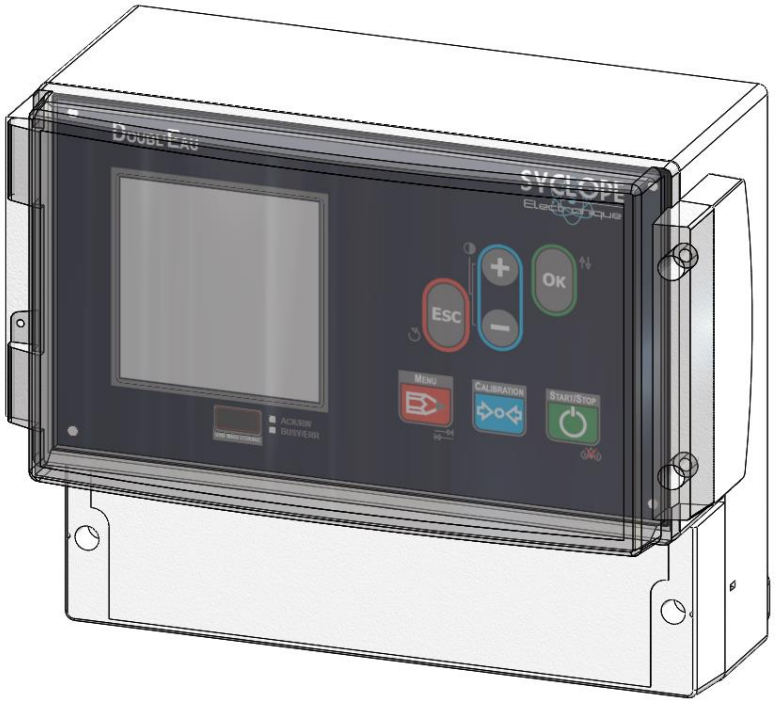


Communication instructions

SYCLOPE Electronique

Référence : DOC0338 Rev : 2

Parts of the general documentation

Part 1 : Installation and starting instructions
Part 2 : Programming instructions
▶ Part 3 : Programming communication instructions

General information:

SYCLOPE Electronique 2016® Notice of the 07/05/2016 Rev 2

Professional Analyzers/Controllers for public swimming pools.
Product line DOUBLEAU®

Part 3 : Communication instructions (Ref : DOC0338)

Editor :

SYCLOPE Electronique S.A.S.
Z.I. Aéropole pyrénées
Rue du Bruscos
64 230 SAUVAGNON - France –
Tel : (33) 05 59 33 70 36
Fax : (33) 05 59 33 70 37
Email : syclope@syclope.fr
Internet : http://www.syclope.fr

Summary

I. Use of the document

Please read this entire document before starting to install, adjust or commission your controller device, in order to ensure the safety of swimmers, users and equipment.

The information provided in this document must be strictly observed. SYCLOPE Electronique S.A.S. declines all responsibility in cases where failure to comply with the instructions of this documents is observed.

The following symbols and pictograms will be used to facilitate reading and understanding of these instructions.

- Information

► Action to be taken

➢ Item of a list or catalogue

 Risk of injury or accident

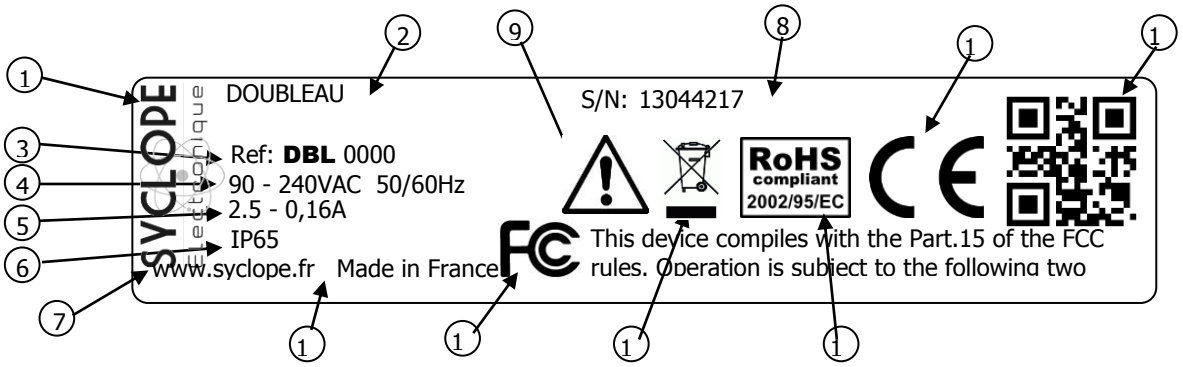
 Electric hazard

 Risk of incorrect operation or damage for the controller

 Comment

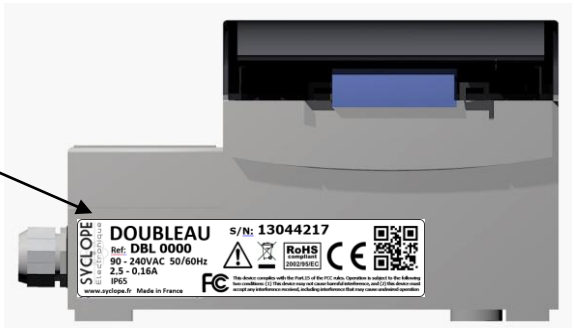
 Recyclable element

1) Labelling and localization of the identification plate



(1) Label of the manufacturer	(9) Particular risks. Read the notice
(2) Model of the product	(10) Product which can be recycled
(3) Reference of the product	(11) Limitation of dangerous substances
(4) Range of power supply	(12) EC compliance
(5) Values of the maximum current	(13) Country of the manufacturer
(6) Class of protection	(14) Manufacturer square code
(7) Identification of the manufacturer	(15) Conformity with the FCC part 15 Class B
(8) Serial number	

