Smellodi Odor Display

protocol specification v0.1

0. System overview

The Smellodi odor display consists of several (six in its initial version) networked odor modules and a network bridge presenting a UART-over-USB interface (virtual COM port) to the PC [1]. Due to latency and synchronization requirements, the bridge collates data coming from and going to individual network nodes. This makes the network look more like a single device with several subunits instead of a group of individually addressable devices.

The initial version of the device has two kinds of odor modules:

- Normal odor modules (5), providing one odor each. They have one mass flow controller, one valve for turning odor on/off, odor source temperature measurement and a chassis heater/thermometer.
- The base module, providing the basic clean air flow for the other sources. It also includes an air humidifier as the sixth odor. The base module has more sensors than the other modules: in addition to common odor module's measurements, it includes two humidity sensors (incoming air and output air), a pressure sensor and a PID sensor. It also includes two mass flow controllers, the other being the controller for dry (dilution) air.

The later expanded version will include one additional module, which is used to dilute the scented air for human participants. It might also include more odor modules, up to ten.

The bridge includes some system-wide control features, namely fan and PID sensor power supply enable/disable functionality.

1. General

The virtual serial port runs at 230400bps, using 8 bits per character, no parity and one stop bit. Flow control is not used. Instead, it is expected that the PC sends only one packet at a time and waits for device's reply before sending another. It is recommended to send packets to the device using atomic writes, as the device's receiver will reset itself if there's more than 100ms between characters. (100ms wait can be used to intentionally reset the receiver, too.) PC, in turn, should wait at least 140ms before assuming the response packet was lost. The PC software should be designed to handle at least 10kB/s sustained incoming data rate.

The protocol uses fixed-format, variable length packets. The most transactions are initiated by the PC and the device reacts to them by sending a response packet, indicating request's success or failure. If additional response data is required, the device sends it as a separate packet, before sending acknowledge. There is one exception to this, namely automatically triggered measurements, where the device sends periodical measurement data without PC interaction.

[1] Windows should be able to install the drivers automatically, but should the automated install fail, the drivers can be downloaded from https://ftdichip.com/drivers/vcp-drivers/.

Preamble (24b)	Type (8b)	From (8b)	To (8b)	Payload size (16b)	(Optional payload)	Check (8b)

Figure 1. Generic packet format.

ID	Value	Direction	Description
ACKNOWLEDGE	0xFA	to PC	Responses to commands
QUERYVERSION	0x70	to device	Query version information
VERSION	0x71	to PC	Hardware, software, and protocol versions
QUERYDEVS	0x50	to device	Query installed odor modules
DEVS	0x51	to PC	List of installed odor modules
QUERYCAPS	0x40	to device	Query odor module's capabilities
CAPS	0x41	to PC	List of odor module's capabilities
SET	0x20	to device	Set odor module actuators
SYSTEMSET	0x60	to device	Enable/disable fans and PID lights
DATA	0x31	to PC	Odor module measurement data
STARTSTOP	0x80	to device	Start, stop, or trigger measurement
RESET	0x90	to device	Reboot the odor modules

Table 1. List of supported packet types.

The generic packet structure is shown in figure 1. All the fields are always present, except the 'optional payload' field. The existence and length of this field is determined by the 'payload size' field: if it is zero, there is no payload data.

The 'preamble' section consists of three 0xCC characters. The receiving state machine can (and should) use this field to regain synchronization in case of random data loss. However, it should be noted that the preamble pattern ***can*** be present in the payload field. The likelihood for this is relatively low, < 0.01% with the longest expected packets, but still high enough that it can't be used as the sole way of detecting packet start.

The 'type' field can be any of the values listed in table 1. Note that all the packets are strictly directional. Sending unrecognized packets results in an error. For example, the bridge won't recognize ACKNOWLEDGE packets and will respond to them by sending an ACKNOWLEDGE packet with applicable error code.

