123 rue de Bellevue, 92100 Boulogne Billancourt
France
Tél : 33-(0) 1 41 41 00 02 - www.nano-sense.com

EP5000-M air quality probe Modbus protocol

Ver	Date	Update
V1	25/05/2018	Initial version
V2	26/05/2018	Remove ASCII traces
V3	06/06/2018	CRC in detail
V4	09/06/2018	Details
V5	27/06/2018	Status code extended
V6	22/11/2018	PM added + registration for POE
V7	29/01/2019	New sensors data (Pressure, Sound) and sensors presence
V8	24/08/2019	Updates
V9	29/10/2019	Lux and light color T° sensor data added
V10	07/11/2019	Flickering, absolute humidity, FRU and physio added + reorganization
V11	08/02/2020	LEDs dimming & management added

Summary

Modbus Protocol

The Modbus protocol allows a master unit to access up to 255 slave units connected on a single bus. Each slave is assigned an address that distinguishes it from other slaves connected to the bus.

The bus address is settable by NFC. By default, the address is 1.

Transactions can be only initiated by the master and are of two types:
- Question / answer One slave is addressed
- Broadcast / no answer All slaves are addressed, but they shall not reply

Characteristics used for communication with Modbus protocol:

Characteristics	RTU (8 bits)
coding System	Binary
Number of bits par character	10
Start bits	1
data bits (least significant first)	8
parity (optional)	No parity
Stop bits	1
Error Checking	CRC16
Baud rate	9600

RTU communication

RTU mode transmission is in binary. Termination of the frame is determined by a time of silence of about 3.5 bytes (in our case about 30ms)

MASTER

ADDRESS	FUNCTION	DATA	ERROR CHECK CRC 16
8-BITS	8BITS	N X 16-BITS	BITS

SLAVE

ADDRESS	FUNCTION	NUMBER OF DATA BYTES	DATA	ERROR CHECK CRC 16
8-BITS	8BITS	8BITS	N X 16-BITS	16 BITS

Function Field "Function"

The function code tells the recipient slave which function to address.

CODE	MEANING	ACTION
01	READ COIL STATUS	Obtains current status, (ON/OFF), of a group of logic coils.
02	READ INPUT STATUS	Obtains current status, (ON/OFF), of a group of discrete inputs.
03	READ HOLDING REGISTER	Obtains current binary value in one or more holding registers.
04	READ INPUT REGISTER	Obtains current binary value in one or more input registers.
05	FORCE SINGLE COIL	Force logic coil to a state of ON or OFF.
06	PRESET SINGLE REGISTER	Place a specific binary value into a holding register.
15	WRITE MULTIPLE COILS	Force a group of logic coils to a defined state.
16	PRESET MULTIPLE REGISTERS	Place specific binary values into a group of holding registers.

Number of data Field

This field contains a number indicating the number of bytes in the Data fields.

Data Field: "Data Field"

Data field contains information necessary for the slave to process the command sent by the master, or contains information that is sent in response by the slave to the master.

Request from the master:

INPUT MODE: Function = 4

FIRST REGISTER	NUMBER OF REGISTERS TO READ
16-BIT	16-BIT

Request from master:

WRITE MULTIPLE REGISTERS: Function = 16

REGISTER'S ADDRESS	NUMBER OF REGISTERS TO WRITE	BYTE COUNT	VALUE(S) TO WRITE
16-BIT	16-BIT	8-BIT	16-BIT

The address of the first register is **0**

Read access: Function = 4 (0x04)

REGISTER # 1: PRODUCT CODE	REGISTER # 2: FIRMWARE VERSION	REGISTER #3: SENSORS PRESENCE	REGISTER #4: BIT STATUS	REGISTER # 5: BUILT IN TEST EQUIPMENT / FRU
16-BITS	4-CHAR = 16-BITS	16-BITS	16-BITS	16-BITS

REGISTER # 6: CO2 CONCENTRATION	REGISTER # 7:VOC CONCENTRATION	REGISTER # 8: TEMPERATURE	REGISTER # 9: RELATIVE HUMIDITY	REGISTER # 10: ABSOLUTE HUMIDITY
16-BITS	16-BITS	16-BITS	16-BITS	16-BITS

