SysCA: System for Access Control

SysCA system was specially developed for the reading of transponder devices (cards or others). SysCA system interfaces, through MCP XT controller, to the well tested CONTATTO world, integrating it with an easy to use access control system.

The access control system SysCA is made by a **ModTPD** module, that can be housed in a standard 503 wall box, and by a transponder reading unit **TPR/T** mounted on a panel for standard 503 wall box, with its integrated antenna and an **optional external antenna** (which operation is identical to the main one). The optional antenna can be installed in remote position in respect to the main unit.

Under request, the transponder programmer **KeyProg** and the proper **cards** are also available.

Two LEDs on the panel of the transponder reader report the information about valid code reading and not valid code reading.

ModTPD has two 9-way connectors; one of these allows the connection to TPR/T reading unit (4 wires) and to 3 LEDs replying the signaling on the panel (2 for valid code and one for not valid code). Another output on the same connector generates a 1 second pulse (at TTL level) when a valid code is detected; this output may be used, via a proper interface, for opening electrical controlled door lock or similar devices.

ModTPD module also provides, on the other 9-way connector, 8 digital generic inputs for the CONTATTO bus (for potential free contacts only).

Method for the access control

The access control is based on the approach of a transponder (card or other shapes) to one of the two antennas of the reading unit TPR/T. ***The two antennas cannot read two transponders at the same time.***

The validation of the read code, operated by the ModTPD module, is made according to rules as described in the following.

The transponder contains a 8-byte code that can be freely defined, within some limits, by the installer and transferred to the transponder through a proper programming device (a read/write transponder must be used). The ModTPD module is instead configured through the CONTATTO bus.

The first 6 bytes of the code are generally the same ones for all the users that must have access to the plant or building. These 6 bytes identify, for instance, the plant where the system is installed. The lat two bytes stored in the transponder identify exactly each user who must have access to the plant. It is therefore possible to define, for every plant, up to 65535 different users.

In addition, it is possible to define up to 32 user code intervals (or classes or groups), in order to make possible the implementation of group commands like the switching on of lights in common passages and/or the enabling of common crosses.

The transponder detected by the reading unit will be accepted as valid ***if the first 6 bytes (plant code) are identical to those stored in ModTPD module, and if the user code falls into one of the defined class***.

In more details, ModTPD can store up to 4 different plant codes, therefore the main condition for the validation is that the plant code, read from the transponder, matches one of the stored codes.

Of course, the definition of all the four plant codes is optional: only the codes required by the specific application have to be specified.

ModTPD module can be also configured in order to ignore (or mask) some bit or whole bytes during the plant code check; see in the following for more details.

Available information on the CONTATTO bus

The ModTPD module takes, inside the CONTATTO system, only one input address. On this address, the module provides 4 x 16-bit channels, with the following meaning:

- channels 1 and 2: 32 mutually exclusive bits, each one related to one of the 32 classes
- channel 3: used to report the status of the generic inputs of the module (point 1÷8), the OR of the points related to the classes (point 9) and the diagnostic about the transponder reading unit failure (point 16)
- channel 4: it reports the code of the transponder currently detected by one of the two antennas

Point	CH1	CH2	CH3	CH4
1	Class 1	Class 17	Input 1	User code currently detected. If zero, then the module is not detecting any valid code
2	Class 2	Class 18	Input 2	
3	Class 3	Class 19	Input 3	
4	Class 4	Class 20	Input 4	
5	Class 5	Class 21	Input 5	
6	Class 6	Class 22	Input 6	
7	Class 7	Class 23	Input 7	
8	Class 8	Class 24	Input 8	
9	Class 9	Class 25	OR of classes	
10	Class 10	Class 26	-	
11	Class 11	Class 27	-	
12	Class 12	Class 28	-	
13	Class 13	Class 29	-	
14	Class 14	Class 30	-	
15	Class 15	Class 31	-	
16	Class 16	Class 32	Read. Failure	

Since the class information reported by ModTPD is in a bit format (and this means Address:Channel.Point), it is very simple to handle it in MCP XT program using the standard equations (e.g. logic, set/reset and toggle).