The packets include addressing fields ('from' and 'to'). However, PC can't directly address any of the odor modules and attempts to do so are ignored by the bridge. Instead, the PC must always direct its packets to the bridge device ('to' = 0xF0, 'from' = 0xF1). Conversely, the bridge will always send its packets to PC ('to' = 0xF1, 'from' = 0xF0).

The 'payload size' field, like mentioned above, specifies the payload field length in bytes. It can be zero, meaning that no payload field is included. This field is 16b long and it is sent least significant byte first. In other words, it is sent as a little-endian value, like all the other values requiring more than 8 bits. The longest payload the bridge can receive is 300 bytes and the longest packet it can send is 990 bytes.

The packets end in an 8b checksum, which is calculated by summing all the packet's bytes except the preamble fields, incrementing the value by one and then bit-wise inverting the result.

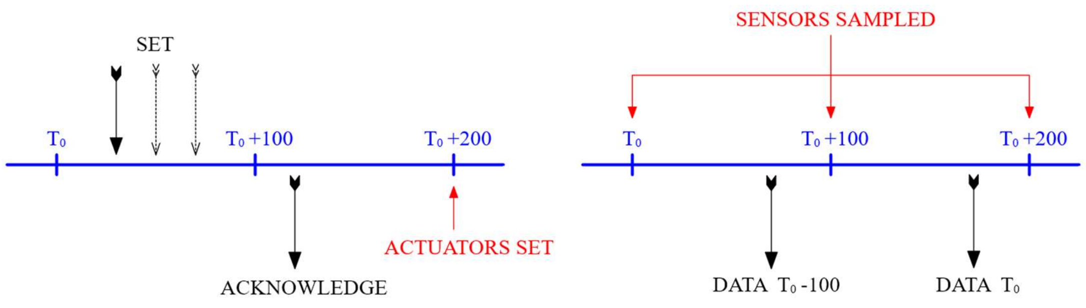

Figure 2. Actuator set timing (left) and measurement timing (right).

2. Arbitration

The system uses a 10Hz / 100ms master clock. On every tick, the bridge sends new actuator values to the odor modules and, if measurement is running, reads the samples and sends them to the PC.

1. When PC sends new actuator values to the odor modules (figure 2, left side):
    - They are sent to the odor modules on the next 100ms boundary. Depending on PC's timing relative to master clock, this can be delayed up to 100ms.
    - New values are acknowledged somewhat after that.
    - The actuators are set on the next 100ms tick.

2. When measurement is running (figure 2, right side):
    - All sensors are sampled on every 100ms tick.
    - The values are latched (and the next cycle is started) on the next tick.
    - The values are then sent to the PC.

Thus, if PC sets any actuators, it might take over 400ms before they are reflected in measurement data. PC can try to compensate for the pipelined part of the delay, but it is probably not worth it.

3. Detailed packet list

3.1 ACKNOWLEDGE (0xFA) packet

The device sends an ACKNOWLEDGE packet as a response to every received valid packet (correct address, correct checksum). This packet has one byte payload, representing an error code as an 8b signed integer. The possible error codes are listed in table 2.

Error code	Value	Description
ERR_OK	0	Request was executed successfully
ERR_INVVAL	-10	Parameter has invalid value
ERR_NOTAVAIL	-11	Resource is not available
ERR_OUTOFMEM	-12	There isn't enough memory for the operation
ERR_INVMODE	-13	Requested operation is not possible in current operating mode
ERR_TIMEOUT	-14	Operation timed out
ERR_NODATA	-15	There's no data
ERR_UNKPACK	-16	Unknown or unsupported packet type
ERR_INVLEN	-17	Invalid amount of data
ERR_INVIDX	-18	Invalid index or device number
ERR_BUSY	-19	Device is busy and can't perform the requested operation
ERR_ERROR	-128	Non-specific error

Table 2. Error codes.