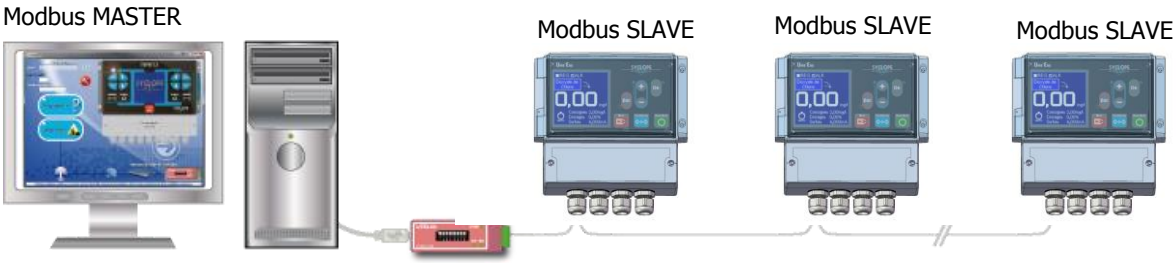
Identification plate



II. Synoptic of communication

The **SYCLOPE DOUBLEAU®** controllers have been created to be connected together to a high tech supervisor in local or distant mode. Many controllers can be connected together in different concepts.

1) Local connection with computer



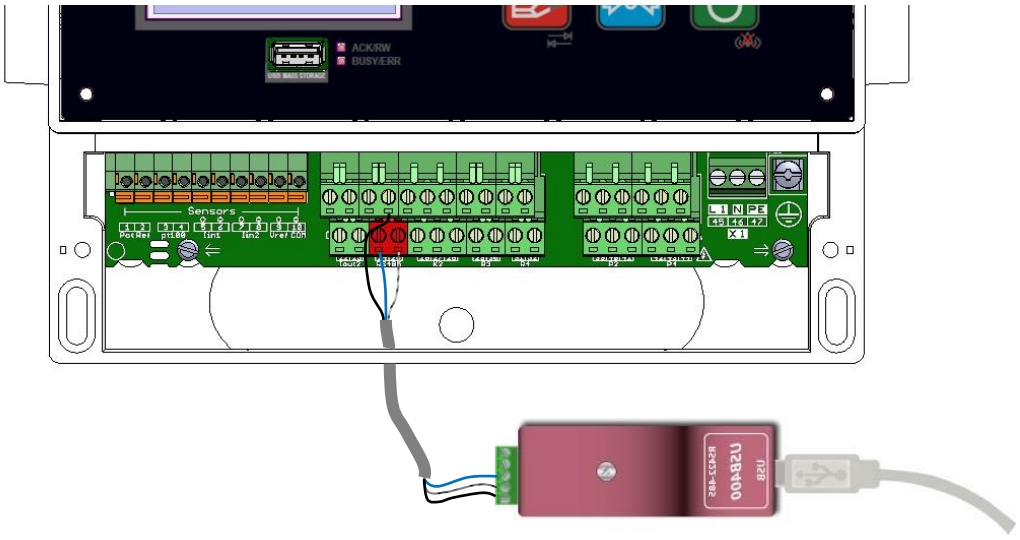
Connection to one or more controllers through the RS485 bus.

To connect your DOUBLEAU to a computer, you can use a USB/RS485 converter.

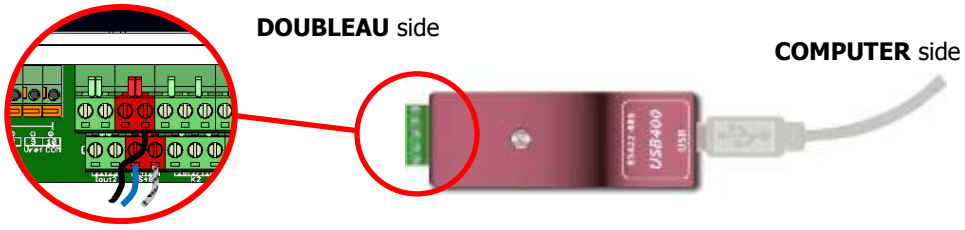
Reference	Designation
INF1021	USB 485 converter

III. Wirings

1) Wiring of the internal RS485 port and the PC converter RS485/USB



The controllers can be chained by respecting the order of the cables (putting in parallel).



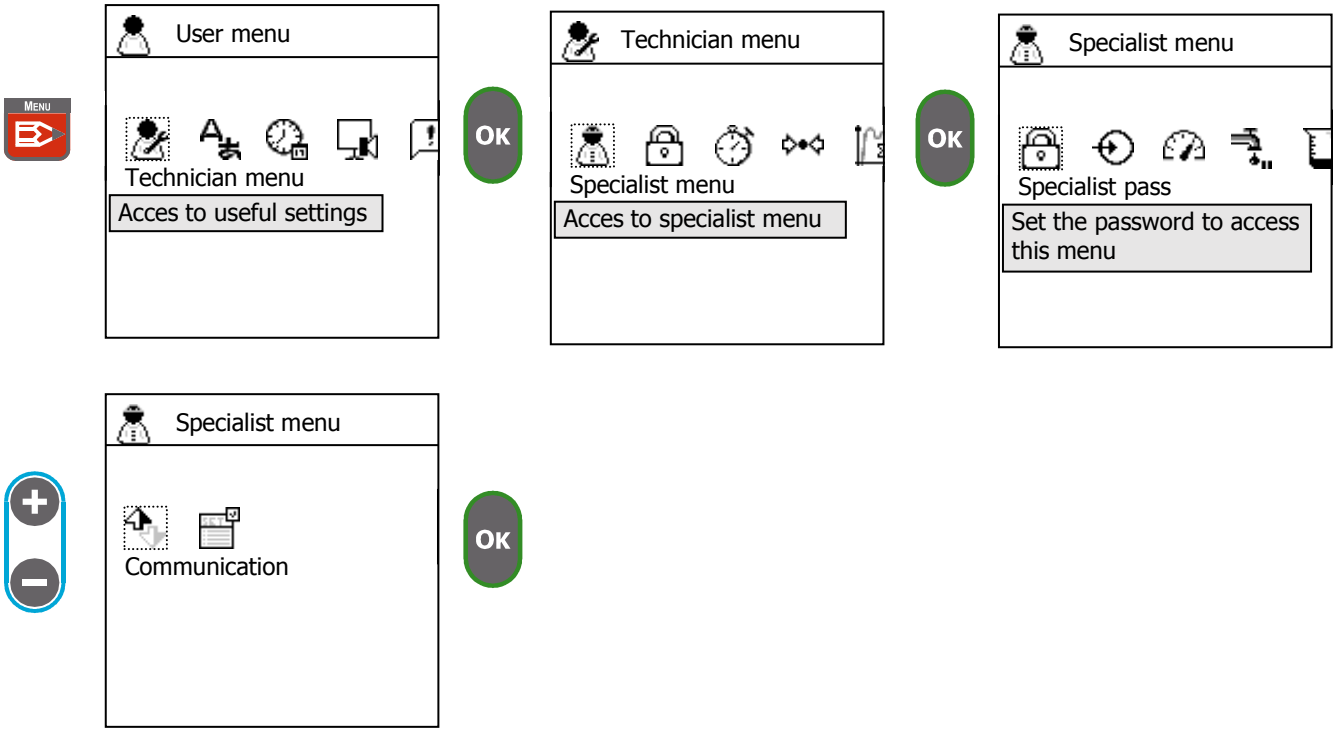
- Blue (Terminal block n°3) : AA' RS485
- White (Terminal block n°4) : BB' RS485
- Black (Terminal block n°5) : Ground RS485