REGISTER #11: ATMOS. PRESSURE	REGISTER # 12: PM10	REGISTER # 12: PM2.5	REGISTER # 14: PM1	REGISTER #15: AVERAGE NOISE LEVEL
	16-BITS	16-BITS	16-BITS	16-BITS

REGISTER # 16: PEAK NOISE LEVEL	REGISTER # 17: LUX	REGISTER # 18: LIGHT COLOR T°	REGISTER # 19: LIGHT FLICKERING	REGISTER # 20: ON OFF VENTILATION COMMAND
16-BITS	16-BITS	16-BITS	16-BITS	16-BITS

REGISTER # 21: LINEAR VENTILATION COMMAND	REGISTER #: 22 HEATER COMMAND IN %	REGISTER #: 23 COOLING COMMAND IN %	REGISTER #24: COGNITIVITY INDEX	REGISTER # 25: QUALITY OF SLEEP INDEX
16-BITS	16-BITS	16-BITS	16-BITS	16-BITS

REGISTER #26: HEALTH INDEX	REGISTER # 27: RESERVED	REGISTER # 28: RESERVED	REGISTER #:29 REMEDIATION ON VALUES OR PHYSILOGICAL EFFECTS	REGISTER #30: CO2 SET POINT
16-BITS	16-BITS			16-BITS

REGISTER # 31: VOC SET POINT	REGISTER # 32: RH SET POINT	REGISTER # 33: PM2.5 SET POINT	REGISTER #34: PRODUCTIVITY SETPOINT	REGISTER # 35: QUALITY OF SLEEP SETPOINT
16-BITS	16-BITS	16-BITS	16-BITS	16-BITS

REGISTER # 36: HEALTH SETPOINT	REGISTER # 37: HEATING SETPOINT	REGISTER # 38: COOLING SETPOINT	REGISTER # 39: LED DIMMING	REGISTER #40: IP RESERVED or NETWORK REGISTRATION
16-BITS	16-BITS	16-BITS	16-BITS	16-BITS

Description of read records:

PRODUCT CODE: Register #1 (address 0)
2-CHAR (16-BITS):
01 E5000
02 P5000
03 EP5000
04 AAQ

FIRMWARE VERSION: Register #2 (address 1)
16-BITS

SENSOR PRESENCE: REGISTER #3 (address 2)

B15	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0

BIT 0: 0 ➔ CO2 sensor not present
1 ➔ CO2 sensor present

BIT 1: 0 ➔ VOCT sensor not present
1 ➔ VOCT sensor present

BIT 2: 0 ➔ Temperature sensor not present
1 ➔ Temperature sensor present

BIT 3: 0 ➔ Humidity sensor not present
1 ➔ Humidity sensor present

BIT 4: 0 ➔ Particles PM1 sensor not present
1 ➔ Particles PM1 sensor present

BIT 5: 0 ➔ Particles PM2.5 sensor not present
1 ➔ Particles PM2.5 sensor present

BIT 6: 0 ➔ Particles PM10 sensor not present
1 ➔ Particles PM10 sensor present

BIT 7: 0 ➔ Pressure sensor not present
1 ➔ Pressure sensor present

BIT 8: 0 ➔ Sound sensor not present
1 ➔ Sound sensor present

BIT 9: 0 ➔ Lux sensor not present
1 ➔ Lux sensor present

BIT 10: 0 ➔ Light Color T° sensor not present
1 ➔ Light color T° sensor present

BIT 11: 0 ➔ Flickering sensor not present
1 ➔ Flickering sensor present

BIT 12: Reserved
BIT 13: Reserved
BIT 14: Reserved
BIT 15: Reserved

BIT STATUS: Register #4 (address 3)

Each failure is allocated to a specific bit so combination of failures can be indicated

