CirPark Scada

Scada software for parking management

XML Protocol Manual

V4.2.2

MU610105-16A-EN-CirPark Scada-5-XML Protocol

1 Variables

In order to use references to the variables measured by devices, in Scada, reports, charts and tables, it will be necessary to know the encoding used by the software for each of them.

The basic encoding is made of the device name of the device and the variable code separated by one point.

name.variable

In this way the software will know which variable is involved and which device should be asked for the said variable.

This encoding (variable-name) can be used in formulas, graphs and tables; although in the case of reports, graphs and tables we can display variables saved in values log files.

On those occasions we wish to discriminate a variable, as long as the variable can be discriminated, for instance in the case of energy, the discriminator information will be added to the basic coding.

name_discriminator @ name_type_time: name.variable

Here the name of the discriminator and the name of the type of hour upon which the variable values will be discriminated are indicated.

This type of encoding with discriminators may only be used in reports, graphs and tables, and with variables that have been saved in values of log files.

2 Devices

2.1 Converters

Variable	Description
PTIME	Line pooling time (the time it takes the motor to read the status of all the converter sensors).

2.2 Sensors

Variable	Description
MODE*	Sensor mode: 0 – Normal (green) 1 – Reserved (orange) 2 – Prohibited (red)
STATE	Sensor Status: 0 – Error 1 – free 2 – occupied
TYPE	Space type 0 – Normal (green) 1 – Handicap (blue) 2 – Electric vehicle (white)

	3 – Family (light orange)
CONFSTATE	Configured sensor status 0 – Normal 1 – Reserved 2 – Prohibited
BUSYDATE	Date and time since the space has been occupied
BUSYTIME	Occupation time in seconds
ALARMTIME	Parking alarm time in seconds
ALARM	Mark if the sensor is in alarm status due to parking time "T" – Yes "F" – No
HANDICAPPED*	Activates (1) if sensor is reserved for handicapped and (0) if not
MOVEMENTS	Returns a history log with the number of sensor occupancies in the period
OCCUPANCY	Returns a history log with percentages of the sensor occupancy time in the period
STEAL	Indicates if the alarm is activated because of theft
STOLEN	Indicates if the alarm was tripped because of theft
STEAL_YEAR	Year of the date of minimum occupation
STEAL_MONTH	Month of the date of minimum occupation
STEAL_DAY	Day of the date of minimum occupation
STEAL_HOUR	Hour of the date of minimum occupation
STEAL_MINUTE	Minute of the date of minimum occupation
RESERVE_BEGIN_YEAR	Year the reservation began
RESERVE_BEGIN_MONTH	Month the reservation began
RESERVE_BEGIN_DAY	Day the reservation began
RESERVE_BEGIN_HOUR	Hour the reservation began
RESERVE_BEGIN_MINUTE	Minute the reservation began
RESERVE_END_YEAR	Year the reservation ends
RESERVE_END_MONTH	Month the reservation ends
RESERVE_END_DAY	Day the reservation ends
RESERVE_END_HOUR	Hour the reservation ends
RESERVE_END_MINUTE	Minute the reservation ends
RESERVE	Programmed reservation
BRIGHT	Enables forcing the sensor brightness (it does not display it, but only forces the value).
OCCUPATION_ALARM	Variable that marks the maximum occupation time of a space (in hours)
OCCUPATION_ALARM_FLAG	Variable that marks the type of maximum occupation time of a space 0 – General 1 – Personalised 2 - Deactivated
LED_ON	(ONLY ONE) Which LEDS are lit 0 – ALL ON 1 – LEFT ON 2 – RIGHT ON 3 – ALL OFF

2.3 Displays

Variable	Description
VALUE	Display value
BRIGHT	Enables forcing the display brightness
SHOW_FREE_TEXT	Shows free text "T" – Yes "F" – No
FREE_TEXT	Free text
ZERO_TEXT	Zero text
DIGIT_COLOR	Digit colour in BGR
ZERO_COLOR	Indicates the colour in BGR of the character to be shown as zero
FREE_TEXT_COLOR	Indicates the colour in BGR of the free text
IMAGEDXVMS	Image for the DXVMSF and DXVMSP

DIGIT_NUMBER	Number of digits
ARROW_POSITION	Indicates the position of the cross arrow character 0 – Left 1 – Right
ZERO_CHARACTER	Indicates the character to be represented when there is a zero being displayed 0 – Arrow 1 – Cross 2 – Text
ARROW_TYPE	Towards the direction of the arrow 1 – Right 2 – Up 3 – Left 4 – Down 7 - Up/Left 8 . Down/Right 9 – Up/Left 10 – Down/Left
REFRESH_TIME	Display refresh time
ARROW_MOVEMENT	Arrow movement 0 – Yes 1 – No