Configuration : All switches are **"ON"**

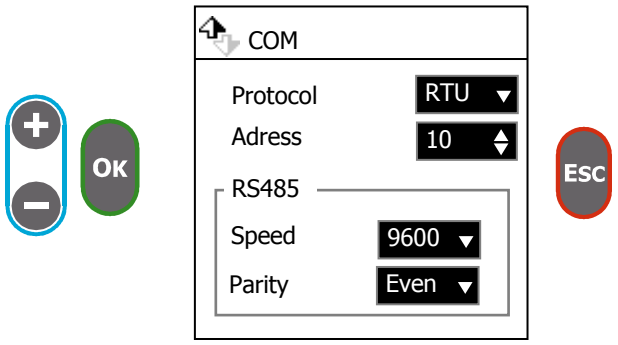
IV. Programming of the controller

1) Communication RS485 sur DOUBLEAU

To connect the controller on the bus, you must program the same communication parameters you have for all units on the bus.



► Selecting the communication protocol.
► Selecting the device identification address.
► Selecting the RS485 speed.
► Selecting the RS485 parity.

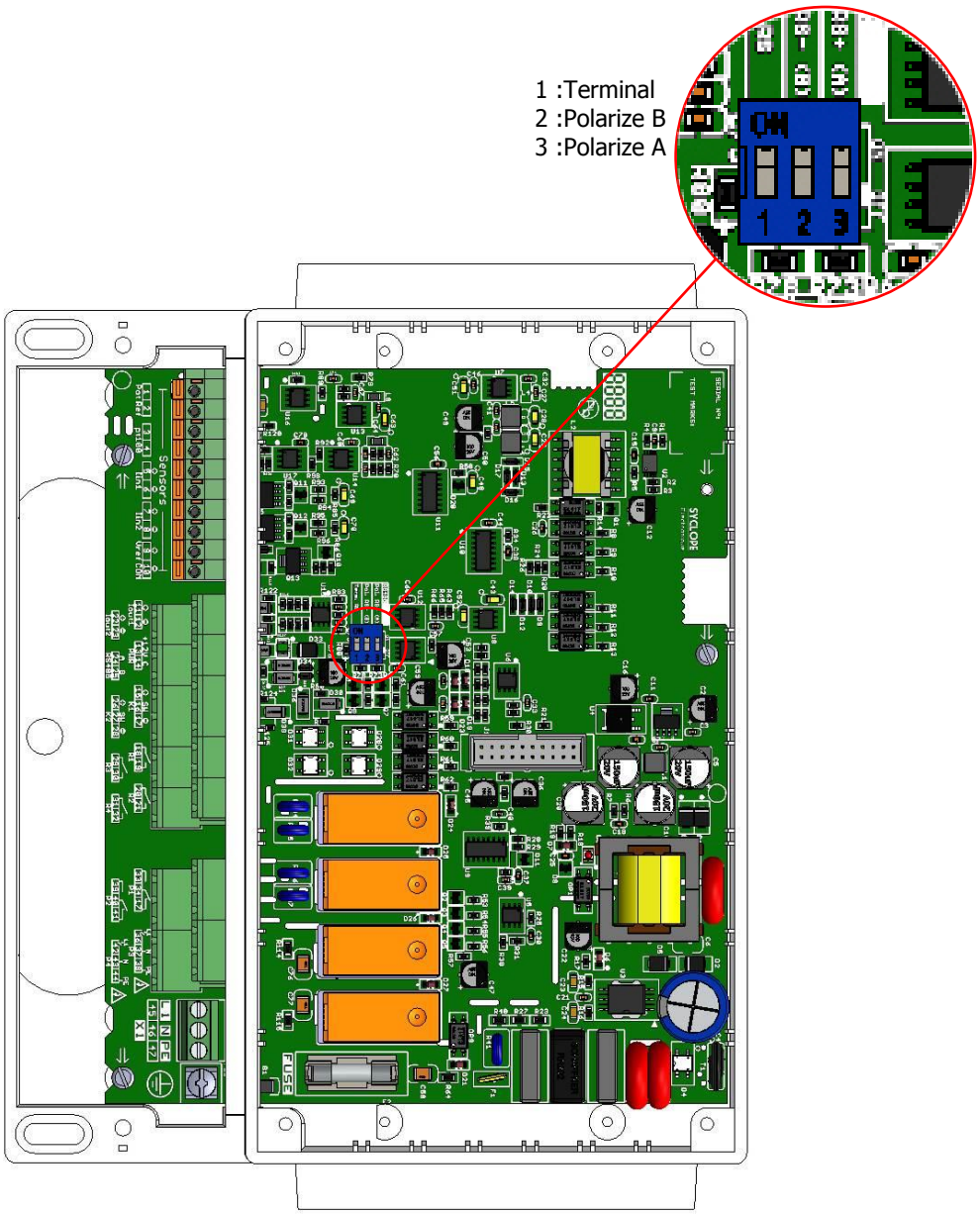


All controllers connected on the same RS485 bus must have the same speed, same parity and each controller must have a different address to ensure a good communication.