	B15	**B14**	**B13**	**B12**	**B11**	**B10**	**B9**	**B8**	**B7**	**B6**	**B5**	**B4**	**B3**	**B2**	**B1**	**B0**
CO2 sensor failure																X
VOC sensor failure															X	
T° & RH Sensor Failure														X		
Particles sensor Failure													X			
Pressure sensor Failure												X				
Sound sensor Failure											X					
Light sensor										X						

failure															
NFC EEPROM default									X						
Power supply too low								X							
Power supply too high							X								
T° too high						X									
T° too low					X										
Sensors life span overpassed				X											
Modbus integrity failure			X												
Reserved		X													
Reserved	X														

BUILT IN TEST FRU: REGISTER #5 (address 4)

B15	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0

BIT 0: 0 ➔ Front PCB FRU to be replaced
1 ➔ Front PCB FRU OK

BIT 1: 0 ➔ Main board PCB to be replaced
1 ➔ Main board PCB OK

BIT 2: 0 ➔ CO2 sensor module to be replaced
1 ➔ CO2 sensor module OK

BIT 3: 0 ➔ VOC sensor module to be replaced
1 ➔ VOC sensor module OK

BIT 4: 0 ➔ Inter board PCB to be replaced
1 ➔ Inter board PCB OK

BIT 5: 0 ➔ Particles sensor to be replaced
1 ➔ Particles sensor OK

BIT 6: 0 ➔ Power supply PCB to be replaced
1 ➔ Power supply PCB OK

BIT 7: Reserved
BIT 8: Reserved
BIT 9: Reserved
BIT 10: Reserved
BIT 11: Reserved
BIT 12: Reserved
BIT 13: Reserved
BIT 14: Reserved

BIT 15: Reserved

CO2 CONCENTRATION (in ppm): Register #6 (address 5)

16-BITS:
Bit 0 to 14: Value
Bit 15 = 1: Saturation

VOC CONCENTRATION (in µg/m3): Register #7 (address 6)

16-BITS:
Bit 0 to 14: Value
Bit 15 = 1: Saturation

TEMPERATURE (in 0.1 °C): Register # 8 (address 7)

16-BITS
16 bits = temperature value (signed)

Examples
0°C = 0
12,9°C (value sent: 129) = 129 (decimal)
-5°C (value sent: -50) = -32718 (decimal) (complement at 2 on 16 bits: 1 bit for sign + 15 bits for value)

RELATIVE HUMIDITY in %: Register # 9 (address 8)

2-CHAR (16-BITS):
16 bits = humidity value (not signed)

ABSOLUTE HUMIDITY in %: Register # 10 (address 9)

2-CHAR (16-BITS):
16 bits = humidity value (not signed)

PRESSURE in 0.1 Pa: Register # 11 (address 10)

2-CHAR (16-BITS):
16 bits = pressure value (not signed)

PM10 in µg/m3: Register # 12 (address 11)

2-CHAR (16-BITS):
16 bits = PM10 value (not signed)

PM2,5 in µg/m3: Register # 13 (address 12)

2-CHAR (16-BITS):
16 bits = PM2.5 value (not signed)

PM1 in µg/m3: Register # 14 (address 13)

2-CHAR (16-BITS):
16 bits = PM1 value (not signed)

AVERAGE SOUND LEVEL: Register # 15 (address 14)

2-CHAR (16-BITS):
16 bits = Sound average value (not signed)

PEAK SOUND LEVEL: Register # 16 (address 15)

2-CHAR (16-BITS):
16 bits = Sound peak value (not signed)

LUX: Register # 17 (address 16)

2-CHAR (16-BITS):
16 bits = Lux value (not signed)

LIGHT COLOR T°: Register # 18 (address 17)

2-CHAR (16-BITS):
16 bits = Light color T° value (not signed)

LIGHT FLICKERING in %: Register # 19 (address 18)

2-CHAR (16-BITS):
16 bits = Light flickering value (not signed)

2 SPEEDS VENTILATION ON OFF COMMAND: Register # 20 (address19)

2-CHAR (16-BITS):
Character 1: Fan Speed 1
Character 2: Fan Speed 2

00: Ventilation Off
FF: Ventilation ON

Example:

- Fan 1 ON
- Fan 2 OFF

REGISTER #7 : ON OFF 2 FAN SPEEDS COMMAND			
Character 1		Character 2	
F	F	0	0
4-CHAR = 16-BITS			

LINEAR VENTILATION COMMAND: REGISTER # 21 (ADDRESS 20)

2-CHAR (16-bit)
Value between 0 and 100%. This value can be used to control ventilation in continues. To be noted that to keep the building health a minimum of 10% is applied.
For Automatic Baseline Calibration of CO2 and VOC sensors, the ventilation will be activated at 100% during 30 minutes once every 15 days.

