Cairsens - UART Version

Communication Protocol

Measured data download

Table of Contents

I UART Port settings 3

II Queries / answers structures between UART cairsens and host 3

III Life 3

IV HeaderUart and TrailerUart definitions 4

V Cyclic redundancy checks 4

1 Compute 4
2 Sample in c 4

VI REF definition 6

1 Product option 6
2 Coefficient 6
3 Gaz Identification 7
4 Measure range 7
5 Interface type 7
6 Example 8

VII Reading of the instant value of the Cairsens (GetValue) 9

1 Query 9
2 Answer 10
3 Example 1 byte by value 10
4 Example 2 bytes by value 12

VIII GetDownload structure for cairsens (Stored data download) 13

1 Query 13
2 Answer 14
3 Example 10 minutes data 1 byte by value 15
4 Example 10 minutes data 2 bytes by value 17

1 UART Port settings

Baud rate	data bits	Parity	Stop bits	Flow control
9600	8	N	1	none

2 Queries / answers structures between UART cairsens and host

The structure of the query / answer frame passing between the Cairsens and the host can be defined by a series of bytes, the number of which varying and being represented in hexadecimal.

The query frames have a fixed length and are structured as follows:
SYNC STX LG REF DATA CRC ETX

The answer frames have a fixed length and are structured as follows:
SYNC STX LG REF DATA END CRC ETX

Bytes definition is:
- SYNC = Synchro Word
- STX = Start Frame
- LG = Length of Data
- REF = Cairsens identification (Serial Number)
- DATA = [CMD+PARAM] (Series of bytes for command and parameters)
- END =End Frame
- CRC [2 bytes/LSB First]
- ETX
- LIFE = Life used

Synchronization and start frame bytes have the following values and constant number of bytes:
- SYNC = 1 byte = 0xFF
- STX = 1 byte = 0x02
- CRC = 2 bytes
- END = 2 bytes = LIFE 0xFF
- ETX = 1 byte = 0x03

3 Life

Byte value	
0x00	the sensor can't return it's % life used
0x80	0 % of life used (New sensor)
..	..
0xC0	50 % of life used
..	..
0xE0	75% life used
..	..
0xFF	100% life used (end of life)

4 HeaderUart and TrailerUart definitions

In the following section of this document, the series of bytes representing SYNC STX LG and a part of CMD will be referred to as **HeaderUart** and will be defined by:

- HeaderUart = SYNC, STX, LG, 0x30, 0x01, 0x02, 0x03, 0x04, 0x05, 0x06

In the same idea, the series of bytes representing END will be referred to as **TrailerUart** and will be defined by:

- TrailerUart = END CRC ETX

5 Cyclic redundancy checks

5.1 Compute

The CRC code is calculated by dividing the binary sequence representing the frame by the following polynomial:

$X^{16} + X^{12} + X^5 + 1$

5.2 Sample in c

```c
#include <stdio.h>

unsigned int FCRC( unsigned char Frame[],unsigned char lg)
{
  unsigned int Poly = 0x8408;
  unsigned int Crc;
  unsigned char j,i_bits,carry;
  Crc=0;
  for (j=0;j<lg; j++) {
    Crc = Crc ^ Frame[j];
    for ( i_bits = 0; i_bits < 8; i_bits++ ) {
      carry = Crc & 1;
      Crc = Crc/2;
      if(carry) {
        Crc = Crc ^ Poly;
      }
    }
  }
  return Crc;
}

int main(int argc, char* argv[])
{
    unsigned int i;

    unsigned char Frame[] = {0xFF, // Synchro Word
    0x02,// Start Frame
    0x13,// Length of Data
     0x30,0x01,0x02,0x03,0x04,0x05,0x06,
     0xff,0xff,0xff,0xff,0xff,0xff,0xff,0xff,
    0x12, // CMD
    0x00,0x00, // CRC [2 bytes/LSB First]
    0x03}; // End Frame
```

```
    unsigned int StartPos = 2; // start position CRC

    printf ( " Frame without CRC =" );
    for( i = 0 ; i < sizeof( Frame) ; i++ )
    {
        if( i > 0 ) putchar(',');
        printf ( " 0x%02X" , Frame[i] );
    }
    putchar('\n');

    i = FCRC ( &Frame[StartPos] , Frame[StartPos] - 2); // compute CRC without
CRC's bytes

    printf ( " CRC=0x%04X\n" , i );

    Frame[19] = i & 0xFF;
    Frame[20] = i >> 8;
    printf ( " CRC IN FRAME(LSB First)= 0x%02X 0x%02X\n" , Frame[19] ,
Frame[20]);


    printf ( " Frame with CRC= " );
    for( i = 0 ; i < sizeof( Frame) ; i++ )
    {
        if( i > 0 ) putchar(',');
        printf ( " 0x%02X" , Frame[i] );
    }
    putchar('\n');


    i = FCRC ( &Frame[StartPos] , Frame[StartPos] );  // check CRC
    if( i == 0 )
       printf ( " CRC OK\n");
    else
       printf ( " CRC ERROR\n" );
}

// output
//
//
// Frame without CRC = 0xFF, 0x02, 0x13, 0x30, 0x01, 0x02, 0x03, 0x04, 0x05,
0x06,0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0x12, 0x00, 0x00, 0x03
// CRC=0x88AF

// CRC IN FRAME(LSB First)= 0xAF 0x88

// Frame with CRC=  0xFF, 0x02, 0x13, 0x30, 0x01, 0x02, 0x03, 0x04, 0x05, 0x06,
0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0x12, 0xAF, 0x88, 0x03       //
CRC OK
```

6 REF definition

6.1 Product option

The REF 8 bytes represents the Cairsens reference (Serial number).
It allows to address individually and directly to a Cairsens, when in a network, several Cairsens are linked to a unique concentrator card.
The reference is included in every query with the Cairsens to allow an individual addressing, as only the concerned Cairsens will answer to the query.

FF FF FF FF FF FF FF FF is a generic address allowing to communicate with any product without knowing its reference.
of course in this situation, it has to be used with only one Cairsens linked to the host, to avoid any BUS corruption as all sensors will respond.

First byte is the product ID :

C = CAIRCLIP
D = CAIRSPM **(new for particulates data and battery management)**
H = CairClip H2S 200ppm **(for CAIRCLOUD)**
M = CairClip H2S 20ppm **(for CAIRCLOUD)**
L = CairClip H2S 2ppm **(for CAIRCLOUD)**

The reference is an 8 bytes series coded as follows: XX YY ZZ AA 00 00 00 00

XX	Product ID C=0x43 D=0x44
YY	Gas identification
ZZ	Measure Range
AA	Interface Type
00 00 00 00	serial number

6.2 Coefficient

For each sensor, you must use a multiplicative coefficient to get the final value :

Sensor	Code	coefficient
CO 20ppm	COV	1
NMVOC 16ppm	CIV	1
H2S 1ppm	CHM	4
H2S 200ppm	CHV	10
H2S 20ppm	CHV or HHV	1
H2S 2ppm	CHV or MHV	1
NH3 25ppm	CAV or LHV	100
O3/NO2 1ppm	CCM	4
O3/NO2 250ppb	CCB	1
NO2 250ppb	CNB	1
SO2 1ppm	CSM	4

6.3 Gaz Identification

Product reference second byte gives the gas identification (NH3 in the example below)

REF	8 bytes	0x43
		0x41
		0x56
		0x32
		0x39
		0x44
		0x30
		0x35

List of gases

ASCII	HEX	DATA
A	**0x41**	Ammonia(NH3)
B	0x42	Benzene
C	0x43	Ozone(O3) and Nitrogen Dioxide(NO2)
D	0x44	Dust
E	0x45	CO2
F	0x46	Formaldehyde(CH2O)
G	0x47	CH4
H	0x48	Hydrogen Sulfide(H2S)
I	0x49	NMVOC
L	0x4C	Chlorine(Cl2)
N	0x4E	Nitrogen Dioxide(NO2)
O	0x4F	CO
P	0x50	Tetrachloroethylene
T	0x54	Toluene
S	0x53	SO2