Name	Signification	Setting	Default value
Mode	Communication protocol	RTU/ASCII	RTU
Address	«slave ID» address of the device	1…247	10
Speed	Communication speed	300, 1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200	9600
Parity	Parity of serial communication	None, Even, Odd	Even

Three switches are located on the electronic card to manage the termination and the polarization resistors of the RS485 line. By default, the controllers are delivered with switches on the OFF position.

1 :Terminal
2 :Polarize B
3 :Polarize A



I. MODBUS registers

1) Address of Modbus register

Registers are indexed according MODBUS standard. There are "HOLDING REGISTER" kind range from 40001 to 49999.

Some of Modbus software and plc, use address coding register range from 0 to 65535.

Therefore, ModBus register 40001 corresponding to Modbus address 0, 40002 corresponding to address 1 and so on.

ModBus Register	Register size 16bits	Name	Attributes	Format	Description
Configuration					
40001	788	eeprom	rw	STRUCT	Mémoire
Interfaces					
41001	2	signal_POT	r	REAL	POT signal [mV]
41003	2	signal_RTD	r	REAL	RTD resistance [ohm]
41005	2	signal_IIN1	r	REAL	IN1 current [mA]
41007	2	signal_IIN2	r	REAL	IN2 current[mA]
41009	2	signal_K1	r	REAL	K1 frequency [Hz]
41011	2	signal_K2	r	REAL	K2 frequency [Hz]
41013	1	supply_IN	r	BOOL	0=12V / 1=24V
41014	1	state_IN1	r	BOOL	0=OPEN / 1=CLOSE
41015	1	state_IN2	r	BOOL	0=OPEN / 1=CLOSE
41016	1	state_K1	r	BOOL	0=OPEN / 1=CLOSE
41017	1	state_K2	r	BOOL	0=OPEN / 1=CLOSE
41018	1	state_P1	r	BOOL	0=OPEN / 1=CLOSE
41019	1	state_P2	r	BOOL	0=OPEN / 1=CLOSE
41020	1	state_P3	r	BOOL	0=OPEN / 1=CLOSE
41021	1	state_P4	r	BOOL	0=OPEN / 1=CLOSE
41021	1	state_R1	r	BOOL	0=OPEN / 1=CLOSE
41023	1	state_R2	r	BOOL	0=OPEN / 1=CLOSE
41024	1	state_R3	r	BOOL	0=OPEN / 1=CLOSE
41025	1	state_R4	r	BOOL	0=OPEN / 1=CLOSE
41026	2	current_IOUT1	r	REAL	OUT1 current [mA]
41028	2	current_IOUT2	r	REAL	OUT2 current [mA]
41030	2	Timestamp Local	r	DWORD	Time since 1st January 1970 0h00 [s]
41032	2	Run time	r	DWORD	Time from device start
Values and states					
41101	1	device_state	rw	WORD	Bit 0: device enabled Bit 1: timer enabled Bit 2: device delayed Bit 3: device stop due to timer
41201	2	param_1_state	rw	DWORD	Bit 0: control and alarm processing Bit 1: sensors boot delay Bit 2: temporary pause Bit 3: flow switches or flowmeter (true == flowing) Bit 4: service is needed Bit 5: dosing Bit 6: alarm active Bit 7: timer pause control and alarm

					Bit 8: sensors over range or disconnected Bit 9: sensor over measurement range Bit 10: sensor measure unstable Bit 11: low alarm threshold Bit 12: high alarm threshold Bit 13: max dosing time or tank empty bit14: use timer bit15: remote control enabled bit16: configuration error
41203	2	param_1_measure_value	r	REAL	Measure value [measure_unit]
41205	2	param_1_control_w	rw	REAL	Set point [measure_unit]
41207	2	param_1_dosage_u	r	REAL	Control value [1/1]
41209	2	param_1_alarm_high	rw	REAL	Low alarm threshold value [measure_unit]
41211	2	param_1_alarm_low	rw	REAL	High alarm threshold value [measure_unit]
41301	2	param_2_state	rw	DWORD	Bit 0: control and alarm processing Bit 1: sensors boot delay Bit 2: temporary pause Bit 3: flow switches or flowmeter (true == flowing) Bit 4: service is needed Bit 5: dosing Bit 6: alarm active Bit 7: timer pause control and alarm Bit 8: sensors over range or disconnected Bit 9: sensor over measurement range Bit 10: sensor measure unstable Bit 11: low alarm threshold Bit 12: high alarm threshold Bit 13: max dosing time or tank empty bit14: use timer bit15: remote control enabled bit16: configuration error
41303	2	param_2_measure_value	r	REAL	Measure value [measure_unit]
41305	2	param_2_control_w	rw	REAL	Set point [measure_unit]
41307	2	param_2_dosage_u	r	REAL	Control value [1/1]
41309	2	param_2_alarm_high	rw	REAL	Low alarm threshold value [measure_unit]
41311	2	param_2_alarm_low	rw	REAL	High alarm threshold value [measure_unit]
41401	2	sensor_POT_value	r	REAL	POT sensor measure value [sensor_unit]
41403	2	sensorl_RTD_value	r	REAL	RTD sensor measure value [sensor_unit]
41405	2	sensor_IN1_value	r	REAL	IN1 sensor measure value [sensor_unit]
41407	2	sensor_IN2_value	r	REAL	IN2 sensor measure value [sensor_unit]
41409	2	sensor_K1_value	r	REAL	K1 sensor measure value [sensor_unit]
41411	2	sensor_K2_value	r	REAL	K2 sensor measure value [sensor_unit]
Device					
42001	22	device	r	STRUCT	State and value of device
Parameters					
42101	124	param_1	r	STRUCT	State and value of parameter 1
42301	124	param_2	r	STRUCT	State and value of parameter 2
Sensors					
42501	34	sensor_POT	r	STRUCT	State and value of POT sensor
42551	34	sensor_RTD	r	STRUCT	State and value of RTD sensor
42601	34	sensor_IIN1	r	STRUCT	State and value of IN1 sensor
42651	34	sensor_IIN2	r	STRUCT	State and value of IN2 sensor
42701	34	sensor_K1	r	STRUCT	State and value of K1 sensor
42751	34	sensor_K2	r	STRUCT	State and value of K2 sensor
Switches					
42801	10	switch_IIN1	r	STRUCT	State of IN1 switch