2.4 CL-Park

Variable	**Description**
BRIGHT	Enables forcing the display brightness

2.5 MR4/dp

Variable	**Description**
DI1	State of input 1, Open (0) or closed (1)
DI2	State of input 2, Open (0) or closed (1)
DI3	State of input 3, Open (0) or closed (1)
DI4	State of input 4, Open (0) or closed (1)
DI5	State of input 5, Open (0) or closed (1)
DI6	State of input 6, Open (0) or closed (1)
DI7	State of input 7, Open (0) or closed (1)
DI8	State of input 8, Open (0) or closed (1)
DO1	State of output 1, Open (0) or closed (1)
DO2	State of output 2, Open (0) or closed (1)
DO3	State of output 3, Open (0) or closed (1)
DO4	State of output 4, Open (0) or closed (1)
CDP1	Pass counter 1 in sense A
CDP2	Pass counter 2 in sense A
CDP3	Pass counter 3 in sense A
CDP4	Pass counter 4 in sense A
CDP5	Pass counter 5 in sense B
CDP6	Pass counter 6 in sense B
CDP7	Pass counter 7 in sense B
CDP8	Pass counter 8 in sense B

Outputs state can be forced using ‘0’ to open the relay and ‘1’ to close the corresponding variable.

2.6 CDU-TCP-ParK

Variable	Description
//Digital Inputs	
“DI1"	Digital Input 1
“DI2"	Digital Input 2
“DI3"	Digital Input 3
“DI4"	Digital Input 4
//Counters	
“C1"	Counter 1
“C2"	Counter 2
“C3"	Counter 3
“C4"	Counter 4
“C1B"	Initial counter 1 value in the period
“C2B"	Initial counter 2 value in the period
“C3B"	Initial counter 3 value in the period
“C4B"	Initial counter 4 value in the period
“C1E"	Final counter 1 value in the period
“C2E"	Final counter 2 value in the period
“C3E"	Final counter 3 value in the period
“C4E"	Final counter 4 value in the period
//Flow rates	
“F1"	Flow Input 1
“F2"	Flow Input 2
“F3"	Flow Input 3
“F4"	Flow Input 4
//Digital Outputs per level	
“DO1"	Level 1 Digital Output
“DO2"	Level 2 Digital Output
“DO3"	Level 3 Digital Output
“DO4"	Level 4 Digital Output
//Digital Outputs per pulse	
“DOP1"	Digital Output 1 Pulse
“DOP2"	Digital Output 2 Pulse
“DOP3"	Digital Output 3 Pulse
“DOP4"	Digital Output 4 Pulse

2.7 MR50-TCP-PARK

Variable	Description
//Digital Inputs	
DIx	Digital input x (x: 1..50)
//Counters	
Cx	Counter x (x: 1..50)
//Flow rates	
Fx	Flow x (x: 1..50)

The MR50 counters can be forced between 0 and the maximum value permitted by the device.
All counter variables can be discriminated.

2.8 MR44-PARK

Variable	Description
//Digital Inputs	
DIx	Digital input (x: 1..4)
//Counters	
Cx	Counter x (x: 1..4)
//Flow rates	
Fx	Flow x (x: 1..4)
//Digital Outputs per level	
DOx	Digital Output x per level (x: 1..4)
//Digital Outputs per pulse	
DOPx	Digital Output Pulse x (x: 1..4)

2.9 MR42-PARK

Variable	Description
//Analogical Inputs	
AIx	Analogical input x (x: 1..4)
//Digital Inputs	
DIx	Digital input (x: 1..4)
//Counters	
Cx	Counter x (x: 1..4)
//Flow rates	
Fx	Flow x (x: 1..4)
//Digital Outputs per level	
DOx	Digital Output x per level (x: 1..4)
//Digital Outputs per pulse	
DOPx	Digital Output Pulse x (x: 1..4)

2.10 PK-ENERGY-KIT - Analizador de redes

Measurement Variables

For a better understanding, the variables are displayed in separate tables depending on the type of variable measured by the devices.

The columns on the left correspond to the type of variable measured: whether the variable is instantaneous, maximum or minimum, the phase to which it corresponds, or any other information depending on the variable, and finally the code used (for example the instantaneous phase-neutral voltage of phase 1 will correspond to the code VI1).

On the other hand, the columns to the right of each table will correspond to devices that can measure each of the variables, indicating with an 'X' if the variable will be measured by the device and with an 'O'.