6.4 Measure range

ASCII	HEX	Range
B	0x42	0-250 ppb
M	0x4D	0-1 ppm
V	0x56	0-20 ppm for H2S 0-2 ppm, 0-20 pm or 0-200 ppm for NH3 0-25 ppm
P	0x50	PACKET data block for CAISPM

6.5 Interface type

HEX	Interface

0x01	USB
0x02	UART

6.6 Example

- Query:

SYNC	1 byte	0xFF
STX	1 byte	0x02
LG	1 byte	0x13
	7 bytes	0x30
		0x01
		0x02
		0x03
		0x04
		0x05
		0x06
REF	8 bytes	0xFF
		0xFF
		0xFF
		0xFF
		0xFF
		0xFF
		0xFF
		0xFF
CMD	1 byte	**0x1C**
CRC	2 bytes	0xD1
		0x61
ETX	1 byte	0x03

- Answer:

SYNC	1 byte	0xFF
STX	1 byte	0x02
LG	1 byte	0x1D
	7 bytes	0x2C
		0x01
		0x02
		0x03
		0x04
		0x05
		0x06
REF	8 bytes	0x43
		0x48
		0x56
		0x02
		0x00
		0x00
		0x10
		0x08
RSP	1 byte	**0x1D**
Product reference	2 bytes	0x43
		0x48
Product option	1 byte	0x56
Product internal ID	5 bytes	0x02
		0x00
		0x00
		0x10
		0x08
END	2 bytes	0x80(LIFE)
		0xFF
CRC	2 bytes	0x06
		0xBA
ETX		0x03

REF: CHV0200001008

7 Reading of the instant value of the Cairsens (GetValue)

7.1 Query

This structure allows the reading of the instant value of the Cairsens (last 1minute data stored)
Query:

- HeaderUart REF CMD CRC ETX
- Command byte CMD = 0x12

7.2 Answer

HeaderUart REF RSP PARAM=0xXX CRC TrailerUart
- Answer byte RSP = 0x13
- Instant value byte PARAM (see below)

The last value (last data stored) is expressed as follows:
- Parameter 1: 1 data

7.3 Example 1 byte by value

- Query:

SYNC	1 byte	0xFF
STX	1 byte	0x02
LG	1 byte	0x13
	7 bytes	0x30
		0x01
		0x02
		0x03
		0x04
		0x05
		0x06
REF	8 bytes	0xFF
		0xFF
		0xFF
		0xFF
		0xFF
		0xFF
		0xFF
		0xFF
CMD	1 byte	**0x12**
CRC	2 bytes	0xAF
		0x88
ETX	1 byte	0x03

- Answer:

SYNC	1 byte	0xFF
STX	1 byte	0x02
LG	1 byte	0x16
	7 bytes	0x2C
		0x01
		0x02
		0x03
		0x04
		0x05
		0x06
REF	8 bytes	0x43
		0x41
		0x56
		0x32
		0x39
		0x44
		0x30
		0x35
RSP	1 byte	**0x13**
Measure	1 byte	0xD1
END	2 byte	0x00(LIFE)
		0xFF
CRC	2 bytes	0x70
		0xFB
ETX	1 byte	0x03

here the value is 0xD1 = 209

it's a cairsens CAV (NH3 25ppm) : measure = 209*100 = 20900pbb = 20.9 ppm

for a cairsens CHM (H2S 1ppm) : measure = 209*4 = 836pbb = 0.836 ppm
for a cairsens CCM (O3 1ppm) : measure = 209*4 = 836pbb = 0.836 ppm
for a cairsens CSM (SO2 1ppm) : measure = 209*4 = 836pbb = 0.836 ppm

7.4 Example 2 bytes by value

- Query:

SYNC	1 byte	0xFF
STX	1 byte	0x02
LG	1 byte	0x13
	7 bytes	0x30
		0x01
		0x02
		0x03
		0x04
		0x05
		0x06
REF	8 bytes	0xFF
		0xFF
		0xFF
		0xFF
		0xFF
		0xFF
		0xFF
		0xFF
CMD	1 byte	**0x12**
CRC	2 bytes	0xAF
		0x88
ETX	1 byte	0x03