42821	10	switch_IIN2	r	STRUCT	State of IN2 switch
42841	10	switch_K1	r	STRUCT	State of K1 switch
42861	10	switch_K2	r	STRUCT	State of K2 switch
Outputs 0/4-20mA					
42901	16	iout_1	r	STRUCT	State and value of OUT1 output
42921	16	iout_2	r	STRUCT	State and value of OUT2 output
Relays					
43001	56	relay_P1	r	STRUCT	State and value of P1 relay
43101	56	relay_P2	r	STRUCT	State and value of P2 relay
43201	56	relay_P3	r	STRUCT	State and value of P3 relay
43301	56	relay_P4	r	STRUCT	State and value of P4 relay
43401	56	relay_R1	r	STRUCT	State and value of R1 relay
43501	56	relay_R2	r	STRUCT	State and value of R2 relay
43601	56	relay_R3	r	STRUCT	State and value of R3 relay
43701	56	relay_R4	r	STRUCT	State and value of R4 relay
Events					
44001	36	device_1_event	r	STRUCT	Device timer event state
44051	36	param_1_event	r	STRUCT	Parameter 1 timer event state
44101	36	param_2_event	r	STRUCT	Parameter 1 timer event state
44151	36	relay_event_P1	r	STRUCT	Relay P1 timer event state
44201	36	relay_event_P2	r	STRUCT	Relay P2 timer event state
44251	36	relay_event_P3	r	STRUCT	Relay P3 timer event state
44301	36	relay_event_P4	r	STRUCT	Relay P4 timer event state
44351	36	relay_event_R1	r	STRUCT	Relay R1 timer event state
44401	36	relay_event_R2	r	STRUCT	Relay R2 timer event state
44451	36	relay_event_R3	r	STRUCT	Relay R3 timer event state
44501	36	relay_event_R4	r	STRUCT	Relay R4 timer event state
Display					
45000	2048	screen	r	STRUCT	2bpp display buffer 128x128pix

2) Data formatting

BOOL
"bool" use 1 register and can have two value 0 or 1.

Example:
Register 41018 is the state of P1 relay.
REG(41018) = 0: relay open
REG(41018) = 1: relay closed

REAL
"real" use 2 registers and can store a 32bit floating point decimal value.

Example:
Register 41102 is the measure value of parameter 1, the value unit is the measurement unit selected in the measure menu.
For 1.94ppm measure value, the hexadecimal coding is 0x3FF851EC.
REG(41103) = 0x51EC
REG(41104) = 0x3FF8

WORD
"word" use 1 register and can store 16bit integer value or bit field flags.

Example (bits):
Register 41101 hold the state flags of the device.
REG(4101) = b0000000000000101

REG(41101)(bit00) = 1: device is enable
REG(41101)(bit01) = 0: timer is not enable
REG(41101)(bit02) = 1: control and alarm processing of at less one parameter is delay
REG(41101)(bit03) = 0: no timer stop condition
REG(41101)(bit04) = 0: not used
REG(41101)(bit05) = 0: not used
REG(41101)(bit06) = 0: not used
REG(41101)(bit07) = 0: not used
REG(41101)(bit08) = 0: not used
REG(41101)(bit09) = 0: not used
REG(41101)(bit10) = 0: not used
REG(41101)(bit11) = 0: not used
REG(41101)(bit12) = 0: not used
REG(41101)(bit13) = 0: not used
REG(41101)(bit14) = 0: not used
REG(41101)(bit15) = 0: not used

DWORD
"dword" use 2 registers and can store 32bits integer value or bit field flags.

Example:
Register 41030 hold the local time, this time is the number of seconds since the January 1st 1970.
April 27th 2015 at 3h35min19sec AM correspond to 1430141719 second since reference date, hexadecimal value is 0x553E3B17.
REG(41032) = 0x3B17
REG(41032) = 0x553E

STRUCT (device)

This data block holds all the states, values and settings of the device.

Name	Size [bytes]	Offset [Bytes]	Kind	Description
dev	1	0	byte	Internal
name	12	1	string	Device name
align	3	13	--	Internal
fd	4	16	integer	internal
flag	1	20	bits	bit0: control and alarm enabled bit1: timer enabled bit2: device delayed bit3: device stop due to timer
align	3	21	--	Interne
device.calendar.flag	1	24	bits	bit0: calendar timer processing bit1: calendar timer event pending bit2: calendar timer is temporary halted
align	3	25	--	Internal
device.calendar.event_list	4	28	--	Internal
device.calendar.next	4	32	--	Internal
device.param.flag	3	36	bits	Bit 0: control and alarm processing Bit 1: sensors boot delay Bit 2: temporary pause Bit 3: flow switches or flowmeter (true == flowing) Bit 4: service is needed Bit 5: dosing Bit 6: alarm active Bit 7: timer pause control and alarm Bit 8: sensors over range or disconnected Bit 9: sensor over measurement range Bit 10: sensor measure unstable Bit 11: low alarm threshold Bit 12: high alarm threshold Bit 13: max dosing time or tank empty bit14: use timer bit15: remote control enabled bit16: configuration error
align	1	39	--	Internal
next	4	40	--	Internal

Example:
To read the state of the device the base register is REG(42001).

byte offset of "flag" is (1+12+3+4) = 20
register offset is 20/2 = 10
the address of register to read "flag" is REG(42001+20) = REG42021)

REG(42021) = 0x0100
Data blocks are "little endian" encoding therefore byte order are inverted.
Flag = 0x01 device is running.