For more complex operations, MCP XT can directly evaluate the user code by means of threshold equation or using the Script utility. In addition, MCP XT can record the user code in a buffer together to the date and time of the detection. This buffer, enabled in MCP XT by the LOG function (or LOGC if a circular buffer is needed), can store up to 1024 records; for more details, see the functions LOG and LOGC in the MCP XT user's manual.

Address programming

The ModTPD module take 1 input address that can be assigned by the FXPRO programmer through the proper programming cable inserted in the connector named PRG.

A white label on the front panel allow to make note of assigned address for an immediate visual identification.

Wiring diagram

ModTPD module must be connected to the reading unit TPR/T through 4 wires as shown in the schematic diagram; the reading unit TPR/T must be also connected to the 24Vdc power supply (+24V and 0V), that can be the same power supplying the bus.

For a proper using of generic inputs of ModTPD module, the command devices (pushbuttons, switches or others) must be free of any supply and connected to the module through the wires assembly connected to the IN connector.

On the OUT connector, in addition to the transponder reader TPR/T, up to 3 LEDs can be connected for reporting the acknowledgement information (two for the code OK and one for the code NON OK information), taking attention to the polarity shown in the schematic diagram; the resistors needed to limit the current flowing into the LEDs are included in the module, therefore no additional external components are required.

The OUT8 output provides a 1 second pulse when reading a valid transponder code, therefore this output can be used to control devices as electrical door lock or other similar ones by means of a proper relay interface (the output of ModTPD module is at 5V TTL level).

The second antenna, if used, must be connected as shown in the schematic through its shielded cable (with the shield connected to TPR/T unit only); the antenna has no polarity.

If an application needs longer wires from ModTPD to TPR/T, a 6 x 0.22mmq + shield cable is recommended, with shield connected to the 0V on the ModTPD side; anyway, do not exceed a length of 15 meters for this cable.

Setting up of ModTPD module

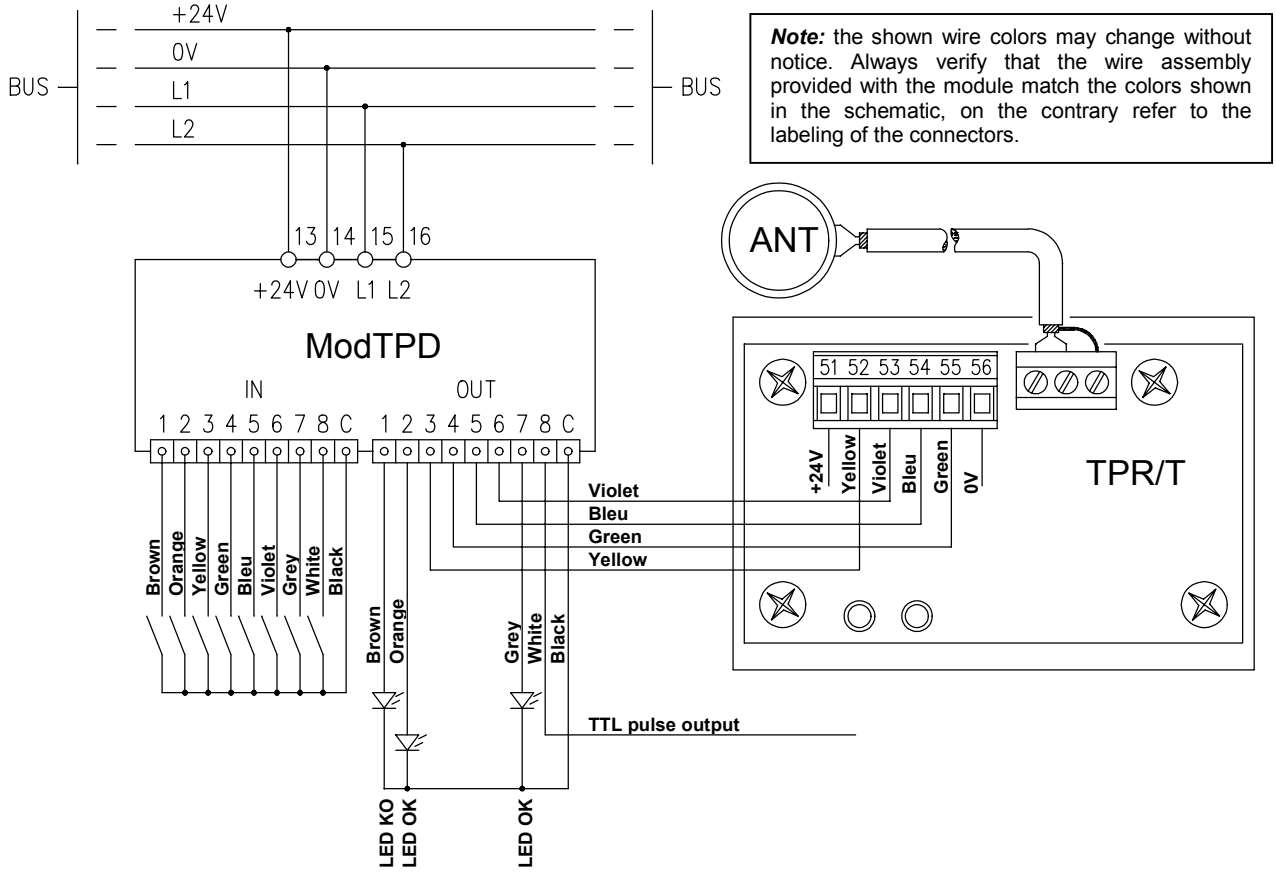
This paragraph describes some suggestions and examples about the using of the ModTPD access control module, its setting up and the programming of MCP XT.

