Data Protocols v2

1 Overview
    1.1 Unit Addresses
2 MID ↔ Sensor Hub
    2.1 Commands
    2.2 Sensor Hub Reply
        2.2.1 Readings Reply Data
            2.2.1.1 Temp and humidity reply
            2.2.1.2 Wind speed reply
            2.2.1.3 Tachometer reply
            2.2.1.4 K Thermocouple reply
            2.2.1.5 Barometric Pressure reply
            2.2.1.6 Barometric Pressure Wide reply
            2.2.1.7 Differential Pressure Wide reply
            2.2.1.8 Multipoint Temp Sensor
                2.2.1.8.1 Temperature Reading
                2.2.1.8.2 Address Reading
            2.2.1.9 Unit Version reply
            2.2.1.10 Get Calibration Value reply
        2.2.2 Successfully Executed Command (Reply Code 2)
        2.2.3 Error Codes
3 Sensor Hub ↔ Sensor Unit
    3.1 Changing Address
    3.2 Calibration
        3.2.1 Get Calibration
        3.2.2 Set Calibration
MID ↔ WWW
    JSON config download format (MID ← WWW)
    JSON reading data upload format (MID → WWW)
        Error codes

1 Overview

This document describes the data protocol for communication between EA custom hardware in the Isadore dryer monitoring system.

1.1 Unit Addresses

Each sensor unit has a 16 bit address stored in the programming ROM. The unit's addresses are 16 bit for two reasons:

1. With new RS485 chips it is possible to have 256 or more units on one port. It also makes it easier to have a installation of units of all different addresses and there wouldn't be a need to make sure they didn't put one with the same address on the same port.
2. It would make it easier to use replacement units with less of a probability of same address. It is a similar reason why ethernet MAC addresses are 48 bit, no one has 2**48 network cards on a single switch.

2 MID ↔ Sensor Hub

The communication between Master Interface Device and Sensor hub will be done through a UDP/IP connection on Port 1082. The Master Interface Device will then send a magic code, command, and optional data (in a single packet), the Sensor Hub will reply to the MID by sending a single UDP packet to the port and address the original command came from. All data is in binary and little endian.

From Master Interface Device:

Magic (4B)	Command (1B)	Data ...

Reply from Sensor Hub:

Reply code(1B)	Data...

The magic is 4 bytes:

Magic in hex	0x44 0x45 0x52 0x56
Magic as little endian 4 byte int	1448232260
Magic as ASCII	DERV

2.1 Commands

The commands the master interface device can send to the sensor hub are listed below. The command is a one byte number and is followed by data depending on the command. *Note: the*

port numbers are one-based, not zero-based.

Command	Code	Data	Version notes
Get temp&hum reading(*)	1	Port,N,addr1,addr2,...,addrN ● Port is 1 x uint8 ● N is 1 x uint8 and is a value indicating how many addresses will be given. N needs to be less than or equal to 32 and greater than 0. If there are more than 32 devices connected you need to get the readings by querying again afterwards with another set of <=32 addresses. ● addrX is 1 x uint16 and corresponds to a Sensor Unit connected to the port on the Sensor Hub.	
Get wind speed reading	2	Same as code 1	
Get Tachometer	3	Same as code 1	
Drive Hz & Current	4	Fill This	
SP&PV and thermostat	5	Fill this	
K thermocouple reading	6	Same as code 1	
Barometric Pressure	7	Same as code 1	
Barometric Pressure wide	8	Same as code 1	Unit>=3.00
Differential Pressure wide	14	Same as code 1	Unit>=3.01
Multipoint Temp sensor reset	9	Same as code 1	Unit>=3.00
Multipoint Temp sensor reading channel 1	10	Same as code 1	Unit>=3.00

