



COUNTER 2 Bytes

ACTinBOX Logical Functions



Edition 1

INDEX


1. Introduction.....	3
2. Implementation.....	4
2.1. Definition.....	4
2.2. Devices	5
2.3. Programming	6
2.3.1. Configuration Function 1	7
2.3.2. Configuration Function 2	10
2.3.3. Topology.....	13
2.3.4. Group Addresses	14
3. Application result.....	15
4. Notes	16
5. Anex I. Memory Values Format.....	17

1. INTRODUCTION

The goal of this document is to make easier the understanding and knowledge acquiring of the functionalities that the integrated logical functions modules in the following Zennio products offer:

 ACTinBOX QUATRO (Ref. ZN1IO AB40)

 ACTinBOX Classic (Ref. ZN1IO AB46)

 ACTinBOX MAX6 (Ref. ZN1IO AB60)

These products are mainly oriented to control loads through relays, but they provide other functionalities that increase the product utility.

The logical functions modules integrated in these products allow the building automation installer to perform advanced applications for controlling the installations, without the need of new modules that could exceed the reasonably number of electrical devices.

The solution explained in the current document is part of the great range of possibilities that Zennio ACTinBOX logical functions offer, being an example of solutions adapted to every need.

It is worth highlighting that these logical functions modules are independent of the main functions of the actuators in which they are placed. However, they can be used in both, those internal main functions and external functions of other devices, providing an extra versatility and great utility.

2. IMPLEMENTATION

The implemented Counter function is defined below, using an example as a basis for the explanation of its performance (the COUNTER.PR4 file for the ETS and other useful files can be downloaded from the Zennio website).

2.1. DEFINITION

The COUNTER function deal with the counting of bits “1” that a sensor detects. As a result, a telegram with the counter value is transmitted to the bus. Through another sensor, a reset of the counter is performed.


- **n1**: internal variable where the counter value is stored.
- **Sensor 1**: this sensor is the source of bits “1” that are desired to be counted.
- **Sensor 2**: this sensor is the one that resets the counter when detecting a bit “1”.

SOURCE	DATA ENTRY (BIT)	OPERATION	OUT (BYTE)
SENSOR 1	“1”	INCREASE ($n1 = n1 + 1$)	n1
SENSOR 1	“0”	NO OPERATION	-
SENSOR 2	“1”	RESET ($n1 = 0$)	n1
SENSOR 2	“0”	-	-

Moreover, the parameters have been programmed so that the output telegram is delayed 1 second since the operations are performed. There are multiple configurable parameters that allow this solution to be adapted for every installation need.

2.2. DEVICES

Among the three Zennio devices that offer the logical functions modules to integrators, we will use the following one for the current example:

 ACTinBOX Classic (Ref. ZN1IO AB46). (**¡Error! No se encuentra el origen de la referencia.**)

As Sensor 1 and Sensor 2 simulator, we will use the following Zennio device:


 InZennio Z38 (Ref. ZN1VI TP38). (**¡Error! No se encuentra el origen de la referencia.**)



Figure 1 ACTinBOX Classic (Actuator 4 outputs/6 inputs)



Figure 2 InZennio Z38 (Multifunction touch panel)

2.3. PROGRAMMING

The block diagrams that explain the programming logic and the configuration of the logical functions module of ACTinBOX Classic are shown in the figures below.