- Answer:

SYNC	1 byte	0xFF
STX	1 byte	0x02
LG	1 byte	0x17
	7 bytes	0x2C
		0x01
		0x02
		0x03
		0x04
		0x05
		0x06
REF	8 bytes	0x43
		0x49
		0x56
		0x32
		0x33
		0x33
		0x30
		0x33
RSP	1 byte	**0x13**
	2 bytes	0xB8
		0x2E
END	2 byte	0x00(LIFE)
		0xFF
CRC	2 bytes	0xF3
		0x8D
ETX	1 byte	0x03

it's a cairsens CIV (2 bytes by value)
measure = (0x2E * 256 + 0xB8) = 11960 ppb = 11.960 ppm

for a cairsens H2S 200ppm the value is: 11960 * 10 = 119600ppb = 119.6 ppm

8 GetDownload structure for cairsens (Stored data download)

8.1 Query

Command byte is **CMD** = 0x0C

The parameter allowing the data download **PARAM** is built on a byte which value can vary and refer to several periods to download:

- 0x00: 10 successive points of measurement
- 0x01: 96 successive points of measurement for 1 byte by value , 48 successive points of

measurement for 2 byte by value
- 0x02: send 7 answer of 96 successive points of measurement for 1 byte by value , 48 successive points of measurement for 2 byte by value
- 0x03: send 30 answers of 96 successive points of measurement for 1 byte by value , 48 successive points of measurement for 2 byte by value
- 0x04: send 60 answers of 96 successive points of measurement for 1 byte by value , 48 successive points of measurement for 2 byte by value
- 0x05: send 90 answers of 96 successive points of measurement for 1 byte by value , 48 successive points of measurement for 2 byte by value
- 0x06: send 240 answers of 96 successive points of measurement for 1 byte by value , 48 successive points of measurement for 2 byte by value
- 0x07: send 300 answers of 96 successive points of measurement for 1 byte by value , 48 successive points of measurement for 2 byte by value

This number of points of measurement is valid for a sampling factory configured (meaning one measurement per minute)

8.2 Answer

HeaderUart REF RSP PARAM **TrailerUart**

Answer byte is RSP = 0x0D
- Information + requested data = PARAM (see below)

PARAM holds various information about Cairsens' status in addition to the requested data.

This sequence of information consists of the 10 following parameters:
- Parameter 1 - 1 byte: number of the RS232 exchange frame, coded in hexadecimal (from 0x01 to 0xFF)
- Parameter 2 - 1 byte: total number of RS232 exchange frames, coded in hexadecimal (from 0x01 to 0xFF)
- Parameter 3 - 2 bytes: not used
- Parameter 4 - 1 byte: not used
- Parameter 5 – 1 byte: not used
- Parameter 6 – 1 byte: not used
- Parameter 7 – 1 byte: not used
- Parameter 8 – 1 byte: not used
- Parameter 9 – 2 bytes: not used
- Parameter 10 – 96 bytes:

  1 byte by value : 96 data of 1 byte each = 96 bytes of pollutant level data
  2 bytes by value : 48 data of 2 byte each = 96 bytes of pollutant level data

8.3 Example 10 minutes data 1 byte by value

- Query:

SYNC	1 byte	0xFF
STX	1 byte	0x02
LG	1 byte	0x14
	7 bytes	0x30
		0x01
		0x02
		0x03
		0x04
		0x05
		0x06
REF	8 bytes	0xFF
		0xFF
		0xFF
		0xFF
		0xFF
		0xFF
		0xFF
		0xFF
CMD	1 byte	**0x0C**
PARAM	1 byte	0x00
CRC	2 bytes	0x63
		0xA8
ETX	1 byte	0x03

- Answer:

SYNC	1 byte	0xFF
STX	1 byte	0x02
LG	1 byte	0x2A
	7 bytes	0x2C
		0x01
		0x02
		0x03
		0x04
		0x05
		0x06
REF	8 bytes	0x43
		0x48
		0x4D
		0x02
		0x09
		0x14
		0x00
		0x22
RSP	1 byte	**0x0D**
number of the RS232 exchange frame, coded in hexadecimal	1 byte	0x01
total number of RS232 exchange frames, coded in hexadecimal	1 byte	0x01
not used	2 bytes	0x00
		0x00
not used	1 byte	0x00
not used	1 byte	0x00
not used	1 byte	0x00
not used	1 byte	0x00
not used	1 byte	0x00
not used	2 bytes	0x00
		0x00
value n°1(oldest)	1 byte	0x00
value n°2	1 byte	0x00
value n°3	1 byte	0x00
value n°4	1 byte	0x00
value n°5	1 byte	0x00
value n°6	1 byte	0x00
value n°7	1 byte	0x00
value n°8	1 byte	0x00
value n°9	1 byte	0x00
value n°10(recent)	1 byte	0x00
END	2 bytes	0x00(LIFE)
		0xFF
CRC	2 bytes	0x4D
		0x90

ETX	1 byte	0x03

for a cairsens CAV (NH3 25ppm) : measure in ppb = value*100
for a cairsens CHM (H2S 1ppm) : measure in ppb = value*4
for a cairsens CCM (O3 1ppm) : measure in ppb = value*4
for a cairsens CSM (SO2 1ppm) : measure in ppb = value*4

8.4 Example 10 minutes data 2 bytes by value

Download of 10 minutes data => GetDownload query (0x00):

- Query:

SYNC	1 byte	0xFF
STX	1 byte	0x02
LG	1 byte	0x14
	7 bytes	0x30
		0x01
		0x02
		0x03
		0x04
		0x05
		0x06
REF	8 bytes	0xFF
		0xFF
		0xFF
		0xFF
		0xFF
		0xFF
		0xFF
		0xFF
CMD	1 byte	**0x0C**
PARAM	1 byte	0x00
CRC	2 bytes	0x63
		0xA8
ETX	1 byte	0x03

- Answer:

SYNC	1 byte	0xFF
STX	1 byte	0x02
LG	1 byte	0x34
	7 bytes	0x2C
		0x01
		0x02
		0x03
		0x04
		0x05
		0x06
REF	8 bytes	0x43
		0x49
		0x56
		0x02
		0x33
		0x33
		0x00
		0x33
RSP	1 byte	**0x0D**
number of the RS232 exchange frame, coded in hexadecimal	1 byte	0x01
total number of RS232 exchange frames, coded in hexadecimal	1 byte	0x01
not used	2 bytes	0x00
		0x00
not used	1 byte	0x00
not used	1 byte	0x00
not used	1 byte	0x00
not used	1 byte	0x00
not used	1 byte	0x00
not used	2 bytes	0x00
		0x00
value n°1(oldest)	2 bytes	0xE8
		0x2B
value n°2	2 bytes	0x60
		0x2C
value n°3	2 bytes	0x1A
		0x2C
value n°4	2 bytes	0x8E
		0x2B
value n°5	2 bytes	0x8E
		0x2B
value n°6	2 bytes	0x8E
		0x2B
value n°7	2 bytes	0x06
		0x2C

value n°8	2 bytes	0x60
		0x2C
value n°9	2 bytes	0xDE
		0x2B
value n°10(recent)	2 bytes	0xE8
		0x2B
END	2 bytes	0x00(LIFE)
		0xFF
CRC	2 bytes	0x69
		0x0D
ETX	1 byte	0x03

value 1 : 0x2B *256 + 0xE8 = 11240 ppb
value 2 : 0x2C *256 + 0x60 = 11360 ppb
value 3 : 0x2C *256 + 0x1A = 11290 ppb
value 4 : 0x2B *256 + 0x8E = 11150 ppb
value 5 : 0x2B *256 + 0x8E = 11150 ppb
value 6 : 0x2B *256 + 0x8E = 11150 ppb
value 7 : 0x2C *256 + 0x06 = 11270 ppb
value 8 : 0x2C *256 + 0x60 = 11360 ppb
value 9 : 0x2B *256 + 0xDE = 11230 ppb
value 10: 0x2B *256 + 0xE8 = 11240 ppb

for a cairsens H2S 200ppm : measure 1 = value1*10 = 112400 ppb = 112.4 ppm