STRUCT (param)

This data block holds all the states, values and settings of the parameters.

Name	Size [bytes]	Offset [bytes]	Kind	Description
par	1	0	byte	Internal
align	3	1	--	Internal
fd	4	4	integer	internal
flag	3	8	bits	Bit 0: control and alarm processing Bit 1: sensors boot delay Bit 2: temporary pause Bit 3: flow switches or flowmeter (true == flowing) Bit 4: service is needed Bit 5: dosing Bit 6: alarm active Bit 7: timer pause control and alarm Bit 8: sensors over range or disconnected Bit 9: sensor over measurement range Bit 10: sensor measure unstable Bit 11: low alarm threshold Bit 12: high alarm threshold Bit 13: max dosing time or tank empty bit14: use timer bit15: remote control enabled bit16: configuration error
align	1	11	--	interne
decice	4	12	--	interne
measure_sensor	16	16	--	interne
measure_kind	1	32	byte	Kind of parameter: 0: Undefined 1: Free chlorine 2: Active chlorine 3: Total chlorine 4: Chloramines 5: Chlorite 6: Chlorite dioxide 7: H2O2 8: BCDMH 9: DBDMH 10: Free bromine 11: Active bromine. 12: Total bromine 13: APA 14: Ozone 15: Dissolved oxygen 16: Nitrate 17: PHMB 18: Salinity 19: TDS 20: Turbidity 21: Conductivity 22: Temperature 23: Flow 24: pH 25: RedOx 26: Chloride 27: Amonia 28: Fluoride 29: ISE 30: Volume
measure_unit	1	33	byte	Unit of measurement 0: Undefined

				1: Void 2: Decade 3: pH 4: ppb 5: ppm 6: µg/l 7: mg/l 8: g/l 9: % 10: µS/cm² 11: mS/cm² 12: NTU 13: FNU 14: °K 15: °C 16: °F 17: °R 18: mA 19: mV 20: Hz 21: str/min 22: ms 23: sec 24: min 25: h 26: l 27: m3 28: l/min 29: l/h 30: m3/h 31: imp/l 32: imp/m3 33: Ohm 34: mOhm 35: pulse
align	2	34	--	Internal
measure_min_value	4	36	float	Low measurement scale value
measure_max_value	4	40	float	High measurement scale value
measure_value	4	44	float	Measure value
measure_m_factor	4	48	float	Measure correction factor [1/1]
measure_t_factor	4	52	float	Temperature correction factor [%/°C]
alarm_flag	1	56	bits	Bits pending alarms bit0: measure value upper the high threshold bit1: measure value under the low threshold bit2: overdose time reached or tank empty bit3: sensor disconnected or out of range
alarm_threshold_delay	1	57	byte	Delay of threshold alarms [s]
alarm_threshold_tick	1	58	byte	Internal
align	1	59	--	Internal
alarm_threshold_hyst	4	60	float	Thresholds alarm hysteresis
alarm_threshold_low	4	64	float	Low threshold
alarm_threshold_high	4	68	float	High threshold
flow_sensor	4	72	--	Internal
flow_switch	16	76	--	Internal
flow.op	1	92	byte	Flowing flag condition operation 0: at less one 1 = all
flow_unit	1	93	byte	Flowmeter unit: 28 : l/min 29 : l/h 30 : m3/h

align	2	94	--	Internal
flow_threshold	4	96	float	Flowrate threshold to indicate flowing condition
flow_q_min	4	100	float	Flowrate low value for control compensation x0%
flow_q_max	4	104	float	Flowrate high value for control compensation x100%
flow_q	4	108	float	Flowrate value
control_flag	1	112	bits	bit0: 0 = 1DOF 1 = 2DOF bit[1~3]: control mode 0 = disabled 1 = hysteresis 2 = threshold 3 = PID Bit4: hold function enable
align	3	113	--	Internal
control_w	4	116	float	Control set point
control_x_dead	4	120	float	Dead band or hysteresis depending control mode
control_xp	4	124	float	Reciprocal value of proportional gain
control_ki	2	128	float	Integral coefficient
control_kd	2	130	float	Derivative coefficient
control_kb	4	132	float	Back compute coefficient
control. threshold_low	4	136	float	Low control threshold
control. threshold_high	4	140	float	High control threshold
control_z_y	4	144	float	Delayed control output
control_z_ex	4	148	float	Delayed control error or input depending control DOF
control_z_dex	4	152	float	Delayed derivative part error
control_sum_e	4	156	float	Integral error part
dosage_flag	1	160	bits	bit[0~1]: control direction 0 = up 1 = down 2 = both bit2: control output is flowrate proportionnal bit3: dosage halted
align	1	161	--	Internal
dosage.tick	2	162	short	Overdose time counter
dosage.control_time	2	164	short	Overdose max time
align	2	166	--	Internal
dosage.control_threshold	4	168	float	Overdose control output threshold
dosage_u_bias	4	172	float	Bias control load
dosage_u	4	176	float	Control dosing output
tank.switch_direct	4	180	--	Internal
tank.switch_invert	4	184	--	Internal
tank.sensor_direct	4	188	--	Internal
tank.sensor_invert	4	192	--	Internal
tank. threshold_direct	4	196	float	Empty tank threshold
tank. threshold_invert	4	200	float	Empty tank threshold
remolte.calendar.flag	1	204	bits	bit0: calendar timer processing bit1: calendar timer event pending bit2: calendar timer is temporary halted
align	3	205	--	Internal
remote.calendar.event_list	4	208	--	Internal
remote.calendar.next	4	212	--	Internal
remotre.flag	1	216	bits	bit[0~1]: remote control mode 0 = disable 1 = timer 2 = sensor 3 = switch

align	3	217	--	Internal
remote.sensor	4	220	--	Internal
remote.control.w	4	224	float	Current remote control set point
remote.control.w_target	4	228	float	Wanted remote control set point
remote.control.kv	1	232	byte	Setpoint variation speed
align	3	233	--	Internal
next	4	236	--	Internal