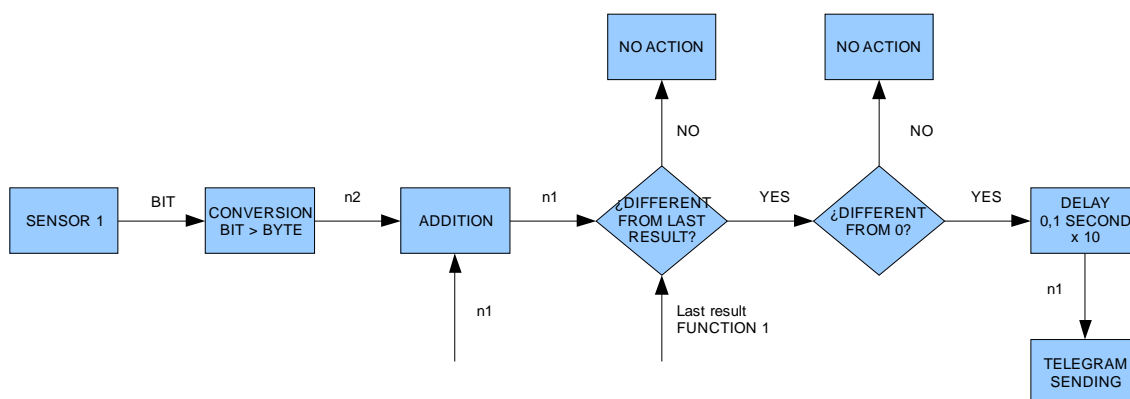


Figure 3 Programming logic for SENSOR 1

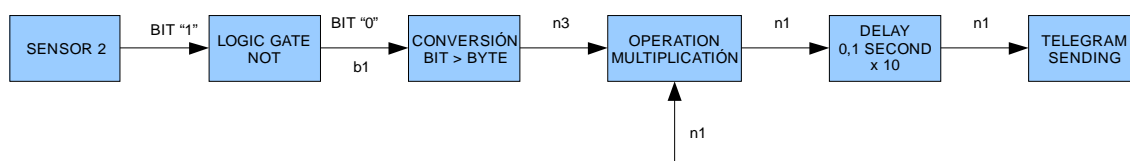


Figure 4 Programming logic for SENSOR 2

In the “LOGICAL FUNCTIONS” section of parameterization window, both the number of modules or logical functions used to implement this solution and the number of data entry objects are selected. The number of communication objects in the ETS Topology window depends on this parameterization.

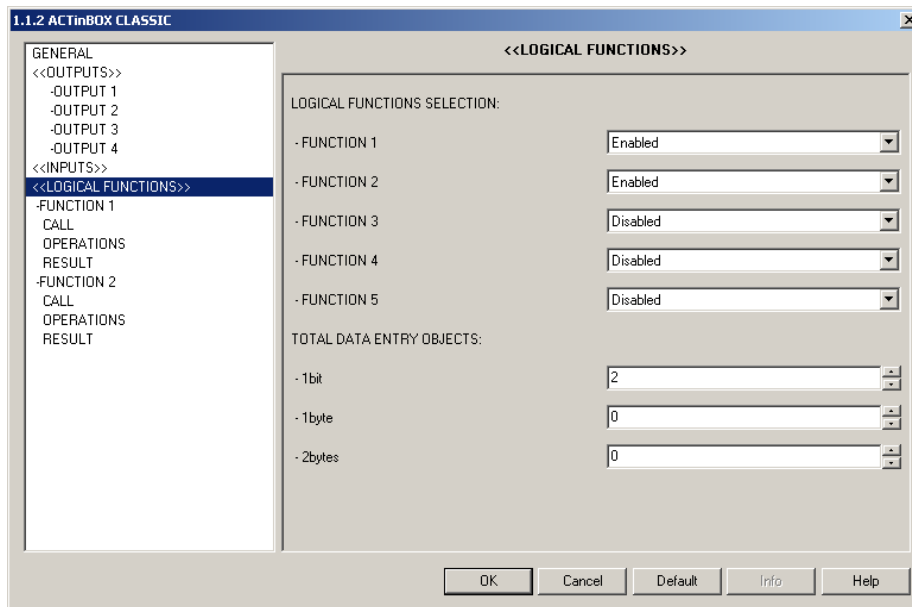


Figure 5 Logical Functions page

- Function 1: **enabled**
- Function 2: **enabled**
- Number data entry objects of 1bit: 2 (**one bit for each sensor**)

2.3.1. CONFIGURATION FUNCTION 1

Call

- CALL – Object 1: (LF) (1 bit) Data Entry 1. (The call to Function 1 is executed when SENSOR 1 transmits a telegram).

