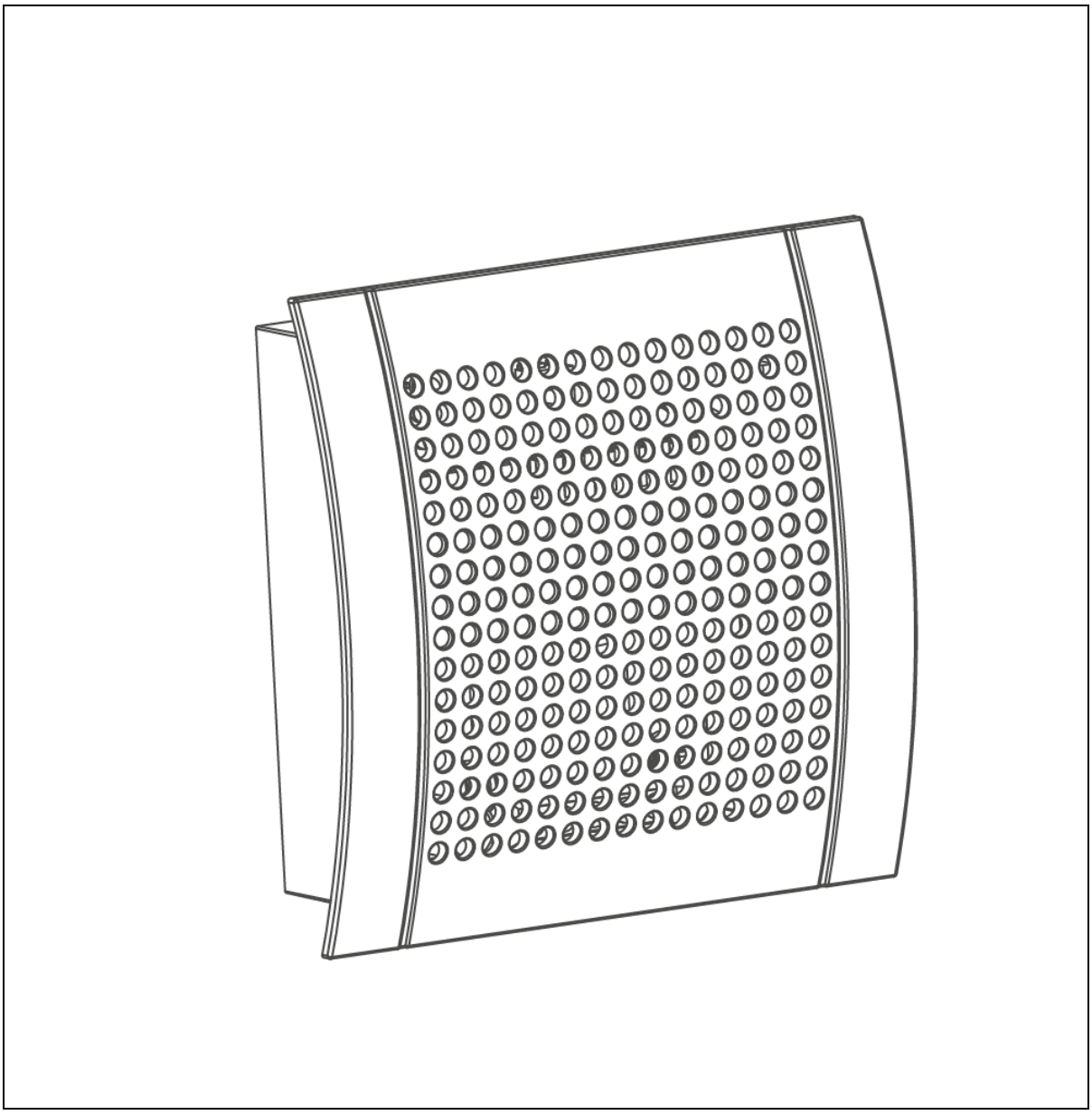
S-C02/T

MODBUS REGISTER TABLE



FLY703_NOT_SENSOR_CO2_MODBUS

1.Modbus Data description

	Virtual address		offset	Register name	Type	Access	Value	Location
Device information	**0x03E8**	1000	0	Section Type	16-bits unsigned	R/O	1	ROM
	0x03E9	1001	1	Section Length	16-bits unsigned	R/O	8	ROM
	0x03EA	1002	2	Device Class High	2 characters	R/O	"CO"	ROM
	0x03EB	1003	3	Device Class Low	2 characters	R/O	"2 "	ROM
	0x03EC	1004	4	Model	16-bits unsigned	R/O	1	ROM
	0x03ED	1005	5	Serial Number High	16-bits unsigned	R/O		ROM
	0x03EE	1006	6	Serial Number Low	16-bits unsigned	R/O		ROM
	0x03EF	1007	7	Version HW	16-bits unsigned	R/O	**257**	ROM
	0x03F0	1008	8	Version FW	16-bits unsigned	R/O	**259**	ROM
	0x03F1	1009	9	Status & Control	16-bits Integer	R/W		RAM
CO2 sensor	**0x03F2**	1010	10	Section Type	16-bits unsigned	R/O	1001	ROM
	0x03F3	1011	11	Section Length	16-bits unsigned	R/O	11	ROM
	0x03F4	1012	12	Measuring Unit	2 characters	R/O	"pm"	ROM
	0x03F5	1013	13	Scale Factor	16-bits unsigned	R/O	1	ROM
	0x03F6	1014	14	Minimum ppm	16-bits unsigned	R/O	0	ROM
	0x03F7	1015	15	Maximum ppm	16-bits unsigned	R/O	2000	ROM
	0x03F8	1016	16	Offset Correction	16-bits unsigned	R/W	0	EEPROM
	0x03F9	1017	17	Gain Correction	16-bits unsigned	R/W	32768	EEPROM
	0x03FA	1018	18	Date of Calibration	16-bits unsigned	R/W	0xffff	EEPROM
	0x03FB	1019	19	Sampling Period	16-bits unsigned	R/W	6000	EEPROM
	0x03FC	1020	20	LowPass Filter	16-bits unsigned	R/W	7	EEPROM
	0x03FD	1021	21	Reported Value	16-bits unsigned	R/O		RAM
	0x03FE	1022	22	Filtered Value	16-bits unsigned	R/O		RAM
Temperature sensor	**0x03FF**	1023	23	Section Type	16-bits unsigned	R/O	1000	ROM