However, the steps here described are not the unique rule to be followed, because the setting up depends on the specific application.

Note: ModTPD module can operate only in systems where MCP XT controller is installed; the setting up and the programming of MCP XT requires the program MCP IDE. The ModTPD programming, instead, requires the TPDTools add-on.

All ModTPD modules installed in the application must be declared in the MCP XT configuration, specifying the addresses as follows:

```
MODTPD = ( I1 )
MODTPD = ( I2 )
MODTPD = ( I3 )
… … … …
```



Note: the shown wire colors may change without notice. Always verify that the wire assembly provided with the module match the colors shown in the schematic, on the contrary refer to the labeling of the connectors.

For recording the accesses detected by the installed ModTPD modules, the **LOG** or the **LOGC** block has to be included in the MCP XT program as follows:

```
LOG = ( \
        AI1:4, \
        AI2:4, \
        AI3:4, \
)
```

This functions causes the recording, in a buffer of MCP XT, of all variations related to the points specified in the block. Since, in this example, the channel 4 of three ModTPD modules has been specified, then the user codes of acknowledged transponders will be registered together to date and time. The difference between the **LOG** and **LOGC** functions is that the first one stores the records of the first 1024 readings from the last reset of the buffer, while the second one stores the records of the last 1024 readings (circular buffer, see MCP XT manual for further details).

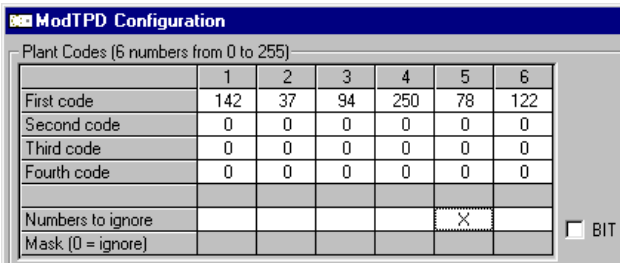
The records will be placed in the RAM memory of MCP XT that is supplied by an internal battery, therefore the records will be maintained even if a failure of the main power supply occurs.

Of course, in order to avoid loss of records, a supervisor connected to MCP XT is required to periodically download the information from the buffer of MCP XT.

After having transferred a program to MCP XT containing at least the addresses of installed ModTPD modules, the second step is the setting up of the plant codes to be recognized (up to 4 codes).

