICE_IND	
	1	ACK_RPT	
		<延时 500ms>	
	2	ACTION_RPT(type=5,value>0)	
		<延时 500ms>	
	3	ACTION_RPT(type=5,value>0)	
	…(假设到 18)	…	其他维护模式动作
			恢复正常工作模式
退出维护模式	17	ACTION_RPT(type=5,value=0)	无任何操作，20 秒后，VMC 自动退出维护模式
		<延时 500ms>	
	18	POLL	退出维护模式后，PC 自动发起数据同步。
	18	HUODAO_IND	
	18	ACK_RPT	
		<延时 500ms>	

	19	POLL	
	19	POSITION_IND	
	19	ACK_RPT	
		<延时 500ms>	
	20	POLL	
	20	SALEPRICE_IND	
	20	ACK_RPT	
		<延时 500ms>	
	21	POLL	
	21	ACK	
		<延时 500ms>	
	22	POLL	
	22	ACK	
		<延时 500ms>	
	23	POLL	
	23	ACK	
		<延时 500ms>	
	24	POLL	
	24	GET_STATUS	PC 周期获取 VMC 状态
	25	STATUS_RPT	
		<延时 500ms>	
	26	POLL	
	26	GET_HUODAO	PC 周期获取货道状态
	27	HUODAO_RPT	
	…(假设到 40)	…	
			模拟 PC 同步工作时间段给 VMC
	41	POLL	
	41	CONTROL_IND.type=7	PC 更新 VMC 的节能工作时间段
	41	ACK_RPT	
		<延时 500ms>	
	42	POLL	
	42	CONTROL_IND.type=8	PC 更新 VMC 的售卖工作时间段
	42	ACK_RPT	
		<延时 500ms>	
	43	POLL	
	43	CONTROL_IND.type=9	PC 更新 VMC 的照明工作时间段
	43	ACK_RPT	
		<延时 500ms>	
	44	POLL	
	44	ACK	
	…(假设到 50)	…	
			模拟 VMC 主动上报状态报告
	51	STATUS_RPT	VMC 检测到纸币器发生故障，主动发送一条状态报告

		<延时 500ms>	
	52	POLL	
	52	ACK	
		<延时 500ms>	
	…(假设到 60)	…	
	61	POLL	
	61	ACK	
	62	STATUS_RPT	VMC 检测到纸币器又自己恢复了，主动发送一条状态报告
		<延时 500ms>	
	63	POLL	
	63	ACK	
		<延时 500ms>	
	64	POLL	
	64	RESET_IND	VMC 收到 PC 的复位消息。 VMC 需要执行一次真正的物理复位。
		<假设 VMC 重启一次的时间为 10s>	在这段时间内，VMC 重启
	0	POLL	
	0	ACK	
		<延时 500ms>	
	1	POLL	
	1	ACK	
		<延时 500ms>	