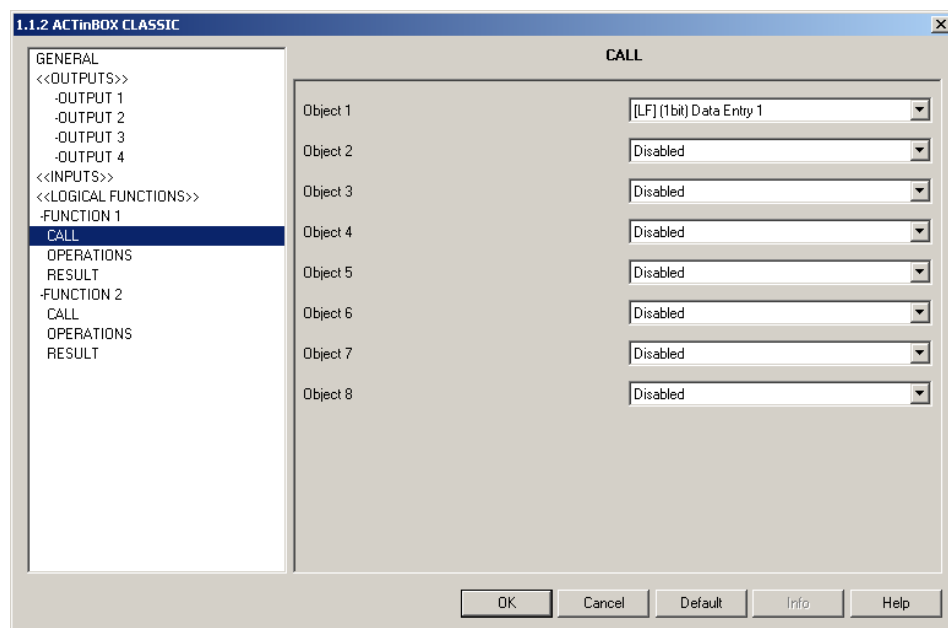


Figure 6 Call page

Operations

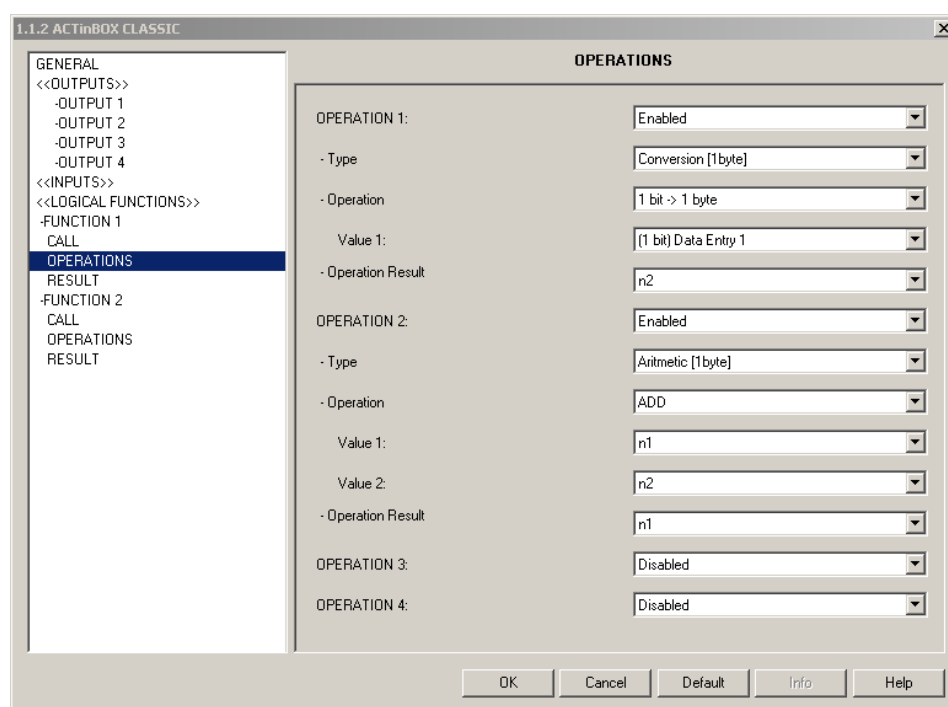


Figure 7 Operation page

- OPERATION 1: **The Data Entry 1 (1bit) is converted to a 1-byte data entry** and the operation **result** is stored in a variable called **n2**.
- OPERATION 2: **Addition of variable n1 and variable n2**