Voltage

	Phase – neutral voltage											
	Instantaneous				Maximum				Minimums			
	L1	L2	L3	III	L1	L2	L3	III	L1	L2	L3	III
	VI1	VI2	VI3	VI	VMX1	VMX2	VMX3	VMX	VMN1	VMN2	VMN3	VMN
CVM-MINI	X	X	X		X	X	X		X	X	X	

	Phase- phase - voltage

	Instantaneous				Maximum				Minimums			
	L1	L2	L3	III	L1	L2	L3	III	L1	L2	L3	III
	VI12	VI23	VI31	VI123	VMX12	VMX23	VMX31	VMX123	VMN12	VMN23	VMN31	VMN123
CVM-MINI	X	X	X		X	X	X		X	X	X	

	Voltage distortion								
	Instantaneous			Maximum			Minimums		
	L1	L2	L3	L1	L2	L3	L1	L2	L3
	DVI1	DVI2	DVI3	DVMX1	DVMX2	DVMX3	DVMN1	DVMN2	DVMN3
CVM-MINI	X	X	X	X	X	X	X	X	X

Current

	Current											
	Instantaneous				Maximum				Minimum			
	L1	L2	L3	III	L1	L2	L3	III	L1	L2	L3	III
	AI1	AI2	AI3	AI	AMX1	AMX2	AMX3	AMX	AMN1	AMN2	AMN3	AMN
CVM-MINI	X	X	X		X	X	X		X	X	X	

	Neutral current		
	Instantaneous	Maximum	Minimum
	ANI	ANMX	ANMN
CVM-MINI	X	X	X

	Current distortion								
	Instantaneous			Maximum			Minimum		
	L1	L2	L3	L1	L2	L3	L1	L2	L3
	DVI1	DVI2	DVI3	DVMX1	DVMX2	DVMX3	DVMN1	DVMN2	DVMN3
CVM-MINI	X	X	X	X	X	X	X	X	X

Frequency

	Frequency		
	Instantaneous	Maximum	Minimum
	HZI	HZMX	HZMN
CVM-MINI	X	X	X

Power

	Apparent power consumed											
	Instantaneous				Maximum				Minimum			
	L1	L2	L3	III	L1	L2	L3	III	L1	L2	L3	III
	VAI1	VAI2	VAI3	VAI	VAMX1	VAMX2	VAMX3	VAMX	VAMN1	VAMN2	VAMN3	VAMN
CVM-MINI	X	X	X	X	X	X	X	X	X	X	X	X

	Apparent power generated											
	Instantaneous				Maximum				Minimum			
	L1	L2	L3	III	L1	L2	L3	III	L1	L2	L3	III
	NVAI1	NVAI2	NVAI3	NVAI	NVAMX1	NVAMX2	NVAMX3	NVAMX	NVAMN1	NVAMN2	NVAMN3	NVAMN
CVM-MINI	X	X	X	X	X	X	X	X	X	X	X	X

	Active power Consumed											
	Instantaneous				Maximum				Minimum			
	L1	L2	L3	III	L1	L2	L3	III	L1	L2	L3	III
	API1	API2	API3	API	APMX1	APMX2	APMX3	APMX	APMN1	APMN2	APMN3	APMN
CVM-MINI	X	X	X	X	X	X	X	X	X	X	X	X

Active power generated											
Instantaneous				Maximum				Minimum			
L1	L2	L3	III	L1	L2	L3	III	L1	L2	L3	III

	NAPI1	NAPI2	NAPI3	NAPI	NAPMX1	NAPMX2	NAPMX3	NAPMX	NAPMN1	NAPMN2	NAPMN3	NAPMN
CVM-MINI	X	X	X	X	X	X	X	X	X	X	X	X

	Capacitive power consumed											
	Instantaneous				Maximum				Minimum			
	L1	L2	L3	III	L1	L2	L3	III	L1	L2	L3	III
	CPI1	CPI2	CPI3	CPI	CPMX1	CPMX2	CPMX3	CPMX	CPMN1	CPMN2	CPMN3	CPMN
CVM-MINI	X	X	X	X	X	X	X	X	X	X	X	X

	Capacitive power generated											
	Instantaneous				Maximum				Minimum			
	L1	L2	L3	III	L1	L2	L3	III	L1	L2	L3	III
	NCPI1	NCPI2	NCPI3	NCPI	NCPMX1	NCPMX2	NCPMX3	NCPMX	NCPMN1	NCPMN2	NCPMN3	NCPMN
CVM-MINI	X	X	X	X	X	X	X	X	X	X	X	X

	Inductive Power consumed											
	Instantaneous				Maximum				Minimum			
	L1	L2	L3	III	L1	L2	L3	III	L1	L2	L3	III
	IPI1	IPI2	IPI3	IPI	IPMX1	IPMX2	IPMX3	IPMX	IPMN1	IPMN2	IPMN3	IPMN
CVM-MINI	X	X	X	X	X	X	X	X	X	X	X	X

	Inductive power generated											
	Instantaneous				Maximum				Minimum			
	L1	L2	L3	III	L1	L2	L3	III	L1	L2	L3	III
	NIPI1	NIPI2	NIPI3	NIPI	NIPMX1	NIPMX2	NIPMX3	NIPMX	NIPMN1	NIPMN2	NIPMN3	NIPMN
CVM-MINI	X	X	X	X	X	X	X	X	X	X	X	X