Multipoint Temp sensor reading channel 2	11	Same as code 1	Unit>=3.00
Multipoint Temp sensor reading channel 3	12	Same as code 1	Unit>=3.00
Multipoint Temp sensor reading channel 4	13	Same as code 1	Unit>=3.00
Multipoint Temp address reading chanel 1	14	Same as code 1	Unit>= 3.02
Multipoint Temp address reading chanel 2	15	Same as code 1	
Multipoint Temp address reading chanel 3	16	Same as code 1	
Multipoint Temp address reading chanel 4	17	Same as code 1	
General Unit Reading	25	Code,R,Port,N,addr1,addr2,...,addrN ● Code is 1 x uint8 from this table ● Port is 1 x uint8 ● R is 1 x uint8 length in bytes of the reply per address (max 8) ● N is 1 x uint8 number of addresses N <= 32 ● addrX is 1 x uint16 sensor unit address Note: Code becomes new command code.	Unit>=v3.00 Hub >=v3.00
Unit version	63	Same as code 1	Unit>=v3.01
Calibration Value	64	Code,R,Port,N,addr1,addr2,...,addrN ● Code is 1 x uint8 from this table ● Port is 1 x uint8 ● R is 1 x uint8 length in bytes of the reply per address (max 8) ● N is 1 x uint8 number of	Hub>=3.03

		addresses N <= 32 ● addrX is 1 x uint16 sensor unit address	
Set Calibration Value	65	Code,D,Port,A,V ● Code is 1 x uint8 from this table of what calibration value to set. ● D is 1 x uint8 length in bytes of the calibration value V (max 8) ● Port is 1 x uint8 ● A is 1 x uint16 sensor unit address ● V is 1 x D bytes	Hub>=3.03
Ping	130	1 x uint16	
Hub version	150	1 x uint16	Hub>=3.03
Reserved	200	Reserved for unit change address (Section 3.1)	

* - Time between measurements should be greater than 1s on per sensor basis to reduce the temperature sensor from heating up and reporting the heat of the chip instead of the air.
** - may not be implemented in prototype

2.2 Sensor Hub Reply

The sensor hub will give a reply after a command has been given. The reply may be any number of the following depending on the command. For example, a port reading command may have a reply of several error codes (Code 4) before a reading reply (Code 1). When the sensor is done replying it will disconnect the TCP/IP connection.

Reading time
The maximum time it takes for a unit to get temperature and humidity is 824ms although almost always it will be much faster. The base Hub will wait a maximum of N*4s. While the hub board is waiting it won't be able to process any other commands so the timeout should be greater than 128 seconds (32*4s) for the MID if the timeout needs to be set to a static value, or N*4s if it can be set dynamically. N is the number of units addressed.

Reply	**Code**	**Data**
Readings Reply	1	S,CC,DATA ● S 1xuint8, size in bytes of DATA ● CC 1 x uint8, original command code given

		• DATA is S bytes, see *Readings Reply* section below.
Success executed command	2	N,echo • N is 1 x uint8 • echo is data given in command. N is number of bytes in echo
Pong, response to Ping	3	P • P is 1x uint16 and is the number given by Ping incremented by 1.
Error	4	L,C_1,A_2,..,C_L,A_L • L is 1 x uint8 • C_X is 1 x uint8 • A_X is 1 x uint8 L<=N See Error Codes section for more details.
Hub version reply	5	V • V is 1xuint16

2.2.1 Readings Reply Data

Type	CC	DATA
Temp and humidity reply	1	N,T1,H1,T2,H2,...,TN,HN • N is 1 x uint8 • TX is 1xuint16 • HX is 1 x uint16 N should match N in initial command. See *Temp and humidity reply* section for more details.
Wind speed reply	2	N,V1,V2,...VN • VX is 1 x uint16 See *wind speed reply* section for more details.