Result

Figure 8 Result Page

- TYPE of output data: **1 byte**
- VALUE, variable that is desired to be sent: **n1**
- SENDING, allows specifying when the value is sent: **Result is different from last sent**
- RESTRICTION, allows setting the restrictions in sending: **Do not send the result when value is zero**
- DELAY: **10 tenths of a second**
- INTERNAL LINKS: **No**

2.3.2. CONFIGURATION FUNCTION 2

Call

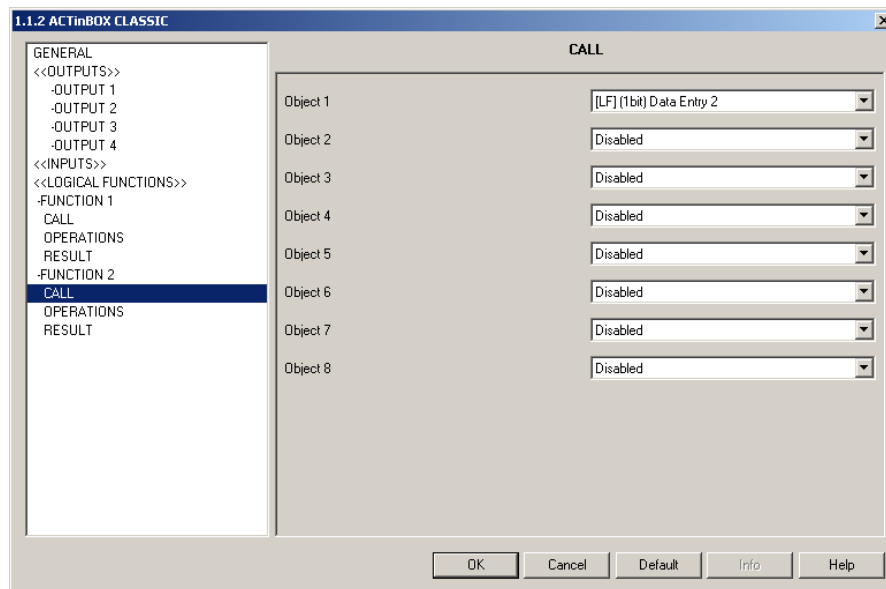


Figure 9 Call Page 2

- CALL – Object 1: (LF) (1 bit) Data Entry 2

Operations

- OPERATION 1: **NOT operation** over the **Data Entry 2** provided by the RESET Sensor (when sending a bit “1”). It is stored in the variable **b1**
- OPERATION 2: **Conversion of 1-bit variable b1 in 1-byte data type, which is stored in variable n3.**
- OPERATION 3: **Multiply** variable **n3** by variable **n1**, which is modified by FUNCTION 1.