	Consumed power factor											
	Instantaneous				Maximum				Minimum			
	L1	L2	L3	III	L1	L2	L3	III	L1	L2	L3	III
	PFI1	PFI2	PFI3	PFI	PFMX1	PFMX2	PFMX3	PFMX	PFMN1	PFMN2	PFMN3	PFMN
CVM-MINI	X	X	X	X	X	X	X	X	X	X	X	X

	Power factor generated											
	Instantaneous				Maximum				Minimum			
	L1	L2	L3	III	L1	L2	L3	III	L1	L2	L3	III
	NPFI1	NPFI2	NPFI3	NPFI	NPFMX1	NPFMX2	NPFMX3	NPFMX	NPFMN1	NPFMN2	NPFMN3	NPFMN
CVM-MINI	X	X	X	X	X	X	X	X	X	X	X	X

	Cos ϕ consumed											
	Instantaneous				Maximum				Minimum			
	L1	L2	L3	III	L1	L2	L3	III	L1	L2	L3	III
	COSI1	COSI2	COSI3	COSI	COSMX1	COSMX2	COSMX3	COSMX	COSMN1	COSMN2	COSMN3	COSMN
CVM-MINI				X				X				X

	Cos ϕ generated											
	Instantaneous				Maximum				Minimum			
	L1	L2	L3	III	L1	L2	L3	III	L1	L2	L3	III
	NCOSI1	NCOSI2	NCOSI3	NCOSI	NCOSMX1	NCOSMX2	NCOSMX3	NCOSMX	NCOSMN1	NCOSMN2	NCOSMN3	NCOSMN
CVM-MINI				X				X				X

Energies

	Apparent energy			
	Consumed		Generated	
	III	Tariff y (y:1..9)	III	Tariff y (y:1..9)
	VAE	VAETy	NVAE	NVAETy
CVM-MINI	X		X	

Active energy Consumed

	L1	L2	L3	III	Partial	Tariff y		Contract x (x:1..3)	
						y:1..3	y:4..9	Tariff y (y:1..9)	Total
	AE1	AE2	AE3	AE	PAE	AETy	AETy	AECxTy	AECxTOT
CVM-MINI				X					

	Active energy generated								
	L1	L2	L3	III	Partial	Tariff		Contract x (x:1..3)	
						y:1..3	y:4..9	Tariff y (y:1..9)	Total
	NAE1	NAE2	NAE3	NAE	NPAE	NAETy	NAETy	NAECxTy	NAECxTOT
CVM-MINI				X					

	Capacitive energy consumed									
	III	Partial	Tariff		2nd quadrant (2Q)					
					L1	L2	L3	III	Contract x (x:1..3)	
			y:1..3	y:4..9					Tariff y (y:1..9)	Total
	CE	PCE	CETy	CETy	CE2Q1	CE2Q2	CE2Q3	CE2Q	CE2QCxTy	CE2QCxTOT
CVM-MINI	X									

	Capacitive energy generated									
	III	Partial	Tariff		4nd quadrant (2Q)					
					L1	L2	L3	III	Contract x (x:1..3)	
			y:1..3	y:4..9					Tariff y (y:1..9)	Total
	NCE	PNCE	NCETy	NCETy	CE4Q1	CE4Q2	CE4Q3	CE4Q	CE4QCxTy	CE4QCxTOT
CVM-MINI	X									

	Inductive energy consumed									
	III	Partial	Tariff		1st quadrant (1Q)					
					L1	L2	L3	III	Contract x (x:1..3)	
			y:1..3	y:4..9					Tariff y (y:1..9)	Total
	IE	PIE	IETy	IETy	IE1Q1	IE1Q2	IE1Q3	IE1Q	IE1QCxTy	IE1QCxTOT
CVM-MINI	X									

	Inductive energy generated									
	III	Partial	Tariff		3rd quadrant (3Q)					
					L1	L2	L3	III	Contract x (x:1..3)	
			y:1..3	y:4..9					Tariff y (y:1..9)	Total
	NIE	PNIE	NIETy	NIETy	IE3Q1	IE3Q2	IE3Q3	IE3Q	IE3QCxTy	IE3QCxTOT
CVM-MINI	X									

All energy variables can be discriminated.

Maximal demand

	Maximum demand consumed									
	Instantaneous					Maximum				
	L1	L2	L3	III	Tariff y (y:1..3)	L1	L2	L3	III	Tariff y (y:1..3)
	MDI1	MDI2	MDI3	MDI	MDITy	MDMX1	MDMX2	MDMX3	MDMX	MDMXTy
CVM-MINI	O	O	O	O		O	O	O	O	

All maximal demand variables can be discriminated.