3.2 QUERYVERSION (0x70) packet

Requests version information. This packet has no payload data. Device will respond by sending a VERSION packet first, then ACKNOWLEDGE:ERR_OK. Malformed requests (extraneous data present) result in ACKNOWLEDGE:ERR_INVLEN and no VERSION packet.

3.3 VERSION (0x71) packet

Response to the QUERYVERSION packet. The payload consists of 3 bytes. The first byte is the hardware version number, the second is the software version number and the last one is the protocol version. The upper nibble of each byte represents the major version number, and the lower nibble is the minor version. The initial version of the device has value 0x10 for all fields, signifying version 1.0 for HW, SW and protocol.

3.4 QUERYDEVS (0x50) packet

Queries the attached odor modules, including the base module and optional dilution module. This packet has no payload data. The bridge responds to a proper request by first sending a DEVS packet and then ACKNOWLEDGE:ERR_OK. If request is incorrect, the response will be ACKNOWLEDGE: ERR_INVLEN alone.

3.5 DEVS (0x51) packet

Response to the QUERYDEVS packet. The payload is an 11 elements long Boolean vector represented as 11 bytes, where non-zero values represent the installed odor modules. The initial version of the device will have the base module and the first 5 generic odor modules present, like table 3 suggests. Dilution module will be added later. At the time of writing, it is unclear whether modules 6…9 will ever be installed and if yes, what kind of modules they will be.

3.6 QUERYCAPS (0x40) packet

Queries the specified odor module's (0…10, see the DEVS packet indices in table 3) capabilities. The device to be queried is presented as a single-byte payload. If the specified odor module is installed, the response will be a 17 bytes long vector representing 17 Boolean values, followed by ACKNOWLEDGE: ERR_OK. The indices are presented in table 4. If the specified device does not exist, response is ACKNOWLEDGE:ERR_NOTAVAIL. If the index is over 10, the response is ACKNOWLEDGE: ERR_INVVAL and if the payload length is incorrect, the response is ACKNOWLEDGE:ERR_INVLEN.

Index	Use
0	Base module present
1	Generic odor module 1 present
2	Generic odor module 2 present
3	Generic odor module 3 present
4	Generic odor module 4 present
5	Generic odor module 5 present
6	(Odor module 6 present)
7	(Odor module 7 present)
8	(Odor module 8 present)
9	(Odor module 9 present)
10	(Dilution module present)

Table 3. Indices to DEVS packet Boolean vector.

Index	Use
0	PID sensor present
1	Bead thermistor present (breathing detector)
2	Chassis temperature sensor present
3	General purpose thermometer 1 present (odor source temperature)
4	General purpose thermometer 2 present
5	Humidity sensor 1 present (incoming air humidity)
6	Humidity sensor 2 present (output air humidity)
7	Pressure sensor present (output pressure)
8	Mass flow controller 1 present (odorant flow measurements)
9	Mass flow controller 2 present (dilution air flow measurements)
10	Valve controller 1 present (odorant valve state feedback)
11	Valve controller 2 present (output flip valve state feedback)
12	Mass flow controller 1 present (odorant flow set value input)
13	Mass flow controller 2 present (dilution air set value input)
14	Chassis heater present (temperature controller set value)
15	Valve controller 1 present (odorant valve control)
16	Valve controller 2 present (output flip valve control)

Table 4. Indices to CAPS packet Boolean vector. Indices 0…11 are for sensors, 12…16 are for actuators. Descriptions in parentheses indicate the typical uses for each sensor and actuator.

Dev (8b)	Type (8b)	Value (32b)	Type (8b)	Value (32b)	Dev (8b)	Type (8b)	Value (32b)	Dev (8b)	… …

Figure 3. SET packet general format.

3.6 CAPS (0x41) packet

Response to the QUERYCAPS packet. Packet contains 17 bytes long payload, representing 17 Boolean values. These Booleans indicate the presence of various sensors and actuators. See table 4 for detailed descriptions and sections 3.7 and 3.9 for data type descriptions.

