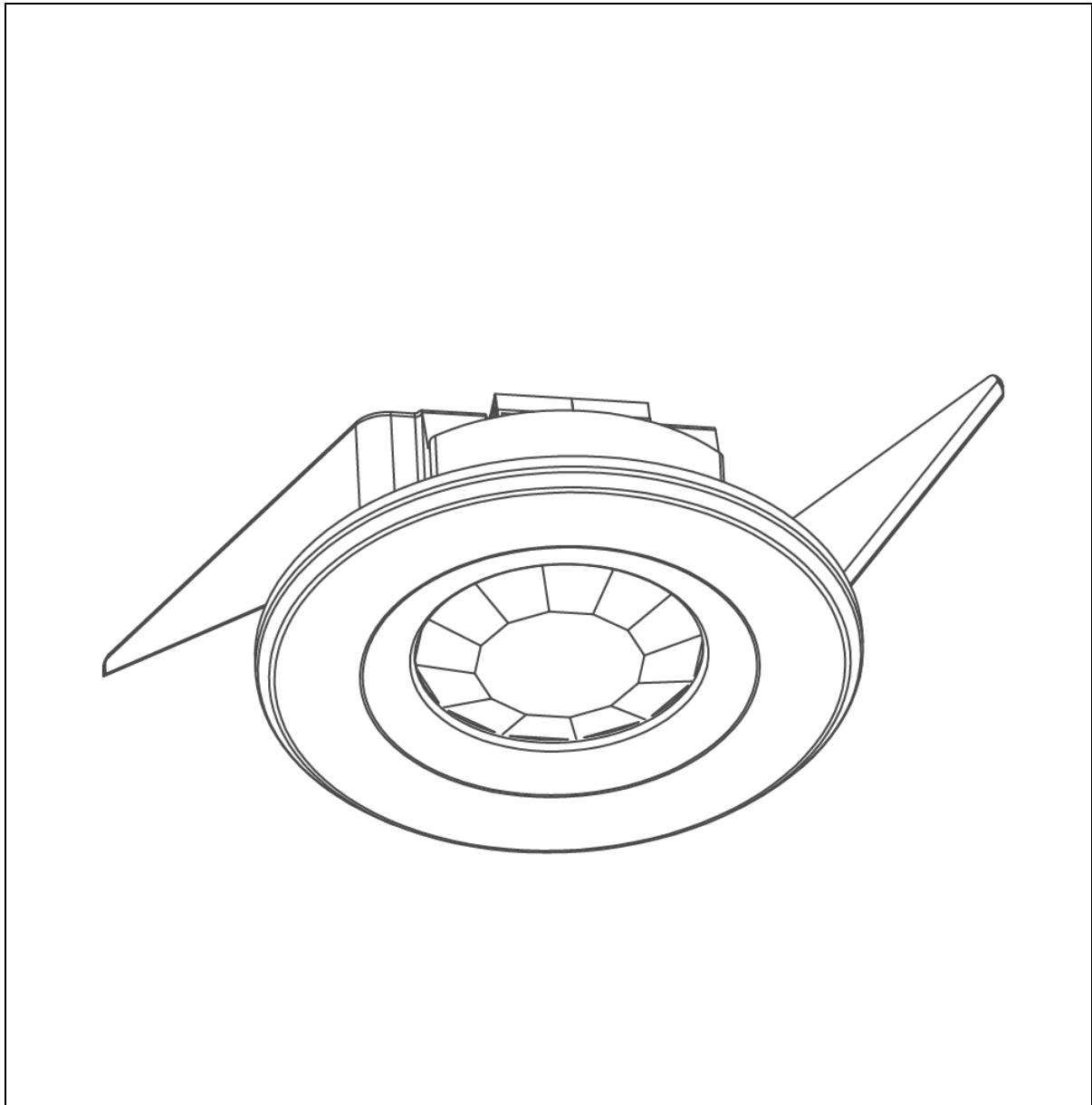


# S-PRE/DAY-NIGHT/T

## MODBUS REGISTER TABLE



FLY704\_NOT\_SENSOR\_OCCUPANCY\_MODBUS



## 1.Modbus Data description

	Address	Offset	Register name	Type	Access	Value	Location
<b>Device information</b>	1000	0	Section Type	16bits unsigned	R/O	1	ROM
	1001	1	Section Length	16bits unsigned	R/O	8	ROM
	1002	2	Device Class High	2 characters	R/O	"PY"	ROM
	1003	3	Device Class Low	2 characters	R/O	"RO"	ROM
	1004	4	Model	16bits unsigned	R/O	1	ROM
	1005	5	Serial Number High	16bits unsigned	R/O	1	ROM
	1006	6	Serial Number Low	16bits unsigned	R/O	0	ROM
	1007	7	Version HW	16bits unsigned	R/O	1.0	ROM
	1008	8	Version FW	16bits unsigned	R/O	1.0	ROM
	1009	9	Status & Control	16bits unsigned	R/W	0	RAM
<b>Inputs</b>	1010	10	logicInputSct	16bits unsigned	R/O	3000	ROM
	1011	11	Section Length	16bits unsigned	R/O	3	ROM
	1012	12	Polarity	16bits unsigned	R/W	0	EEPROM
	1013	13	Hold Time	16bits unsigned	R/W	10	EEPROM
	1014	14	inSignals	16bits unsigned	R/W	0-3	RAM
<b>SIMPLE_QTY_SENSOR</b>  <b>section for Temperature sensor</b>	1015	15	sectionTypeld	Type Identifier	R/O	1000	ROM
	1016	16	sectionLength	16-bits long Integer	R/O	5	ROM
	1017	17	measuringUnit	2 chars/char[2]	R/O	"°C"	ROM
	1018	18	scaleFactor	16-bits Integer	R/O	10 mV/°C	ROM
	1019	19	minQty	16-bits Integer	R/O	0	ROM
	1020	20	maxQty	16-bits Integer	R/O	50	ROM
	1021	21	qty	16-bits Integer	R/W	0-1024	RAM
<b>REGISTERS END</b>	1022	22	End Section	16bits unsigned	R/O	0	ROM

**R/O = Read only , R/W = Read and Write**

## 1.1 InSignals bit description

inSignals bit description															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								padding_low				detectLight		detectPresence	

Bit 15 – 2: **RESERVED** Unimplemented: Read as '0'

Bit 1 : **Light detection Flag**

1 = Light was detected by the device.

0 = No light was detected by the device.

Bit 0 : **Presence detection Flag**

1 = Presence was detected by the device.