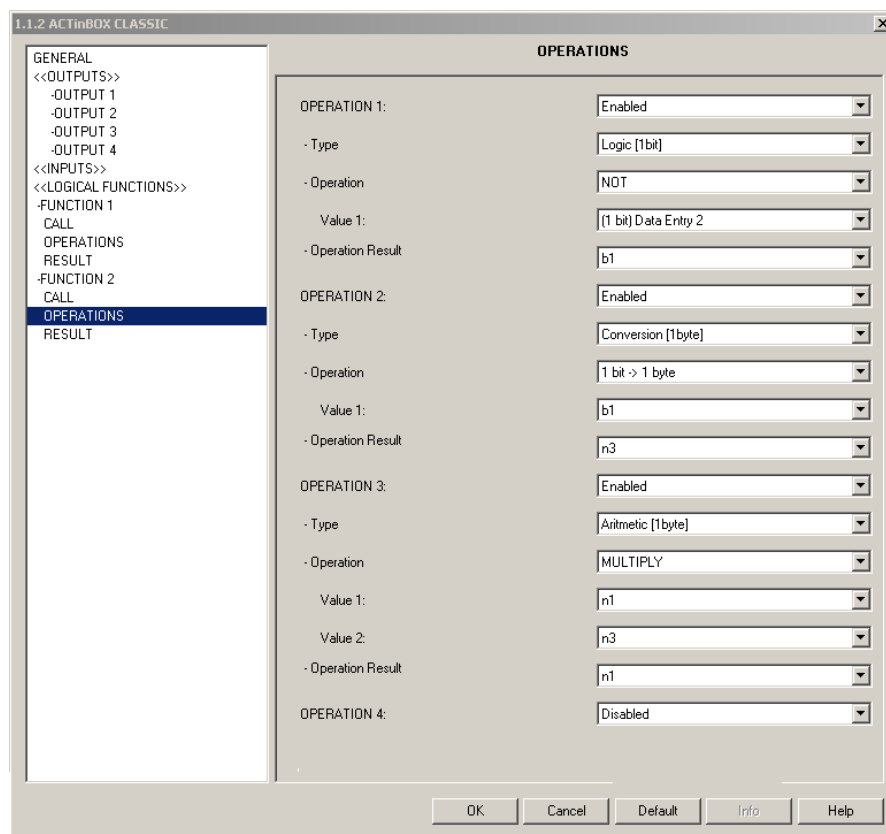


Figure 10 Operations Page 2

Result

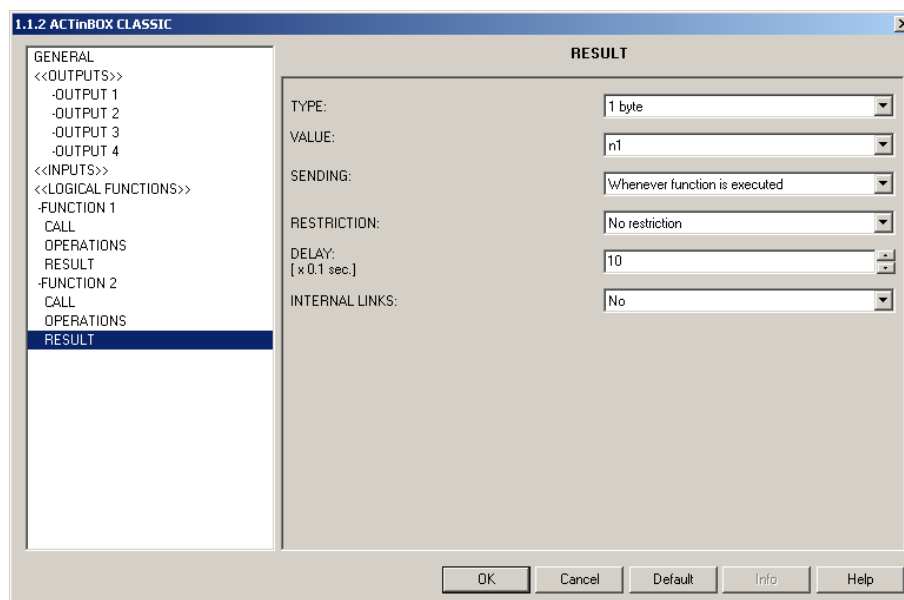


Figure 11 Result Page 2

- TYPE of the output data: **1 byte.**
- VALUE, variable to send: **n1**
- SENDING, allows specifying when the value is sent: **Whenever the function is executed.**
- RESTRICTION, allows setting the restrictions in sending: **No restriction.**
- DELAY: **10 tenths of a second.**
- INTERNAL LINKS: **No.**