3.7 SET (0x20) packet

The SET packet is used to adjust mass flow controller flow rates, open/close valves and adjust chassis heater set point. One SET packet can contain set values for several odor modules, if necessary. The system applies all the given values simultaneously to all the specified modules. On success, bridge responds with ACKNOWLEDGE:ERR_OK, but this response can be delayed up to 100ms. See section 2 about bus arbitration details. If there are errors in packet's formatting, device responds with ACKNOWLEDGE:ERR_INVLEN (incorrect payload length), ACKNOWLEDGE:ERR_INVIDX (indices out of bounds) or ACKNOWLEDGE:ERR_BUSY (previous SET command is still in progress). Bus errors are indicated with ACKNOWLEDGE:ERR_TIMEOUT (should be very rare; in this case one or more odor modules have failed to receive the new setup) and invalid set values with ACKNOWLEDGE: ERR_INVVAL.

The payload follows the format presented in figure 3. It starts with an 8b long device index 'dev' (see table 3), logically ORed with value 0x80. After this field one or more 'type' – 'value' pairs follow, until another 'dev' field or payload end is encountered. The 'type' field of each set value refers to the actuators (indices 12…16) in tables 4 and 5. The 'type' field itself is 8b long and while the 'value' fields could be of various lengths, all the currently used types are 32b long (and in little-endian format).

Index	Length	Type	Interpretation
12	32b	float	Normalized mass flow set value in range 0…1
13	32b	float	Normalized mass flow set value in range 0…1
14	32b	float	Chassis heater set point in Celsius, range 0…100
15	32b	long int	Valve 'on' time in milliseconds (positive values), valve permanently off (zero) or permanently on (negative values)
16	32b	long int	Valve 'on' time in milliseconds (positive values), valve permanently off (zero) or permanently on (negative values)

Table 5. Actuator set value types and their meanings.

Index	Length	Type	Interpretation
0	32b	float	PID sensor voltage in volts.
1	64b	2 × float	Bead thermistor values; first resistance in ohms, then sensor input voltage in volts. Resistance value will be infinite if the sensor is not connected.
2	32b	float	Chassis thermometer temperature in Celsius.
3	32b	float	Odor source thermometer value in Celsius.
4	32b	float	General purpose thermometer value in Celsius.
5	64b	2 × float	Incoming air humidity values; first relative humidity in percent, then air temperature in Celsius.
6	64b	2 × float	Output air humidity values; first relative humidity in percent, then air temperature in Celsius.
7	64b	2 × float	Output pressure sensor: first pressure in millibars, then temperature in Celsius.
8	96b	3 × float	Odor (or humidifier) MFC readings; first flow rate in SLPM, then temperature in Celsius and finally pressure in millibars.
9	96b	3 × float	Dilution air MFC readings; first flow rate in SLPM, then temperature in Celsius and finally pressure in millibars.
10	8b	boolean	Odorant valve state (on or off)
11	8b	boolean	Output swap valve state (on or off)

Table 6. Sensor value types and their meanings.

3.8 SYSTEMSET (0x60) packet

This packet can be used to turn some system-wide features on or off. The packet has a two-byte payload, each of them representing a Boolean value. The first Boolean turns the system fans on or off and the second value enables or disables all the PID sensors. The response is normally ACKNOWLEDGE: ERR_OK, except when payload has wrong length. In that case ACKNOWLEDGE:ERR_INVLEN is returned. Note that the fans are initially (after power-on) enabled, while the PID sensors are off.

3.9 DATA (0x31) packet

If measurement is enabled (see STARTSTOP packet), device sends DATA packets at 10Hz rate. These packets strongly resemble the SET packets: each odor module's data is preceded by an 8b 'dev' field (formed by ORing module number with 0x80) and the data is presented as 'type' – 'value' pairs. The 'type' field is 8b and refers to table 4's numbering (entries 0…11). The 'value' field length and interpretation vary. Table 6 lists the measurement types.