HEATING COMMAND LINEAR: REGISTER # 22 (ADDRESS 21)

2-CHAR (16-bit)
Value between 0 and 100%. This value can be used to control heating in continues.

COOLING COMMAND LINEAR: REGISTER # 23 (ADDRESS 22)

2-CHAR (16-bit)
Value between 0 and 100%. This value can be used to control air conditioning in continues.

COGNITIVITY INDEX: Register # 24 (address 23)

2-CHAR (16-bit)
Cognitivity / productivity
Value between 0 and 100%

QUALITY OF SLEEP INDEX: Register # 25 (address 24)

2-CHAR (16-bit)
Quality of sleep
Value between 0 and 100%

HEALTH INDEX: Register # 26 (address 25)

2-CHAR (16-bit)

RESERVED: Register # 27 (address 26)

RESERVED: Register # 28 (address 27)

REMEDIATION ON THRESHOLD OR PHYSIOLOGICAL EFFECTS: Register # 29 (address 28)

2-CHAR (16-BITS):

B15	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0

BIT 0: 0 ➔ Thresholds on concentration setpoints with OR
1 ➔ Physiological effects

BIT 1: 0 ➔ Health not taken into account in Physiological effects
1 ➔ Health not taken into account in Physiological effects

BIT 2: 0 ➔ Productivity / cognitivity taken into account in Physiological effects
1 ➔ Quality of sleep taken into account in Physiological effects

CO2 SET POINT: REGISTER # 30 (ADDRESS 29)
2-CHAR (16-bit)

VOC SET POINT: REGISTER # 31 (ADDRESS 30)
2-CHAR (16-bit)

RH SET POINT: REGISTER # 32 (ADDRESS 31)
2-CHAR (16-bit)

PM2.5 SET POINT: REGISTER # 33 (ADDRESS 32)
2-CHAR (16-bit)

PRODUCTIVITY SET POINT: REGISTER # 34 (ADDRESS 33)
2-CHAR (16-bit)
Value between 0 and 100%

QUALITY OF SLEEP SET POINT: REGISTER # 35 (ADDRESS 34)
2-CHAR (16-bit)
Value between 0 and 100%

HEALTH SET POINT: REGISTER # 36 (ADDRESS 35)
2-CHAR (16-bit)
Value between 0 and 100%

HEATING SETPOINT (in 0.1 °C): Register # 37 (address 36)
2-CHAR (16-BITS)
16 bits = temperature value (unsigned)
Examples
20.9°C (value sent: 209) = 209 (decimal)

COOLING SETPOINT (in 0.1 °C): Register # 38 (address 37)
2-CHAR (16-BITS)
16 bits = temperature value (unsigned)
Examples
28.9°C (value sent: 289) = 289 (decimal)

LEDs DIMMING: REGISTER # 39 (address 38)
2-CHAR (16-bit)
Character 1: LEDs Dimming in %
Character 2: LEDs Rules

LEDS dimming
Value between 0 and 100%

BIT 0: 0 ➔ Dimming on physiological effects
1 ➔ Dimming on thresholds

BIT 1: 0 ➔ Off at night
1 ➔ On at night

BIT 2: 0 ➔ 100% of dimming set point at night (if Bit 1 at 1)
1 ➔ 10% of dimming set point at night (if Bit 1 at 1)

IP NETWORK REGISTRATION: Register # 40 (address 39) (Specific to POE version with external POE interface)

16-BITS
Byte 1: = 00 and Byte 2: FF: Registration requested
Byte 1: = FF and Byte 2: 00: Registration acknowledge, no request

REGISTER #8 : IP NETWORK REGISTRATION			
Byte 1		Byte 2	
F	F	0	0
16-BITS			

Write access: Function = 16 (0x10)

Only registers 29 to 40 are writable to allow adjustment of ventilation, heating and cooling set points and registration request.