2.3.3. TOPOLOGY

Next, the ETS Topology window of the programmed performance is shown:

Number	Name	Object Function	Description	Group Addresses	Length	C	R	W	T	U	Data Type	Priority
10	Time	Current Time			3 Byte	C	-	W	T	-		Low
11	Date	Current Date			3 Byte	C	-	W	T	-		Low
12	Screens	One only object used by device			1 Byte	C	-	-	T	-		Low
13	Temperature	Internal sensor value			2 Byte	C	R	-	T	-		Low
14	Timer (Channel 1)	Linked to Timer Pages			1 bit	C	-	-	T	-		Low
15	Timer (Channel 2)	Linked to Timer Pages			1 bit	C	-	-	T	-		Low
16	Timer (Channel 3)	Linked to Timer Pages			1 bit	C	-	-	T	-		Low
17	Timer (Channel 4)	Linked to Timer Pages			1 bit	C	-	-	T	-		Low
18	[Home; Box 1] Light On/Off	0=Off; 1=On		0/0/1	1 bit	C	-	-	T	-		Low
19	[Home; Box 2] Binary Control	1 bit generic control		0/0/2	1 bit	C	-	-	T	-		Low
174	Touch Block	1=Touch Disabled; 0=Touch Free			1 bit	C	-	W	-	-		Low

Figure 12 View of the communication objects of the device 1.1.1 InZennio Z38

Number	Name	Object Function	Description	Group Addresses	Length	C	R	W	T	U	Data Type	Priority
101	[O1] ON/OFF	N.O. (0=Open Relay; 1=Close)		0/0/1	1 bit	C	-	W	-	-		Low
102	[U1] (1 bit) Data Entry 1	Binary Data Entry (0/1)		0/0/1	1 bit	C	-	W	-	-		Low
103	[U1] (1 bit) Data Entry 2	Binary Data Entry (0/1)		0/0/2	1 bit	C	-	W	-	-		Low
139	[U1] Function 1 RESULT (1 byte)	FUNCTION 1 Result		0/0/3	1 Byte	C	R	-	T	-		Low
140	[U1] Function 2 RESULT (1 byte)	FUNCTION 2 Result		0/0/3	1 Byte	C	R	-	T	-		Low
149	Reset 0	Voltage Recovery->Sending of 0			1 bit	C	-	-	T	-		Low
150	Reset 1	Voltage Recovery->Sending of 1			1 bit	C	-	-	T	-		Low

Figure 13 View of the communication objects of the device 1.1.2 ACTinBOX Classic

DEVICE	INDIVIDUAL ADDRESS
InZennio Z38	1.1.1
ACTin BOX Classic	1.1.2

2.3.4. GROUP ADDRESSES

ADDRES S	NAME	OBJEC T	DEVICE	DESCRIPTION
0/0/1	INCREASE COUNTER	8	1.1.1	It increases the counter value in one each time a bit “1” is detected by the sensor 1.
		0	1.1.2	
		102	1.1.2	
0/0/2	RESET COUNTER	10	1.1.1	It resets the counter each time a bit “1” is detected by the sensor 2.
		103	1.1.2	
0/0/3	COUNTER RESULT	139	1.1.2	It sends a telegram to the bus with the counter value when it changes.
		140	1.1.2	

Figure 14 Group Address 1. Increase Counter

Figure 15 Group Address 2. Reset Counter

Figure 16 Group Address 3.Counter Result

3. APPLICATION RESULT

The application result is a counter useful for implementing more complex functionalities or for putting into practice certain events.

An application example is the quantification of power consumption (KWh) or water (m^3) per building or house. This application could be implemented installing a KNX installation with counters capable of emitting a small electric signal every time the consumption, measured by the counter itself, is increased in a prederterminated unit. For this, using a 2-byte counter is very useful as it allows counting to 65535 events or pulses.