Tachometer reply	3	N,R1,R2,...,RN ● RX is 1 x uint16 See *Tachometer reply* section for more details.
Drive Hz & Current reply	4	V,I Fill out
SP&PV and Thermostat set value reply	5	Fill out
K thermocouple reply	6	N,W1,B1,W2,B2,...,WN,BN ● N is 1 x uint8 ● WX is 1 x uint16 TC 1 temp ● BX is 1 x uint16 TC 2 temp See K Thermocouple reply section for more details.
Barometric Pressure reply	7	N,P1,P2,...,PN ● N is 1 x uint8 ● PX is 1 x uint16 See Barometric Pressure reply section for more details.
Barometric Pressure wide reply	8	N,P1,P2,...,PN ● N is 1 x uint8 ● PX is 1 x uint32 See Barometric Pressure wide reply section for more details.
Differential Pressure wide reply	14	N,P1,P2,...,PN ● N is 1 x uint8 ● PX is 1 x uint32 See Differential Pressure wide reply section for more details.
Multipoint Temp Sensor reset reply	9	N,P1,P2,...,PN ● N is 1 x uint8 ● PX is 1 x uint8 See Multipoint Temp Sensor section below.
Multipoint Temp sensor reading address channel 1-4 reply	14-17	N,A1,A2,...,AN ● N is 1 x uint8 ● AX is 1 x uint64 See Multipoint Temp Sensor section below.
Multipoint Temp sensor reading temperature channel 1-4 reply	10-13	N,P1,P2,...,PN ● N is 1 x uint8 ● PX is 1 x uint16

		See Multipoint Temp Sensor section below.
Unit version	63	N,V1,V2,...,VN ● VX is 1 x uint16 See Unit Version reply section for more details
Calibration value	64	N,V1,V2,...,VN ● VX is 1 x R See Calibration value reply section for more details
Reserved	200	Reserved (Section 3.1)

2.2.1.1 Temp and humidity reply

The temperature and humidity readings that are sent back are raw values replied back in the same order given in the command. Some calculations will have to be done.

Temperature Calculation
Note: This conversion works for both SHT-75 and PressureV4.1 sensors.
Temperature data is given to the MIB as uint16 (T), to convert this number into Fahrenheit[1]:

$F = d_1 + d_2 T$
$d_1 = -40.2$
$d_2 = 0.018$

Humidity Calculation
There are two sensor that give humidity the SHT-75 and PressureV4.1 module. An attempt was made to make the pressure module compatible was it proved difficult to keep the firmware size under 8k.

Humidity data is given to the MIB as uint16 (H),

PressureV4.1
$RH_{true} = H/100.0$
Example H might be 3418
$RH_{true} = 3418/100.0 = 34.19\%$

[1]See section 4.3 in SHT7X datasheet, sensor running at 5V.

SHT-75

The following conversions when SHT-75 is in use is listed below[2].

$$RH_{linear} = c_1 + c_2 H + c_3 H^2$$
$$c_1 = -2.0468$$
$$c_2 = 0.0367$$
$$c_3 = -1.5955 \cdot 10^{-6}$$

RH_{linear} is relative humidity if temperature is 77F, to compensate for temperature you can use the formula below.

$$RH_{true} = \left(T_c - 25\right)\left(t_1 + t_2 H\right) + RH_{linear}$$
$$t_1 = 0.01$$
$$t_2 = 0.00008$$
$$T_c = \left(T_f - 32\right)/1.8$$

T_f is temperature in Fahrenheit.

Unit unresponsive

If both temperature and humidity uint16 values are zero (0x0000) for a unit then that will indicate there was a problem getting the data for that addresses. An error reply code would also be given for that address.

Reference documents

- Introduction to Relative Humidity http://www.sensirion.com/en/pdf/product_information/Introduction_to_Relative_Humidity_E.pdf
- Humidity Formulae http://www.sensirion.com/en/pdf/product_information/AN-humidity-formulae.pdf
- SHT7X datasheet http://www.sensirion.com/en/pdf/product_information/Datasheet-humidity-sensor-SHT7x.pdf

2.2.1.2 Wind speed reply

V is the value from the unit, a rolling average of up to 30 30s time periods

[2]See section 4.1 and 4.2 in SHT7X datasheet.