Harmonics

	Voltage harmonics							
	L1		L2		L3		Neutral	
	x:1..15	x:16..50	x:1..15	x:16..50	x:1..15	x:16..50	x:1..15	x:16..50
	ARMxV1		ARMxV2		ARMxV3		ARMxVN	
CVM-MINI	X		X		X			

	Current harmonics											
	L1			L2			L3			Neutral		
	x:1..15	x:16..31	x:32..50	x:1..15	x:16..31	x:32..50	x:1..15	x:16..31	x:32..50	x:1..15	x:16..31	x:32..50
	ARMxA1			ARMxA2			ARMxA3			ARMxAN		
CVM-MINI	X			X			X					

Inputs and outputs

	Analogue inputs											
	Instantaneous				Maximum				Minimum			
	x:1..1	x:2..3	x:4..8	x:9..24	x:1..1	x:2..3	x:4..8	x:9..24	x:1..1	x:2..3	x:4..8	x:9..24
	AIIx				AIMXx				AIMNx			
CVM-MINI	X				X				X			

The analogue input of the CVM-MINI devices is the temperature variable measured in the devices types.

	Digital outputs:						
	1	2	3	4	x:5..18	x:19..20	x:21..24
	DO1	DO2	DO3	DO4	DOx		
CVM-MINI	X	X					

Digital outputs can be forced to 0 (Open output) or to 1 (closed output).

Graphical variables and special tables

Special variables for graphs and tables serve to indicate to CirPark Scada that we want to make a graph or non-standard table. On a non-standard graph or table the values that appear in the X and Y axis, rows and columns on a table, may switch from one type to another, depending on the variables represented. For a better understanding of these graphs and tables see the section on the java client.

When you want to make special graphs or tables, such as the waveform or harmonics, it will be necessary to use the following variables.

	Harmonics							
	Voltage:				Current			
	L1	L2	L3	Neutral	L1	L2	L3	Neutral
	ARMV1	ARMV2	ARMV3	ARMVN	ARMA1	ARMA2	ARMA3	ARMVN
CVM-MINI	X	X	X		X	X	X	

3 Devices state

Sometimes it will be necessary to know the status of a device, for example when you want to activate an event when a device stops communicating. To ascertain the status we use the STATUS variable.

name. STATUS

The following are the possible values that this status variable may have.

Status of the device	*Value*
The device is working properly.	1
Device not initialized, initial communication with the device has not been established.	4
The communication port is incorrect or nonexistent.	18
Communication error with the device is produced.	34
The type of device you are trying to communicate with is different from the specified device.	66
The device communicates correctly, but it detects that some of the phases are badly connected.	130
The version of the device is incompatible with the software.	258
The SD Memory Card is invalid, write-protected or not present.	514

4 Parking counters

The parking counters will be coded for use in formulas and expressions such as

R$PKC_nombreContador.variable

Using the R$PKC prefix the program will ascertain that this is a parking counter rather than a device variable.

The following table shows the different variables associated with each counter.

Variable	**Description**
EMPTY	Number of free spaces in normal mode
FULL	Number of occupied spaces in normal mode
STOLEN	Spaces in stolen status
OCCUPIED	Occupied spaces
ERROR	Number of spaces in communication error in normal mode
UNKNOWN	Number of spaces where the normal mode is unknown
EMPTYRESERVED	Number of free spaces in reserved o prohibited mode
STOLENRESERVED	Reserved spaces in stolen status
OCCUPIEDRESERVED	Occupied reserved spaces
ERRORRESERVED	Number of spaced in communications error reserved or prohibited
UNKNOWNRESERVED	Number of spaces where the reserved or prohibited mode in unknown
FULLRESERVED	Number of occupied spaces in reserved or prohibited mode
//totals	
EMPTYTOTAL	Total number of free spaces
FULLTOTAL	Total number of occupied spaces
STOLENTOTAL	Total spaces in stolen status
OCCUPIEDTOTAL	Total occupied spaces (for maintaining a logic state)
ERRORTOTAL	Total number of spaces in communications error
UNKNOWNTOTAL	Total number of unknown state spaces
TOTAL	Total spaces number
// historical variables	
EMPTYPERCENT	Percentage of free spaces in normal mode