Within the download file, there is another file with .TRX extension which corresponds with the bus reading for the explained example. This file simulates a series of events in the sensor input that allows understand the behavior of the counter implemented in this document. For looking at the mentioned file, open the groups monitor in ETS (**¡Error! No se encuentra el origen de la referencia.**), and then, proceed to charge the telegrams from a file (Figure 18).



Figure 17 Groups Monitor Icon

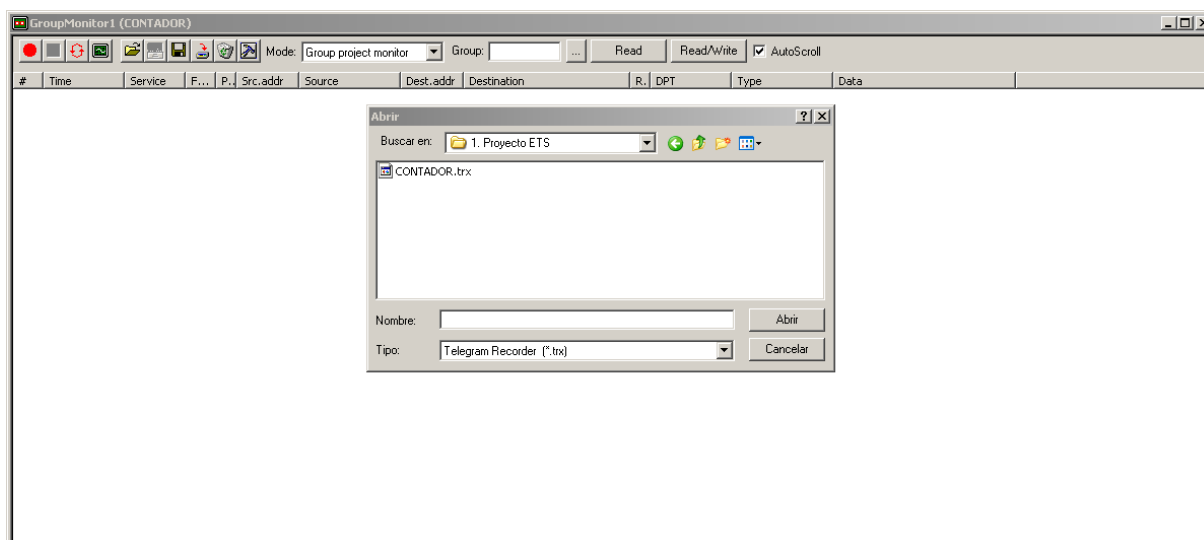


Figure 18 Charging telegrams from file

4. NOTES

In the following lines, there is a series of notes that allows the domotic systems integrator to deeply understand the operation of the device ACTinBOX Classic.

- The logical functions modules belonging to Zennio devices are independent of the physical inputs and outputs of the devices, being capable of operating only with the telegrams received in the bus from other devices.
- The direct operation with the inputs and outputs of the actuator itself is carried out by means of internal links (virtual group addresses) which are not transmitted to the bus, avoiding the actuator overload.
- The internal variables (b, n, x) are memory spaces where it is possible to save values in different formats (See Annex I. Memory Values Format).
- The internal variables (b, n, x) are the same within the same functions. In other words, the variable n1 of Function1 is the same variable n1 of Function 2. This way, it is possible to use a value calculated within a function in other functions.
- The internal variables (b, n, x) remain in the device memory after KNX bus voltage drops or the voluntary disconnection of the device.
- The call values in a function do not have necessarily to act as elements within an operation of the function. In this case, they act only as triggers of the function.

5. ANNEX I. MEMORY VALUES FORMAT



VALUE	FORMAT
CALL	1 BIT
	1 BYTE
	2 BYTES
OPERATIONS	1 BIT
	1 BYTE
	2 -BYTE UNSIGNED INTEGER
	2-BYTE FLOAT
RESULT	1 BIT
	1 BYTE
	2-BYTE UNSIGNED INTEGER
	2-BYTE FLOAT



BECOME USER!

<http://zennioenglish.zendesk.com>

TECHNICAL SUPPORT