	0x0400	1024	24	Section Length	16-bits unsigned	R/O	5	ROM
	0x0401	1025	25	Measuring Unit	2 characters	R/O	"°C"	ROM

	0x0402	1026	26	Scale Factor	16-bits unsigned	R/O	10	ROM
	0x0403	1027	27	Minimum Temperature	16-bits unsigned	R/O	20	ROM
	0x0404	1028	28	Maximum Temperature	16-bits unsigned	R/O	500	ROM
	0x0405	1029	29	Reported Value	16-bits unsigned	R/O		RAM
REGISTERS_END	**0x0406**	1030	30	End Section	16-bits unsigned	R/O	0	ROM

R/O = Read only

R/W = Read and Write

Status & control bit description														
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
					EEPROM_ERROR	**CO2_SENSOR_ERROR**	**SENSOR_POWER**				**DO (cleared when done)**			**Operation**	**BUSY (read only)**

1.1 Status & Control description

Field	Name	Meanings
15…11	Reserved	
10	EepromErr	All registers placed in EEPROM have three copies for safety If these copies are different then EepromErr is set
9	CO2Err	I2C communication error
8	CO2Pow	When set, the E+E CO2 module is powered
7,…,5	Reserved	
4	Do	When set; it starts the operation specified in OP code
3,2,1	Op	The code of operation to be performed when DO bit is set
0	Busy	When set, it indicates that it cannot serve requests

There are five operations that run with command transmitted by Do bit:

Op = 0; Sync - read raw and average value and starts a new measurement
Op = 1; Write in custom area the value from the Offset Correction register (calibration)
Op = 2 ; Write in custom register the Gain Correction register (calibration)
Op = 3; Poweroff the EE892
Op = 4; Poweron the EE892

Sync operation is set by the master to synchronize the measurement cycles of the CO2 devices.