Example:
To read the flowrate of parameter 1 the base register is REG(42101).

byte offset of "flow_q" is 108
register offset is 108/2 = 54
data is stored on 4 bytes so the address of registers to read "flow_q" are REG(42101+54) = REG(42055) and REG(42056).

REG(42155) = 0xA470
REG(42156) = 0x4541
Data blocks are "little endian" encoding therefore value is 0x414570A4, 12.34 float 32bits

STRUCT (sensor)

This data block holds all the states, values and settings of the sensors.

Name	Size [bytes]	Offset [bytes]	Kind	Description
sen	1	0	byte	Internal
ch	1	1	byte	Internal
def	1	2	byte	Internal
align	1	3	--	interne
fd	4	4	integer	Internal
flag	2	8	bits	bit0: fault state bit1: unplugged bit2: interface reaches the high limit bit3: interface reaches the low limit bit4: measure reaches the high limit bit5: measure reaches the low limit bit6: measure unstable bit7: service/calibration required bit8: delayed state
kind	1	10	byte	Kind of measure: 0: Undefined 1: Free chlorine 2: Active chlorine 3: Total chlorine 4: Chloramines 5: Chlorite 6: Chlorite dioxide 7: H2O2 8: BCDMH 9: DBDMH 10: Free bromine 11: Active bromine. 12: Total bromine 13: APA 14: Ozone 15: Dissolved oxygen 16: Nitrate 17: PHMB 18: Salinity 19: TDS 20: Turbidity 21: Conductivity 22: Temperature 23: Flow 24: pH 25: RedOx 26: Chloride 27: Amonia 28: Fluoride 29: ISE 30: Volume
unit	1	11	byte	Unit of measurement 0: Undefined 1: Void 2: Decade 3: pH 4: ppb 5: ppm 6: µg/l 7: mg/l 8: g/l 9: % 10: µS/cm²

				11: mS/cm² 12: NTU 13: FNU 14: °K 15: °C 16: °F 17: °R 18: mA 19: mV 20: Hz 21: str/min 22: ms 23: sec 24: min 25: h 26: l 27: m3 28: l/min 29: l/h 30: m3/h 31: imp/l 32: imp/m3 33: Ohm 34: mOhm 35: pulse
transducer	1	12	byte	Kind of transducer: 0: None 1: 0…20mA 2: 4…20mA 3: pH -> 4…20mA 4: RedOx -> 4…20mA 5: pt100 -> 4…20mA 6: fluoride -> 4…20mA 7: fluoride (100) -> 4…20mA 8: ISOCAP pH -> 4…20mA 9: ISOCAP RedOk -> 4…20mA 10: UNISO P -> 4…20mA 11: UNISO R1 -> 4…20mA 12: UNISO R -> 4…20mA 13: UNISO B -> 4…20mA 14: 0…2000mV 15: 0…-2000mV 16: (potentiometric) mV 17: pulse frequency 18: PT100 19: PT1000
align	1	13	--	Internal
delay	2	14	short	Startup delay 0.5s
tick	2	16	short	Startup time counter 0.5s
fault_tick	1	18	byte	Error time counter 0.5s
align	1	19	--	Internal
min_value	4	20	float	Measurement range low value
max_value	4	24	float	Measurement range high value
cal_slope	4	28	float	Slope
cal_offset	4	32	float	Offset
std_slope	4	36	float	Slope before calibration
std_offset	4	40	float	Offset before calibration
z_dex	4	44	--	Internal
interface	4	48	float	Interface value
signal	4	52	float	Sensor signal value
std_value	4	56	float	Measure value without calibration
value	4	60	float	Measure value after calibration
next	4	64	--	Internal

Example:
To read the measure unit of sensor connected on IN1 input, the base register is REG(42601).

byte offset of "unit" is 11
register offset is 11/2 = 5
address of registers to read "unit" is REG(42601+5) = REG(42606)

REG(42606) = 0x0205
Data blocks are "little endian" encoding therefore "unit" is the lower byte 0x05.
0x05 = [ppm]

STRUCT (switch)

This data block holds all the states, values and settings of the switches.

Name	Size [bytes]	Offset [bytes]	Kind	Description
sw	1	0	byte	Internal
ch	1	1	byte	Internal
align	2	2	--	Internal
fd	4	4	integer	Internal
flag	1	8	bits	bit0: switch enable bit1: stand; NO=0; NC=1 bit2: internal bit3: physical state; open=0; closed=1 bit4: delayed state; open=0; closed=1 bit5: active state, relative to stand and delayed state
align	1	9	--	Internal
delay	2	10	short	Debounce delay, ½ sec
tick	2	12	short	Debounce time
align	2	14	--	Internal
next	4	16	--	Internal

Example:
To read the debounce time K1 switch, the base register is REG(42841).

byte offset of "tick" is 12
register offset is 12/2 = 6
address of registers to read "tick" is REG(42841+6) = REG(42847)

REG(42847) = 0x0A00
Data blocks are "little endian" encoding therefore byte order are inverted
tick = 0x000A
"tick" unit is ½ sec so the debounce time is 0xA x 0.5 sec = 5 sec.

STRUCT (relay)

This data block holds all the states, values and settings of the relays.