Time (32b)	Dev (8b)	Type (8b)	Value (8…96b)	Type (8b)	Value (8…96b)	Dev (8b)	Type (8b)	… …

Figure 4. DATA packet general format.

The primary differences to SET packets are that all the measurement data is preceded by a 32b 'time' field (long integer representing milliseconds since measurement start) and that the sensor value fields are more variable. Figure 4 shows the DATA packet payload's general format.

3.10 STARTSTOP (0x80) packet

The STARTSTOP packet can be used to start a measurement (so that the bridge will output DATA packets at 10Hz rate; see section 2) and to stop a measurement. It can also be used to trigger a single measurement in polled manner. The packet has one-byte payload:

- If value is 0, measurement is stopped.
- If value is 1, measurement runs continuously.
- If value is 2, one measurement is performed.

The response to this packet is ACKNOWLEDGE:ERR_OK if payload length is correct and the payload value is one of the allowed values. If payload length is incorrect, response will be ACKNOWLEDGE: ERR_INVLEN and if the value is out of bounds, ACKNOWLEDGE:ERR_INVMODE.

3.11 RESET (0x90) packet

The option to reboot the system is provided primarily for debugging purposes, as there shouldn't be any need for it during normal operation. Successful execution of RESET takes around 1500ms. The bridge ***does not*** send an ACKNOWLEDGE packet after a successful reboot. However, if the RESET packet has a spurious payload present, bridge responds with ACKNOWLEDGE:ERR_INVLEN and does not reset the system.

4. Programming considerations

4.1 Finding device serial port automatically

The simplest way to auto-detect the device serial port is to open all available serial ports in sequence and attempting to connect to them (see section 4.2). However, this method can be somewhat slow on computers with several serial ports present. It also results in garbage being sent to the other ports and is aesthetically unpleasant.

One alternative is to query the USB device information before connecting. The Smellodi odor printer's manufacturer string is 'TUNI', and the product description string is 'Smellodi Odor Printer'. The latter is present in Windows' "Bluetooth & other devices" applet as well. One way to read these strings is to use FTDI's D2XX drivers instead of the VCP drivers. (It is unclear if the D2XX drivers are installed automatically with VCP drivers, though.) Windows itself provides a way to read these strings too (as evidenced by the applet), but author of this document has not experimented with that.

4.2 Connecting to the device

The following procedure is recommended for connecting to the device:

1. Open serial port.
2. Send STARTSTOP packet to stop sampling.
3. Wait at least 140ms.
4. Flush serial port reception buffer and reset reception routines.

5. Send QUERYVERSION packet.
6. Wait for VERSION and ACKNOWLEDGE packets at least 140ms.
7. If both packets are received correctly and they have expected contents, device is ready for use. Otherwise steps 2…6 can be retried. Retrying more than once is unlikely to help.

This guarantees proper synchronization for both ends and verifies that the connected device is the Smellodi odor printer. Steps 2…4 guarantee interface reset on both ends, while the remaining steps verify that the correct device is present.

4.3 Setting actuators

It is possible to set several actuators simultaneously, like described in section 3.7. The system guarantees that all the set values contained in one SET packet are loaded to all the addressed odor modules almost simultaneously (within 2ms). In particular, the valve controllers across different odor modules are synchronized accurately. However, the mass flow controllers can take tens of milliseconds to set. The result is that flow settings change slightly after the valves change their states. More importantly, if odor module has two mass flow controllers, the second controller is updated after the first controller and can thus be up to 50ms late.

4.4 Running a measurement

Measurements can be performed both in polled and automated manner. The use of automated measurement is strongly recommended, as it allows higher data rates with guaranteed timing and because it does not require constant low-latency attention from PC. The polled measurement is primarily intended for taking occasional, non-time critical snapshots. These snapshots can be used to make application UI more informative, for example.