When the Sampling period is set to 0, the device will sample the CO2 immediately after reading the CO2 Filtered Value and if no such reading happens for one or many hours, the device will sample by itself once each hour.
On Poweron command (Op = 4) the following sequence happens:
- switch on the power to the CO2 module
- wait 5 seconds
- set the measurement time interval, in custom area, at 3600 seconds if it is not already set at this value.
- set the Offset, Gain, Lower Calibration Point and High Calibration Point registers in custom area, equal to the corresponding values from device registers residing in EEPROM, if they are not already equal.

Examples of operation

Some examples of operation follow here below, for a CO2 device having the address 3.
Example for **Synch** operation:
Master frame = {0x03, 0x06, 0x03, 0xf1, 0x00, 0x10, CRClow, CRCHigh}
This means: slaveaddr 3, function 6(write single register), register address 0x3f1, Status&Control Do=1, Op=0
Example for **OffsetCalibration** operation:
Master frame = {0x03, 0x06, 0x03, 0xf8, 0x00, 0x0f, CRClow, CRCHigh}
This means: slaveaddr 3, function 6(write single register), register address 0x3f1, Offset = 15
Master frame = {0x03, 0x06, 0x03, 0xf1, 0x00, 0x12, CRClow, CRCHigh}
This means: slaveaddr 3, function 6(write single register), register address 0x3f1, Status&Control Do=1, Op=1
Example for **GainCalibration** operation:
Master frame = {0x03, 0x06, 0x03, 0xf9, 0x80, 0x00, CRClow, CRCHigh}
This means: slaveaddr 3, function 6(write single register), register address 0x3f1, Gain = 32768
Master frame = {0x03, 0x06, 0x03, 0xf1, 0x00, 0x14, CRClow, CRCHigh}
Thus means: slaveaddr 3, function 6(write single register), register address 0x3f1, Status&Control Do=1, Op=2
Example for **PowerOff** operation:
Master frame = {0x03, 0x06, 0x03, 0xf1, 0x00, 0x16, CRClow, CRCHigh}
This means: slaveaddr 3, function 6(write single register), register address 0x3f1, Status&Control Do=1, Op=3
Example **Poweron** operation:
Master frame = {0x03, 0x06, 0x03, 0xf1, 0x00, 0x18, CRClow, CRCHigh}
This means: slaveaddr 3, function 6(write single register), register address 0x3f1, Status&Control Do=1, Op=4

2. S-C02/T Modbus register table description

- ⇨ All the Modbus devices will map their specific registers virtual addresses contiguously, beginning with the **base address 1000.**
- ⇨ All the r/w registers will be saved in EEPROM and restored at power on time.
- ⇨ The virtual register addresses are organized in **contiguous linked sections** as follows.

2.1 **device informations**

The registers map will begin at the base address with the **device information** section as follows**:**

- ***sectionTypeId* = DEVICE_INFO** - r/o – the section type identifier describing the general information about the device.
- ***sectionLength* = 8** - r/o - the number of additional registers in this section
- ***deviceClassHigh*** = "cc" - r/o – the first two characters of the class name (e.g. for "HU" we have 'H' * 256 + 'U' = 0x4855)
- ***deviceClassLow*** = "cc"- r/o – the last two characters of the class name (e.g. for "B5" we have 'B' * 256 + '5' = 0x4235)
- **deviceModel** = nnnnn - r/o – the model Identification number (e.g. 00007)
- ***serialNumberHigh*** = "nnnnn" - r/o – the higher word of a 32 bits serial number
- ***serialNumberLow*** = "nnnnn" - r/o – the lower word of a 32 bits serial number
- **hwVersion**= 256*hwV + hwSV - r/o – the hardware version and subversion
- **fwVersion** = 256*fwV + fwSV - r/o – the firmware version and subversion
- **deviceStatusAndControl** - r/w & r/o– the bit 0 is r/o and shows the BUSY state of the slave, that is when the device it cannot receive new commands. **Bit 15 is reserved, must not be set**. Other bits are device dependent as per the datasheet of the device model. Any bit declared as r/o will not be affected by writing.