To execute this operation, the TPDTools program is required, allowing also to program the transponders (unless a different supervision program also supporting the setting up of ModTPD and of the transponders is provided for that plant). From the menu of MCP Visio select Configuration and then ModTPD; the TPDTools window will appear.

Select from the menu Configuration and then ModTPD Configuration. The following figure shows the section of the setting up window related to the programming of the 4 plant codes:

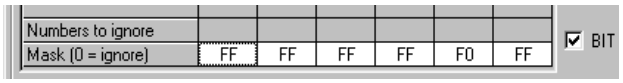


The unused plant codes (all bytes set to 0) will be automatically set by the program equal to one of the other codes; this because, for safety reasons, it is better to avoid to set to 0 all bytes of a plant codes.

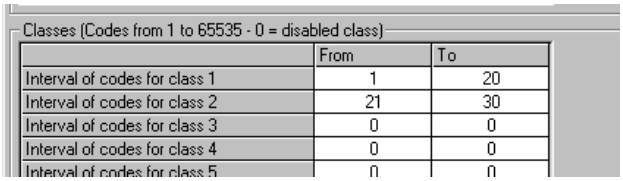
Clicking on one of the text boxes named “Numbers to ignore”, a X symbol will be shown; this means that ModTPD module, when comparing the code read from a transponder, will ignore that byte. Clicking again in the same text box, the X symbol will be removed. The byte 5, in the example shown in the figure, will be then ignored.

Clicking in the “BIT” check box, the “Mask” line will be enabled instead of the just described one; in this case the single bits of the plant code can be masked (and therefore ignored), simply inserting the related hexadecimal codes and taking in account that a bit = 0 means “ignore it”.

In the following figure, ModTPD module will ignore the 4 less significant bits of the byte 5 (because F0 hex is equal to 11110000 binary).



The following figure shows the program section related to the classes definition:



Take in account that the transponder, to be acknowledged, must belong to one of the specified classes.

The classes defined by an interval 0-0 will be ignored; for this reason the user code 0 is not allowed.

In the example shown by the previous figure, two classes have been specified:

- Class 1: from 1 to 20
- Class 2: from 21 to 30

In this example, when a ModTPD module having address **n** detects a transponder with user code (last 2 bytes) in the range 1 to 20, then it will activate the point **AIn:1.1**.

When instead a ModTPD module having address **n** detects a transponder with user code in the range 21 to 30, then it will activate the point **AIn:1.2**.

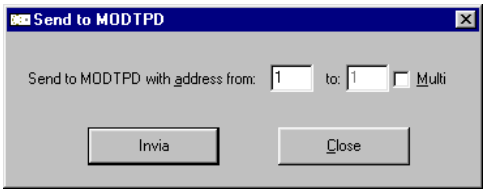
In both cases ModTPD module will activate also the point **AIn:3.9**, because this is the OR of the class points.

The point identifying the class of the acknowledged transponder, like the point related to the OR of the classes and the user code, remains activated for all time the reader detects the transponder itself. Of course, the point related to the class will be activated ***if and only if*** the plant code stored in the transponder (the first 6 bytes), considering the eventual masking, matches one of the 4 codes specified for that ModTPD module.

The user code field will contain a code other than zero if and only if the detected transponder has been accepted.

The point related to the user classes can be used in the standard equations of MCP XT, in order to execute several actions depending on the acknowledged user (e.g. employee, visitor, staff personnel, etc.).

Once all wanted parameters have been entered in the TPDTools program, press the button “Write ModTPD” to transfer the displayed information. The following window will be shown:



Specify the address of the ModTPD to be programmed or, activating the “Multi” option, specify a starting and a ending address; in this case the program will transfer the setting to all ModTPD modules found in that address range.

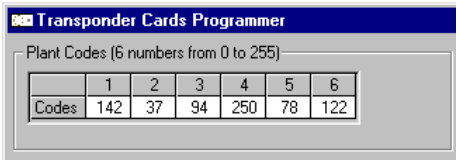
Of course, before the transferring, the communication with MCP XT must be enabled by the button “Communication”.