To find V in mph:
$V_{mph} = V/100.0$

2.2.1.3 Tachometer reply

The replied value R is the last value of rpm calculated by the sensor. The following formula may be useful.

Calculate frequency (Hertz)
$f = R/60$

2.2.1.4 K Thermocouple reply

The following formula should be used to get the temperature in Celsius.

D is the integer value read.

$T_c = 1023.75\frac{D}{2^{12}}$

2.2.1.5 Barometric Pressure reply

The following formula should be used to get the absolute pressure in kilo pascals.
D is the integer uint16 read.

$P = 0.0022888 \cdot D + 50$

It doesn't matter which sensor is in the unit the MPX or KP the conversion to a common function is done in the unit.

2.2.1.6 Barometric Pressure Wide reply

The following formula should be used to get the absolute pressure in kilo pascals.
D is the integer uint32 read. D comes in units of in Pascals.

$P = D/1000.0$

Calibration info

- R - 4 (int32)
- V - value in pascals, positive or negative

TODO: Not implemented

2.2.1.7 Differential Pressure Wide reply

The following formula should be used to get the differential pressure in kilopascals.
D is the integer uint32 read. D comes in units of centipascals. Note: This can report a negative pressure differential.

$$P = (D - 2147483647)/100000$$

Calibration info

- R - 4 (int32)
- V - value in centipascals, positive or negative

TODO: Not implemented

2.2.1.8 Multipoint Temp Sensor

The multipoint temperature sensor module can takes cables with many temperature sensors on them.

2.2.1.8.1 Temperature Reading

Reading temperature values should go like the following:

1. X is the channel we want to read (1,2,3,4)
2. Multipoint Temp Reset (code 9) sent to units
3. Multipoint Temp reading (code [9+X]) sent to units
4. Repeat step 3. until all units report back 0xFFFF
5. If desire to read from another channel change start at step 1.

It is incorrect to change channels without a reset, it will give unexpected results.

On a multipoint temperature cable there are a number of ds18b20 sensors. When the unit gets the code to read the temperature it probes for a ds18b20 address, then reads the temperature from that address. When the unit gets the code again it will probe for the next ds18b20 address and read the temperature. At the point when there are no more ds18b20 temperature sensors on the cable it will reply back the value 0xFFFF. It will continue to reply 0xFFFF until a reset command is issued.

The order in which a value is read from a particular sensor is dependent on the sensor's address and not the position the sensor is on the cable. The order will be the same for a particular cable. Thus sensors can be identified by query index, and one can manually map a location and query index.

The multipoint temperature sensor module supports up to four channels (Up to four multipoint sensor cables). You should never take readings from a different channel without sending a reset

first.

The number of sensors on a cable is not fixed, so you can have some units or channels with more or less sensors than others. This is why you should do readings until all sensors reply back 0xFFFF, while ignoring the units that reply with a 0xFFFF reading (This means you have reached the end of the sensors on that particular channel for that unit).

The following algorithm should be used to get the temperature in Celsius.
'ds18b20_value' sent to the function is the integer uint16 read.

```
double ds18b20_conversion(uint16_t ds18b20_value)
{
	uint8_t digits;
	uint8_t decimals;
	uint16_t t = ds18b20_value;
	double negative = 1.0;
	if (t & 0xF800) {
		negative = -1.0;
		t=~t+1;
	}
	digits = (uint8_t)((t&0x07F0) >> 4);
	decimals = t & 0x000F;
	return negative*((double)digits + (double)decimals/16.0);
}
```

If you want to test your conversion function, you can test with these values listed:

Value(Hex)	**Temperature (C)**
0x07D0	125
0x0550	85
0x0191	25.0625
0x00A2	10.125
0x0008	0.5
0x0000	0
0xFFF8	-0.5
0xFF5E	-10.125

0xFE6F	-25.0625
0xFC90	-55

2.2.1.8.2 Address Reading

Readings addresses is done in a similar fashion to reading the temperature. The size of the data returned is different(8 bytes instead of 2 bytes) and the command code is different.