0 = No presence was detected by the device.

**NOTE:** When the presence detection Flag is set, the Modbus Occupancy sensor device will maintain this flag for 10 seconds and the Controller Unit has to read this flag within this 10 seconds otherwise this flag will be automatically cleared.

## 2. Modbus register name 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)
- **Model** = 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 Inputs

A Modbus based device containing one or many **logic input sensors** (example inputs from one or many switches or other kind of binary signals like from a presence detector) will include the next registers section:

- **logicinputSct** = LOGIC\_IN - r/o – the section type identifier for logic sensor function.
- **sectionLength** = 3 - r/o - the number of additional registers in this section
- **polarity - r/o** - the bits show the active polarity of the bits in inSignals register. 1 or 0 means the corresponding signal is active high respectively low.
- **holdTime** – r/w – the minimum time, in seconds, a signal is reported active after its transition from inactive to active happened. This enables a master to observe short active states which may happen between its queries.
- **inSignals - r/o** – the register holding maximum 16 logic inputs , one per each bit. The useful bits present in a device are specified by the datasheet of the device model and their assignment to various logic signals is specified by the system designer.

## 2.3 Simple\_QTY\_sensor

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
- **minQty** - r/o – a signed number showing the minimum measurable value multiplied by the scale factor
- **maxQty** - r/o - a signed number showing the maximum measurable value multiplied by the scale factor
- **qty** - 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 and 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.