FULLPERCENT	Percentage of occupied spaces in normal mode
STOLENPERCENT	Stolen spaces
OCCUPIEDPERCENT	Occupied spaces (for maintaining a logic state)
ERRORPERCENT	Percentage of spaces in communication error in normal mode
UNKNOWNPERCENT	Percentage where state in unknown in normal mode
EMPTYRESERVEDPERCENT	Percentage of free spaces in reserved or prohibited mode
FULLRESERVEDPERCENT	Percentage of occupied spaces in reserved or prohibited mode
STOLENRESERVEDPERCENT	Stolen reserved spaces
OCCUPIEDRESERVEDPERCENT	Occupied reserved spaces (for maintaining a logic state)
ERRORRESERVEDPERCENT	Percentage of spaces in communication error in reserved or prohibited mode
UNKNOWNRESERVEDPERCENT	Percentage of spaces where the state is unknown in reserved or prohibited mode
EMPTYTOTALPERCENT	Percentage of total free spaces
FULLTOTALPERCENT	Percentage of total occupied spaces
STOLENTOTALPERCENT	Total stolen spaces
OCCUPIEDTOTALPERCENT	Total occupied spaces (for maintaining a logic state)
ERRORTOTALPERCENT	Percentage of total spaces in communication error
UNKNOWNTOTALPERCENT	Percentage of total spaces where state is unknown
//Transit zone counter	
CROSSINGZONE	Value of the actual pass zone
BRIGHT	Allow forcing the brightness of the sensors belonging to the counter.
TYPE	Enables forcing the mode of the sensors comprising the counter
CONF_STATE	Enables forcing the configured state mode of the sensors comprising the counter
MOVEMENTS	Returns the historic with the number of counter occupancies in the period
OCCUPANCY	Returns the historic with the percentage of occupancy time of the counter in the period.
MODE*	Sensor Mode: - Normal (green): 0 - Reserved (Orange): 1 - Prohibited (Red): 2
HANDICAPPED*	Reserved bay configured with handicapped bay type

5 Events variables

The event variables will be coded for use in formulas and expressions such as

R$EVE_event.variable

Using the R$EVE prefix the program will ascertain that this is an event variable rather than a device variable.

The following is a table with the different variables associated with each event

Identifier	*Description*
ST	State of the event (0 off, 1: Enabled)
TACT	Time the event has been active, the value is 0 when the event is disabled
CACT [1]	Counter of the number of activations
CDIS [1]	Counter of the number of deactivations
CACK [1] [2]	Counter of the number of acknowledgments
CTIM [1]	% Of the time of the consultation period that the incident is activated

[1] Only if the event is recorded on file

[2] Only if the incident is reported

6 Calculated Variables

The calculated variables will be coded for use in formulas and expressions such as

R$CAL_GrupoVariablesCalculadas.VariableCalculada

Using the R$CAL prefix the program will ascertain that this is a calculated variable rather than a device variable.

7 XML Services

CIRPARK SCADA provides a range of XML services to enable, in some respects, communication with other applications.

In petitions where it is necessary to express a date and time, both in service petitions and the data from the response, these will be represented in UTC (Universal Coordinated Time) with the format DDMMAAAAHHMMSS (two digits for the day, two for the month, four for the year and two for the hour, minutes and seconds. It will also be possible represent only a date as DDMMAAAA assuming that the time is 00:00:00, or represent an hour as HHMMSS. Finally in cases where milliseconds are required these are represented with three digits after the seconds, DDMMAAAAHHMMSSUUU or HHMMSSUUU.

The petitions must follow the URI standard (RFC 2396) so that the user of these petitions has to take into account this detail when making such calls (especially in cases where the name of any device contains non-ASCII characters).

7.1 /services/user/devices.xml

Returns the list of configured devices.

```
<devices>
        <id> … </id>
        …
</devices>
```

Where:

- **devices:** Main field which will identify the main XML as a response to the devices list request.
- **id**: Name of each one of the devices.

An extra parameter can be included to list virtual devices as well:
/services/user/decvices.xml?info=ALL

7.2 /services/user/deviceInfo.xml?id=device&...

Returns information on devices. Each of the devices on which information is required must be included in the petition as:

?id=device1&id=device2

```
<devices>
        <device>
                <id> … </id>
                <description>… </ description>
                <type>… </ type>
                <typeDescription>… </ typeDescription>
                <var>… </ var>
                …
        </device>
        …
</devices>
```

Where:

- ***devices:*** Main Field which identified the XML as a response to the information request from the devices.
- ***device:*** Information from each of the devices requested:
    - ***id:*** Name of the device.
    - ***description:*** Description of device
    - ***type:*** Type of device (for example SP3)
    - ***typeDescription:*** A description of the type of device (for example: SP3)
    - ***var:*** Name each of the variables of the device. The name is expressed as device.variable.

7.3 /services/user/varInfo.xml?var=device.variable&...&id=device& ...

Returns variable information when carrying out the XML request. Each of the variables that the value is required from should be included in the petition as:

?var=device.variable

And if you want to get information from all the variables of a device this must be indicated as

&id=device

With it being possible to request information from one or more variables and one or more devices in the same request.

```
<varInfo>
        <var>
                <id> … </id>
                <title>… </ title>
                <hasValue> T </ hasValue>
                <hasLogger> T </ hasLogger>
                <sampleMode>… </ sampleMode>
                <measureUnits>… </ measureUnits>
                <unitsFactor>… </ unitsFactor>
                <decimals>… </ decimals>
        </ var>
        …
<varInfo>
```

Where:

- ***varInfo:*** Main field which identifies the XML as a response to the requested info. about variables.
- ***var:*** Information from each of the variables requested:
  - ***id:*** Variable name in device.variable format (see 0
  - Variables)
  - ***title:*** Brief description of the variable.
  - ***hasValue:*** Indicates if it possible to ask the instantaneous value of the variable (T) or not (F).
  - ***hasLogger:*** Indicates whether it is possible to ask the log value of the variable (T) or not (F).
  - ***sampleMode:*** Variable type, mode used to group together the values of a variable:
    - *none:* Without type
    - *average:* Average value:
    - *max:* Maximum value.
    - *min:* Maximum value.
    - *pfAverage:* Power factor, average value
    - *pfMax:* Power factor, maximum value
    - *pfMin:* Power factor, minimum value
    - *last:* Last value:
    - *differential:* Differential Value between the current value and the previous one.
    - *samples:* samples. The value can not be grouped
    - *discrete:* Discreet values. The value can not be grouped
  - ***measureUnits:*** Variable units:
    - #NONE → without units
    - #V → Voltage
    - #A → Current
    - #VA → Apparent power
    - #W → Active power
    - #VARL → Inductive power
    - #VARC → Capacitive power
    - #PF → Power factor
    - #HZ → Frequency
    - #PERCENT → Percentage
    - #WH → Active energy
    - #VARLH → Inductive energy
    - #VARCH → Capacitive energy
    - #DATETIME → Date and time
    - If not preceded by an # this is a unit defined by the user
  - ***unitsFactor:*** Power of 10 that indicates the value the variable is multiplied by the variable in the log file.
  - ***decimals:*** Decimals with this variable.

7.4 /services/user/values.xml?var=device.variable&...&id=device&..

Returns the instantaneous value of the variable when the XML request is carried out. Each of the variables that the value is required from should be included in the petition as:

?var=device.variable

If you want to ascertain the value of all the variables of a device it should be indicated as:

&id=device

With it being possible in a single request to request the value of one or more variable and values of one or more devices

```
<values>
        <variable>
                <id> … </id>
                <value>… </ value>
        </ variable>
        …
</values>
```

Where:

- ***values:*** Main field which will identify the XML as a response to the request from variable values
- ***variable:*** List of variables:
  - ***id:*** Identifier of the variable in the format device.variable (see 0
  - Variables)
  - ***value:*** Value of variable at the time of the request.

7.5 /services/user/forceVariables.xml?id=device

With this petition we may send the order to force variables to CIRPARK SCADA. The request must include the name of the device we want to force so that, if necessary, authentication can be checked. Only the variables belonging to the device indicated in the petition will be forced

```
<forceVariables>
        <forceVar>
                <forceName>… </ forceName>
                <forceValue>… </ forceValue>
        </ forceVar>
        …
</forceVariables>
```

Where:

- ***forceVariables:*** Main field that will identify the XML as a petition to force variables.
- ***forceVar:*** Information on each of the variables to be forced:
  - ***forceName:*** Name of variable with format device.variable (0 Measurement Variables). Only variables that can be forced variables, for example digital output variables.
  - ***forceValue:*** Value at which we wish to force the variable.

7.6 /services/user/records.xml?begin=…&end=…&var=…&period=900

Returns information on one or more variables between the dates *"begin"* and *"end"*. Each of the variables for which information is to be obtained must be included in the request as:

?var=device.variable

The format of *"begin"* and *"end"* will be DDMMAAAA when you wish only to indicate the date (in this case the time will 00:00:00) or DDMMAAAAHHMMSS when both the date and the hour are specified. Both "begin" as "end" must be expressed in UTC (Universal Coordinated Time).

Finally, we may specify the period of data grouping using the "*period"* parameter.
This value may be:

FILE	→Data not grouped, returning the register as they have saved in the log.
AUTO	→Grouping will take place automatically depending on the specified dates *"begin"* and "end"
ALL	→Data is grouped into a single value
> 0	→Value in seconds in which the data is grouped.

If the "*period"* parameter does not appear on the petition it shall be considered as value 0 and the data will not be grouped data.

```
<recordGroup>
        <period>… </ period>
        <record>
                <dateTime>… </ datetime>
                <field>… </ field>
                …
        </record>
        …
</recordGroup>
```

Where:

- ***recordGroup:*** Main field which will identify the XML as a response to the requested variables. ***period:*** Register period. Will report on time elapsed between records.
- ***record:*** Will Identify each of the records:
    - ***dateTime:*** Date and time of the sample.
    - ***field:*** Standard value record.
    - ***fieldARM:*** Harmonic value record

Here are the different types of values that can be returned by this request:

- Standard value record (voltages, currents, power, energy, etc.).

```
<field>
        <id> … </id>
        <value>… </ value>
</ Field>
```

- ***id:*** Variable identifier (device.variable)
- ***value:*** Value

- Harmonic value record

```
<fieldARM>
        <id> … </id>
        <element>
                <harmonic>… </ harmonic>
                <value>… </ value>
        </ element>
        …
</fieldARM>
```

- ***id:*** Variable identifier (device.variable)

- ***Element:*** Reports from each of the harmonics
  - ***harmonic:*** Number of harmonic
  - ***value:*** Harmonic value.