Reading address values should go like the following:

1. X is the channel we want to read (1,2,3,4)
2. Multipoint Temp Reset (code 9) sent to units
3. Multipoint Temp reading (code [13+X]) sent to units
4. Repeat step 3. until all units report back 0xFFFFFFFF
5. If desire to read from another channel change start at step 1.

It is incorrect to change channels without a reset, it will give unexpected results.

2.2.1.9 Unit Version reply

V is integer uint16 read. V value to version

V	**Version**
1	3.00 + V/100

Unit version before 3.01 do not support the version command code.

2.2.1.10 Get Calibration Value reply

See each code to see what the reply value means. Will get an error response back in calibration is currently not supported for a command code in the unit version.

2.2.2 Successfully Executed Command (Reply Code 2)

This reply command is only a valid reply for **non-reading**, and **non-ping** commands. For example this is a valid reply for command codes 5 and 6. You will **not** get this reply for command codes 1, 2, 3, 4, 7.

2.2.3 Error Codes

The error reply code data consists is composed of the following:

L(1B)	C_1(1B)	A_1(1B)	...	C_L(1B)	A_L(1B)

The number of bytes in the data part of a error reply is 2*L+1. C_X is a code indicating the error listed in the below table. A_X is the index of the address of the sensor unit that had the error. If A_1=1 then the sensor unit from the first address given in command has the error C_1 associated with it. If A_X=0 this means it is a general error that doesn't belong to any sensor unit.

Note: More error codes will probably get added later.

Code	**Error**	**Includes Address**	**Recommended Action**
1	Ethernet board responsive but rest of sensor hub timed out.	No	Try request again.
2	Invalid number of addresses to query on port reading.	No	Try request again.
3	Reply size to large. This probably shouldn't happen, it would indicate a bug somewhere in the microcontroller code.	No	Try request again.
4	Sensor unit timed out or unit not on port. Address is the sensor unit that timed out.	Yes	Throw measurement away.
5	Unsupported command code	No	
6	Sensor unit's reply failed CRC8 check.	Yes	Throw measurement away.
7	Unit reply size mismatch (wacked out unit, or wrong firmware or something).	Yes	Throw measurement away.
8	Bad magic from MID.	No	Try request again.
9	Hub timed out. (MID generated)	No	Try request again.
10	Unit is not configured with sensor (Ex. Asked for wind speed when not configured with	Yes	Throw measurement away.

	anemometer).		

3 Sensor Hub ↔ Sensor Unit

The sensor hub will send a request addressed to a particular sensor unit on the correct port. The sensor unit and only that sensor unit will then respond through the RS485 network. Communication between sensor hub and sensor unit is binary little endian.

Request:

Magic (4B)	Address (2B)	CC (1B)

Reply:

Magic (4B)	Address (2B)	CC(1B)	S (1B)	Data (SB)	CRC8 (1B)

3.1 Changing Address

This only works for v3 sensor units

Request:

Magic(4B)	00 (2B)	200 (1B)	New Address (2B)

Reply:
None

3.2 Calibration

3.2.1 Get Calibration

Request:

Magic (4B)	Address (2B)	64(1B)	CC (1B)

Reply:

Magic (4B)	Address (2B)	64(1B)	S (1B)	Data	CRC8

				(SB)	(1B)

3.2.2 Set Calibration

Request:

Magic (4B)	Address (2B)	65(1B)	CC (1B)	V(DB)

Reply:

Magic (4B)	Address (2B)	65(1B)	S (1B)	V (SB)	CRC8 (1B)