Description of writable registers

REMEDIATION ON THRESHOLD OR PHYSIOLOGICAL EFFECTS: Register # 29 (address 28)
2-CHAR (16-BITS):

B15	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0

BIT 0: 0 ➔ Thresholds on concentration setpoints with OR
1 ➔ Physiological effects

BIT 1: 0 ➔ Health not taken into account in Physiological effects
1 ➔ Health not taken into account in Physiological effects

BIT 2: 0 ➔ Productivity / cognitivity taken into account in Physiological effects
1 ➔ Quality of sleep taken into account in Physiological effects

CO2 SET POINT: REGISTER # 30 (ADDRESS 29)
2-CHAR (16-bit)
16 bits = CO2 value (unsigned)

Examples
1000ppm (value sent: 1000) = 1000 (decimal)

VOC SET POINT: REGISTER # 31 (ADDRESS 28)
2-CHAR (16-bit)
16 bits = VOC value (unsigned)

Examples
1000µg/m^3 (value sent: 1000) = 1000 (decimal)

RH SET POINT: REGISTER # 32 (ADDRESS 31)
2-CHAR (16-bit)
16 bits = T° value (unsigned)

Examples
50.5°C (value sent: 505) = 505 (decimal)

PM2.5 SET POINT: REGISTER # 33 (ADDRESS 32)
16 bits = PM2.5 value (unsigned)

Examples
11µg/m^3 (value sent: 11) = 11 (decimal)

PRODUCTIVITY SET POINT: REGISTER # 34 (ADDRESS 33)
16 bits = Productivity value (unsigned)

Examples
88.5% (value sent: 805) = 805 (decimal)

QUALITY OF SLEEP SET POINT: REGISTER # 35 (ADDRESS 34)
16 bits = Productivity value (unsigned)

Examples
88.5% (value sent: 805) = 805 (decimal)

HEALTH SET POINT: REGISTER # 36 (ADDRESS 35)
16 bits = Productivity value (unsigned)

Examples
88.5% (value sent: 805) = 805 (decimal)

HEATING SETPOINT (in 0.1 °C): Register # 37 (address 36)
2-CHAR (16-BITS)
16 bits = temperature value (unsigned)

Examples
20.9°C (value sent: 209) = 209 (decimal)

COOLING SETPOINT (in 0.1 °C): Register # 38 (address 37)
2-CHAR (16-BITS)
16 bits = temperature value (unsigned)

Examples
28.9°C (value sent: 289) = 289 (decimal)

IP NETWORK REGISTRATION: Register # 40 (address 39)
16-BITS
Byte 1: = FF and Byte 2: 00: Registration acknowledge

REGISTER #8 : IP NETWORK REGISTRATION (POE)			
Byte 1		Byte 2	
F	F	0	0
16-BITS			

Response to master when using function 16 (0x10)

Write acknowledge:

FUNCTION CODE (0x10)	**REGISTER'S ADDRESS**	**NUMBER OF REGISTERS TO WRITE**

8-BIT	16-BIT	16-BIT

When ending with an error:

ERROR CODE (0x90)	EXCEPTION CODE
8-BIT	8-BIT

The exception code used by the EP5000 is the exception n° 3 (Illegal data value). This exception means registration is impossible.