Name	Size [byte]	Offset [byte]	Kind	Description
rel	1	0	byte	Internal
ch	1	1	byte	Internal
align	2	2	--	Internal
fd	4	4	integer	Internal
flag	1	8	bits	bit[0~2]: mode 0 = disable 1 = parameter dosage 2 = parameter alarm 3 = device alarml 4 = switch state 5 = relay state 6 = timer bit3: stand NO=0; NC=1 bit4: physical state, open=0; closed=1 bit5: active state, active=1 bit6: internal
align	3	9	--	Internal
param/device /switch/relay	4	12	--	Internal
alarm.delay	1	16	integer	Delay to set relay during an alarm
alarm.tick	1	17	integer	Pending alarm time counter
alarm.pending	1	18	bits	bits pending alarm bit0: no water flowing bit1: measure value under low threshold bit2: measure value upper high threshold bit3: sensor(s) out of range or disconnected bit4: sensor out of range bit5: overdose (max time or empty tank) bit6: sensor delayed
alarm.enable	1	19	bits	bits to enable alarms bit0: flowing detection bit1: low threshold bit2: high threshold bit3: sensor error bit4: sensor out of range bit5: overdose bits6: delayed
dosage.u_min	4	20	float	
dosage.u_max	4	24	float	
dosage.u	4	28	float	Dosing power [1/1]
dosage.period	4	32	integer	Relay cycle time
dosage.min_width	2	36	integer	Minimum time between state changes
align	2	38	--	Internal
dosage.compute_time	4	40	integer	Internal
dosage.tilt_time	4	44	integer	Internal
dosage.ref_time	4	48	integer	Internal
dosage.delay	4	52	integer	Internal
dosage.flag	1	56	bits	bits[0~1]: mode 0 = ON/OFF 1 = PWM 2 = PFM bit2: dosage direction, 0=direct; 1=invert bit3: internal bit[4~5]: action 0 = none 1 = direct 2 = invert
dosage.q_unit	1	57	bits	

align	2	58	--	Internal
timer.calendar.flag	1	60	bits	bit0: calendar timer processing bit1: calendar timer event pending bit2: calendar timer is temporary halted
align	3	61	--	Internal
timer.calendar.event_list	4	64	--	Internal
timer.calendar.next	4	68	--	Internal
timer.action	1	72	bits	
align	3	73	--	Internal
delay_on	1	76	integer	
delay_off	1	77	integer	
tick_active	2	78	integer	
timer.handler	4	80	--	Internal
timer.proc	4	84	--	Internal
timer.delay	4	88	integer	
timer.trig_time	4	92	integer	
timer.next	4	96	--	Internal
next	4	100	--	Internal

Example:
To read the active state of P1 relay, the base register is REG(43001).

byte offset of "flag" is 8
register offset is 8/2 = 4
address of registers to read "flag" is REG(43001+4) = REG(43005)

REG(43005) = 0x3500
Data blocks are "little endian" encoding therefore "flag' byte is the highest byte
flag = 0x35 = 0b00110110
active state is bit 5 = 1, relay is active.

STRUCT (iout)

This data block holds all the states, values and settings of the iouts (0/4…20mA current outputs).

Name	Size [bytes]	Offset [bytes]	Kind	Description
out	1	0	integer	Internal
ch	1	1	integer	Internal
align	2	2	--	Internal
param / sensor	4	4	--	Internal
fd	4	8	integer	Internal
flag	2	12	bits	bit0: current range 0: 4…20mA 1: 0…20mA bit[1~2]: fault current 0: 0mA 1: 0 or 4mA according the range 2: 2.6mA bit[3~4]: inhibition current 0: no change 1: 0mA 2: 0 or 4mA according the range 3: 3.4mA bit[5~6]: out of range current 0: 21.7mA 1: 20mA 2: 20.8mA Bit[7~9]: mode 0: parameter dosage value 1: parameter control 2: parameter measure 3: sensor measure 4: sensor interface bit10: inhibition state bit11: halted state
align	2	14	--	internal
point_0_4mA	4	16	float	Measure or control value equivalent to 0/4mA
point_20mA	4	20	float	Measure or control value equivalent to 20mA
current	4	24	float	Current loop value [mA]
next	4	28	--	Internal

Example:
To read the current of IOUT2 output, the base register is REG(42921).

byte offset of "current" is 28
register offset is 28/2 = 14
address of registers to read "current" is REG(42921+14) = REG(42935) and REG(42936)

REG(42935) = 0xA470
REG(42936) = 0x4541
Data blocks are "little endian" encoding therefore value is 0x414570A4, 12.34mA.

STRUCT (calendar)

This data block holds all the states, values and settings of the calendars to manage timers events.

Name	Size [bytes]	Offset [bytes]	Kind	Description
calendar.flag	1	0	bits	bit0: calendar enabled bit1: event(s) pending bit2: halted
align	3	1	--	Internal
calendar.event_list	4	4	--	Internal
calendar.next	4	8	--	Internal

STRUCT (calendar_event)

Name	Size [bytes]	Offset [bytes]	Kind	Description
name	8	0	string	
ev	1	8		
align	3	9	--	Internal
fd	4	12	integer	Internal
flag	2	16	bits	Active days bit0: monday bit1: tuesday bit2: wednesday bit3: thursday bit4: friday bit5: saturday bit6: sunday bit7: internal bit8: enable bit9: pending
align	2	18	--	Internal
date_start	4	20	integer	Event start date and time
date_end	4	24	integer	Event end date and time
interval	1	28	integer	Number of week between event repetitions
align	3	29	--	Internal
next	4	32	--	Internal

SYCLOPE Electronique S.A.S.
Z.I. Aéropole pyrénées
Rue du Bruscos
64 230 SAUVAGNON - France –
Tel : +33 (0) 559 337 036
Fax : +33 (0) 559 337 037
Email : contact@syclope.fr
Internet : http://www.syclope.fr