MID ↔ WWW

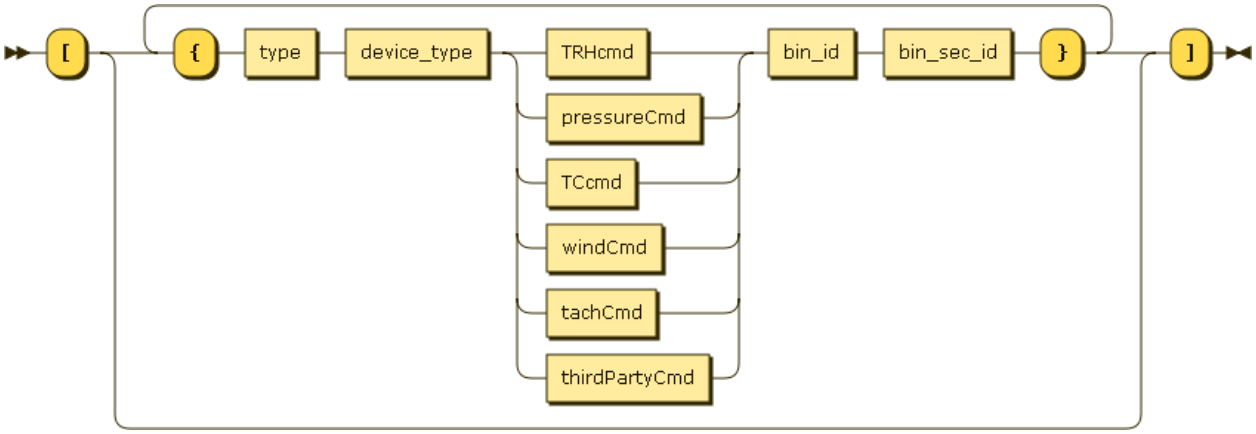
JSON config download format (MID ← WWW)

queryInterval:

midPass:

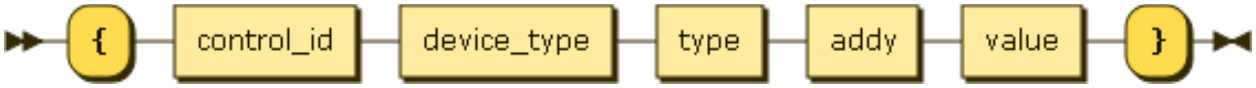
enabled_p:

commandInfo:



RC:

RCitem:



TRHcmd:

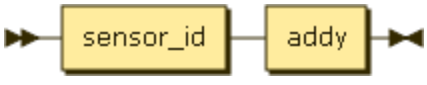
pressureCmd:

TCcmd:

windCmd:

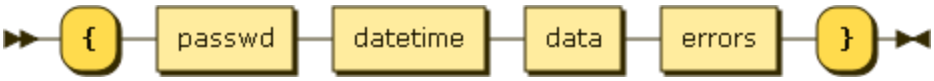
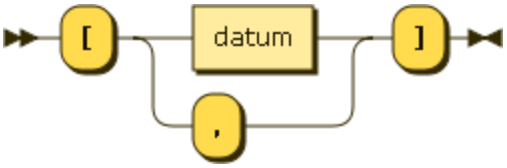
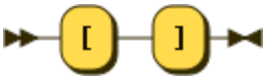
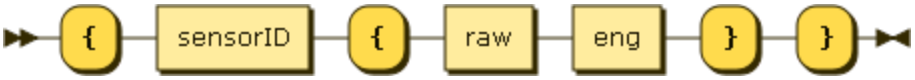
tachCmd:

thirdPartyCmd:



Addy:

JSON reading data upload format (MID → WWW)[3]

MIDtoWWW	
passwd	
datetime	
data	
errors	
datum	

Error codes

Code	Description	Recommended action
1-10	See table of MID/hub error codes	
51	Error encountered while parsing hub response	Throw measurement away

[3]Created using http://bottlecaps.de/rr/ui. EBNF syntax saved in MID repository.

52	Error encountered while computing engineering value	Keep raw value, throw engineering value away