7.7 /services/ user/eventsExtended.xml

Returns information on one or more events identified by id or by group, between the dates *"begin"* and *"end" and with the flag Totalize*.

Permited parameters:

- **id:** event identifier
- **group:** group of events identifier
- **begin:** date of the begin of the period
- **end:** date of the end of the period
- **totalize:** flag to indicate if it is totalized or not (T o F)

❖ Request examples:

/services/user/eventsExtended.xml?id=S1&begin=21052013000000&end=21052013230000

/services/user/eventsExtended.xml?id=S1&begin=21052013000000&end=21052013230000&totalize=T

/services/user/eventsExtended.xml?group=GRUP1&begin=21052013000000&end=21052013230000

❖ Respons examples (without totalize -> totalize=F):

```
<recordEventsGroup>
        <localEvents>                         (List of Local Events)
                <record>
                        <date>20052013063931</date>                                  (Date of Event)
                        <event>
                                <id>S1</id>                                        (Identifier Event)
                                <dateAck>20052013064708</dateAck>        (Acknownledge Date of Event)
                                <dateDisabled>20052013064729</dateDisabled>   (Disabled Date of Event)
                        </event>
                </record>
        </localEvents>
        <remoteEvents>                                                         (List of remote Events)
                <record>
                        <date>20052013063932</date>                                  (Date of Event)
                        <event>
                                <id>REMOTE_ENGINE.S1</id>                          (Event Identifier)
                                <dateAck>20052013064708</dateAck>        (Acknownledge Date of Event)
                                <dateDisabled>20052013064729</dateDisabled>   (Disabled Date of Event)
                        </event>
                </record>
        </remoteEvents>
</recordEventsGroup>
```

❖ Respons examples (totalize -> totalize=T):

```
<recordEventsGroup>
        <localEvents>                         (List of local events)
                <record>
                        <id>S1</id>                                            (Identifier of event)
                        <activated>1</activated>               (Number of times of event activation)
                        <acknowledge>1</acknowledge>            (Number of reconnaissance answer)
                        <disabled>1</disabled>                          (Number of disabilitations)
                        <duration>478</duration>                                    (time of whole event)
                </record>
        </localEvents>
        <remoteEvents></remoteEvents >
</recordEventsGroup>
```

7.8 /services/parking/findYourCarSensor.json

Returns sensors occupied on a defined time period

Request:

/services/parking/findYourCarSensor.json

Permited parameters:

- **begin** (mandatory). Starting date of search.
- **end** (Optional). Ending date of search. If you not specify any date will be considered 5 minutes after starting date.
- **counter** (Optional). Search of occupied sensors will be done only on specified counter sensors. Will be possible to parameterize more than one counter (&counter=…&counter=…).

  If any counter is specified, the search will be done over all of parking sensors.

Respons example:

```
{
        "findYourCar": {
                "sensors": [
                        { "name": "SONE-0101" },
                        {"name": "SP1"}
                ]
        }
}
```

Where:

- sensors: List of found sensors.
- name: name of sensors

7.9 /services/parking/findYourCarMap.json

Returns background of the screen where the sensors are positioned and plotted in it.

Request:

/services/parking/findYourCarMap.json

Permited parameters:

- **id** (optional). Sensor identifier that want to be searched. Will be possible to parameterize more than one sensor for each request (&id=…&id=…)
- **counter** (Optional).). Search of sensors will be done only for sensors of the counter. Will be possible to parameterize more than one counter (&counter=…&counter=…).

Respons example:

```
{
"findYourCar": {
        "parking": [
        {
                "scada": { "id": "PARKING-FLOOR-1", "width": "211.670000", "height": "199.760000"},
                "backgroundImage": {
                        "image64": "iVBORw0KGgoAAAANS...QAAAABJRU5ErkJggg==",
                        "horizontalAlign": "CENTER",
                        "verticalAlign": "CENTER",
                        "maintainAspectRatio": "T",
                        "maintainRealSize": "T"
                },
                "places": [
                        {
                                "sensorName": "SP1", "x": "136.190000", "y": "23.290000", "width": "34.820000",
                                "height": "25.650000"
                        }
                        ]
        }
        ]
}
```

8 Service support

In the event that operation or damage, call for technical service of **CIRCONTROL, S.A.**

CIRCONTROL, S.A. – After sales service
Innovació, 3 (Industrial Park Can Mitjans)
08232 Viladecavalls
Tel: (+34) 93 736 29 40
Fax: (+34) 93 736 29 41
Web: www.circontrol.com
E-mail: ps-support@circontrol.com