2.2 C02 sensor data description

A device containing one or many **calibrated physical quantity sensors** (e.g. pressure, CO2 sensors) may include the next registers section, one instance for each sensor fitting this interface. This section type is dedicated to the sensors measuring in a rage from a specified minimum value to a specified maximum value and which may be calibrated by offset correction and gain correction.

- ***sectionTypeId* = CALIBRATED_QTY_SENSOR** - r/o – the section type identifier for calibrated physical quantity sensor function
- ***sectionLength* = 11** - r/o - the number of additional registers in this section
- ***measuringUnit*** - r/o - holds two characters defining the measuring unit
- ***scaleFactor*** - r/o - holds a positive number indicating the scale factor
- ***minQty***- r/o - the minimum measurable value multiplied by the scale factor
- ***maxQty***- r/o - the maximum measurable value multiplied by the scale factor
- ***offsetCorrection*** - r/w - holds an integer indicating correction by addition applied by the device to the measurement
- ***gainCorrection*** - r/w - holds an integer number indicating a correction by multiplication applied by the device to the measurement
- ***dateOfCalibration*** - r/w - holds the day when the last calibration was done in the next bit format:

b15 - b9: the year of the 3rd millennium (2016 -> 16)

b8 - b5: month of the year

b0 - b4: day of the month

- ***samplingPeriod*** – r/w – the period from the last measurement, when the device automatically starts a new measurement in 10ms units. Maximum of samplingPeriod is 6min 55s.
- ***lowPassFilter*** - r/w - the selection of the low pass filter cut off frequency. This is an unsigned number and must be between 0 and 7. If the master writes a bigger number, then the device will overwrite back immediately 7. The cutoff frequency is:
- $$\frac{1}{2^{lowPassFilter+1} * samplingPeriod}$$
- ***Reported value*** - r/o - the instant quantity reported value comprised between -32768 and +32767. To get profit of integer numbers, the measured quantity is scaled up by a scale factor before being reported as qty. Therefore, the right measured quantity, expressed in measuring units, will be obtained by dividing this reported qty by the scaleFactor. It is recommended to set the scaleFactor as a power of 2.

The next formula defines the device must apply the scaling and corrections before setting the reported qty

$$\text{qty} = (\text{measure} * \text{scaleFactor} + \text{offsetCorrection}) * \left(1 + \frac{\text{gainCorrection}}{2^{17}}\right)$$

The offset correction may extend up to +/- 32767/scaleFactor measuringUnit. The gain correction may extend up to +/-25%.

The reading of the qty triggers, an immediate measurement and restart the sampling period timer. Using this register reading, the master may synchronize the sampling.

- ***Filtered value*** - r/o - the low pass filtered form of qty signal

2.3 Temperature sensor data description

A device containing one or many **simple physical quantity sensors** (e.g. a thermometer) may include the next registers section, one instance for each sensor:

- ***sectionTypeId* = SIMPLE_QTY_SENSOR** - r/o – the section type identifier for simple physical quantity sensor function.
- ***sectionLength* = 5** - r/o - the number of additional registers in this section
- ***MeasuringUnit***- r/o - two characters defining the measuring unit
- ***scaleFactor*** - r/o - a positive number indicating the scale factor
- ***minimum temperature*** - r/o – a signed number showing the minimum measurable value multiplied by the scale factor
- ***maximum temperature*** - r/o - a signed number showing the maximum measurable value multiplied by the scale factor
- ***reported value*** - r/o - the reported value comprised between -32768 and +32767; the measured value, expressed in quantity units, will be obtained by dividing this qty by the scaleFactor. It is recommended to set the scaleFactor as a power of 2.

General

The Modbus RTU is supported by all units ad controllers, the transport layer is **Modbus RTU, 9600 baud, 8 bits, no parity bit, one stop bit.**

Units are slaves (servers); each unit has a slave address (1 to 15 set by rotary switch) and responds to the requests from a master(client).

Supported Modbus functions are:

a. **0x03 Read Holding Registers**
b. **0x04 Read Input Registers**
c. **0x06 Write Single Register**
d. **0x08 Diagnostics. Subfunctions 0-4, 10-18 & 20.**
e. **0x10 Write Multiple registers.**