Serial communication

Output (device version)

On the picture below is shown a sample list of messages that could be sent by device:

```
@<CR>                               //    begin of the list
I 00 0128 A03000020000 78<CR> //    sensor #0: identifier
V 00 028F 37<CR>                    //    sensor #0: temperature
I 01 0102 A03000020000 CA<CR> //    sensor #1: identifier
V 01 0280 37<CR>                    //    sensor #1: humidity
$<CR>                               //    end of the list
```

The list begins and ends with ‘@’ and ‘$’ character suitably. A single message is separated with ‘\r’ character and begins with ‘I’- device identifier or ‘V’- value from sensor. The rest of characters in a message compose a string of hexadecimal representation of binary data.

The last binary byte is a CRC of whole single message in binary representation calculated together with beginning character.

Message type	Message structure (with respect to size)	Description

‘I’	typedef struct // 10 bytes
	{
	unsigned char idx; // index
	unsigned char type; // device type
	unsigned char family; // device family
	unsigned char serial[6];// serial number
	unsigned char crc; // 8 bit CRC
	} dev_id;
		device identifier

‘V’	typedef struct // 4 bytes
	{
	unsigned char idx; // index
	short val; // value format
	unsigned char crc; // 8 bit CRC
		sensor value

Die technischen Informationen in dieser Dokumentation wurden von uns mit großer Sorgfalt geprüft und sollen über das Produkt und dessen Anwendungsmöglichkeiten informieren. Die Angaben sind nicht als Zusicherung bestimmter Eigenschaften zu verstehen und sollten vom Anwender auf den beabsichtigten Einsatzzweck hin geprüft werden. Etwaige Schutzrechte Dritter sind zu berücksichtigen. Ausgabe 2000-05-17. Dieses Datenblatt ersetzt alle früheren Auflagen.

	`}     dev_val;`	

Output (network version)

On the picture below is shown a sample list of messages that could be sent by device:

```
@<CR>                                   //    begin of the list
M 01 A501 00 08FC 37<CR>                //    sensor #0: temperature
M 01 A501 01 2710 14<CR>                //    sensor #1: humidity
$<CR>                                   //    end of the list
```

The list begins and ends with ‘@’ and ‘$’ character suitably. A single message is separated with ‘\r’ character and begins with ‘M’. The rest of characters in a message compose a string of hexadecimal representation of binary data.

The first binary byte (after ‘M’) is a message code; the last one is a CRC of whole single message in binary representation calculated together with beginning character ‘M’.

Message code	**Message structure (with respect to size)**	**Description**

0x01	`typedef struct                  //    6 bytes`
	`{`
	`      src_addr          addr; //    network address`
	`      unsigned char     idx;  //    index`
	`      signed short      val;  //    value format`
	`      unsigned char     crc;  //    8 bit CRC`
	`}     value;`
	
	`typedef struct                  //    2 bytes`
	`{`
	`      unsigned char     domain:1;`
	`      unsigned char     node:7;`
	`      unsigned char     subnet;`
	`}     src_addr;`
		sensor value

Device types

Code	**Device type**	**LonMark® SNVT**	**SNVT index**	**range**

0x00	unknown sensor	-	-	-
0x01	temperature sensor	SNVT_temp_p	105	-273.17 .. +327.66 C (0.01 C). The value 0x7FFF represents invalid data.
0x02	relative humidity sensor	SNVT_lev_percent	81	-163.84% .. 163.83% (0.005%). The value 0x7FFF represents invalid data.
0x0A	lux level sensor	SNVT_lux	79	0 ..65,535 lux (1 lux)
0x0B	PIR sensor	SNVT_count	8	0 .. 65,535 counts (1 count)
0x0C	wind speed sensor	SNVT_speed	34	0 .. 6,553.5 meters/sec (0.1 m/s)

Device families

Code	Device family
0x00	not used / unknown family
0x01	1308 multi-sensor
0x02	1311 multi-sensor
0x04	1307 humidity sensor
0x10	Dallas DS1820 temperature sensor
0x28	Dallas DS18B20 temperature sensor