CRC16 calculation

```
static const unsigned char auchCRCHi[] = {
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81,
0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0,
0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01,
0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41,
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81,
0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0,
0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01,
0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40,
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81,
0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0,
0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01,
0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,
0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81,
0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0,
0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01,
0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81,
0x40
} ;

static const unsigned char auchCRCLo[] = {
0x00, 0xC0, 0xC1, 0x01, 0xC3, 0x03, 0x02, 0xC2, 0xC6, 0x06, 0x07, 0xC7, 0x05, 0xC5, 0xC4,
0x04, 0xCC, 0x0C, 0x0D, 0xCD, 0x0F, 0xCF, 0xCE, 0x0E, 0x0A, 0xCA, 0xCB, 0x0B, 0xC9, 0x09,
0x08, 0xC8, 0xD8, 0x18, 0x19, 0xD9, 0x1B, 0xDB, 0xDA, 0x1A, 0x1E, 0xDE, 0xDF, 0x1F, 0xDD,
0x1D, 0x1C, 0xDC, 0x14, 0xD4, 0xD5, 0x15, 0xD7, 0x17, 0x16, 0xD6, 0xD2, 0x12, 0x13, 0xD3,
0x11, 0xD1, 0xD0, 0x10, 0xF0, 0x30, 0x31, 0xF1, 0x33, 0xF3, 0xF2, 0x32, 0x36, 0xF6, 0xF7,
0x37, 0xF5, 0x35, 0x34, 0xF4, 0x3C, 0xFC, 0xFD, 0x3D, 0xFF, 0x3F, 0x3E, 0xFE, 0xFA, 0x3A,
0x3B, 0xFB, 0x39, 0xF9, 0xF8, 0x38, 0x28, 0xE8, 0xE9, 0x29, 0xEB, 0x2B, 0x2A, 0xEA, 0xEE,
0x2E, 0x2F, 0xEF, 0x2D, 0xED, 0xEC, 0x2C, 0xE4, 0x24, 0x25, 0xE5, 0x27, 0xE7, 0xE6, 0x26,
0x22, 0xE2, 0xE3, 0x23, 0xE1, 0x21, 0x20, 0xE0, 0xA0, 0x60, 0x61, 0xA1, 0x63, 0xA3, 0xA2,
0x62, 0x66, 0xA6, 0xA7, 0x67, 0xA5, 0x65, 0x64, 0xA4, 0x6C, 0xAC, 0xAD, 0x6D, 0xAF, 0x6F,
0x6E, 0xAE, 0xAA, 0x6A, 0x6B, 0xAB, 0x69, 0xA9, 0xA8, 0x68, 0x78, 0xB8, 0xB9, 0x79, 0xBB,
0x7B, 0x7A, 0xBA, 0xBE, 0x7E, 0x7F, 0xBF, 0x7D, 0xBD, 0xBC, 0x7C, 0xB4, 0x74, 0x75, 0xB5,
0x77, 0xB7, 0xB6, 0x76, 0x72, 0xB2, 0xB3, 0x73, 0xB1, 0x71, 0x70, 0xB0, 0x50, 0x90, 0x91,
0x51, 0x93, 0x53, 0x52, 0x92, 0x96, 0x56, 0x57, 0x97, 0x55, 0x95, 0x94, 0x54, 0x9C, 0x5C,
0x5D, 0x9D, 0x5F, 0x9F, 0x9E, 0x5E, 0x5A, 0x9A, 0x9B, 0x5B, 0x99, 0x59, 0x58, 0x98, 0x88,
0x48, 0x49, 0x89, 0x4B, 0x8B, 0x8A, 0x4A, 0x4E, 0x8E, 0x8F, 0x4F, 0x8D, 0x4D, 0x4C, 0x8C,
0x44, 0x84, 0x85, 0x45, 0x87, 0x47, 0x46, 0x86, 0x82, 0x42, 0x43, 0x83, 0x41, 0x81, 0x80,
0x40
} ;


unsigned short crc16(unsigned char *puchMsg , unsigned short usDataLen)
{
```



```
  unsigned char uchCRCHi = 0xFF ; /* high byte of CRC initialized */
  unsigned char uchCRCLo = 0xFF ; /* low byte of CRC initialized */
  unsigned uIndex ; /* will index into CRC lookup table */

  unsigned short usVal1;
  unsigned short usVal2;


  while (usDataLen--)/* pass through message buffer */
  {


   uIndex = uchCRCHi ^ *puchMsg; //++ ; /* calculate the CRC */
   puchMsg++ ;
   uchCRCHi = uchCRCLo ^ auchCRCHi[uIndex];
   uchCRCLo = auchCRCLo[uIndex] ;
  }

 usVal1 = uchCRCHi;
 usVal2 = uchCRCLo;

 usVal1 = usVal1 << 8;
 usVal1 = usVal1 | usVal2;

 return usVal1; //( ((unsigned short)uchCRCHi << 8) | (unsigned short)uchCRCLo) ;
}
```