The TPDTools program allows to save in a file, having .TPD extension, the data shown in the ModTPD setting window (“Write on File”). It is also possible the opposite operation (“Read from File”).

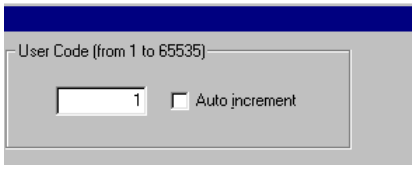
Setting up of the transponders

The TPDTools program also allows the setting up of the transponders (card or other devices) through the proper programmer KeyProg connected to the PC via the RS232 port; **DUEMMEGI** provides the programmer under request.

From the menu of TPDTools select Tool and then Card Programmer. The following figure shows the program section related to the entering of the plant code:



The plant code programmed in the cards, of course, must be equal to one of the 4 codes programmed into the installed ModTPD. The user code to be programmed in the card must be entered in the related text box of the program as in the following figure:



To open the communication with the transponder programmer press the button “Communication”. The button “Program” transfers the settings to the card placed on the programmer, while the button “Read” execute the opposite transferring.

The option “Auto Increment” allows to increment by 1 the user code after each programming; this feature is useful when many cards have to be programmed with consecutive user codes.

User code matching with MCP XT

For more complex operations, if the class information cannot perform the wanted operations, MCP XT can directly evaluate the user code by means of threshold equations as in the following example. Suppose that the application requires to control 120 different CONTATTO outputs for each one of the 120 users of a plant. Said 100 the address assigned to ModTPD, and supposed to have assigned the user code 1 to 120, the following 120 equations have to be loaded into MCP XT:

```
O1.1 = AI100:4 == 1
O1.2 = AI100:4 == 2
… … … …
O15.7 = AI100:4 == 119
O15.8 = AI100:4 == 120
```

Each one of these outputs will be driven when the ModTPD module detects the specified user code from a transponder, of course if the card contains a valid plant code and if that user code has been enclosed in a class.

Of course, it is also possible to use virtual points instead of the real outputs, in order to perform more complex operation; in some cases it is also possible to use the Script utility of MCP XT.

Technical characteristics

Supply voltage (ModTPD)	24V ⎓ ± 25% SELV
Supply voltage (TPR/T reader)	12÷26V ⎓ ± 10% SELV
MAX current consumption (ModTPD + TPR/T)	60mA (30mA + 30mA)
Number of digital inputs	8 for potential free contacts
Current for each input	1mA (with closed contact
Number of LED outputs	3
Current for each LED output	3.5mA internally limited
TTL pulse output: - Nominal duration - Low voltage level - High voltage level(active) - Available current	 1 second 0V 5V (on infinite load) 3.5mA internally limited
Max cable length: - digital inputs - outputs - ModTPD to TPR/T	 10 meters 10 meters 2 meters, 6x0.22 mm^2 shielded cable
Operating temperature	-10 ÷ +50 °C
Storage temperature	-30 ÷ +85 °C
Protection degree	IP20

ModTPD module in the standard 503 box

Note: the 503 standard box is not provided.

KeyProg transponder programmer

Note: provided with power supply and RS232 cable.

TPR/T reading unit

Note: the frame and the cover plate are not provided.

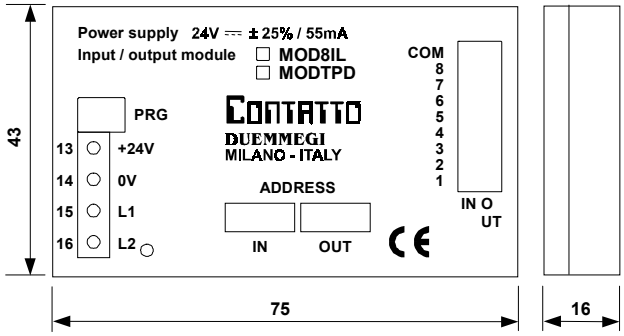
KEY transponder (badge shape)

AE/T external antenna

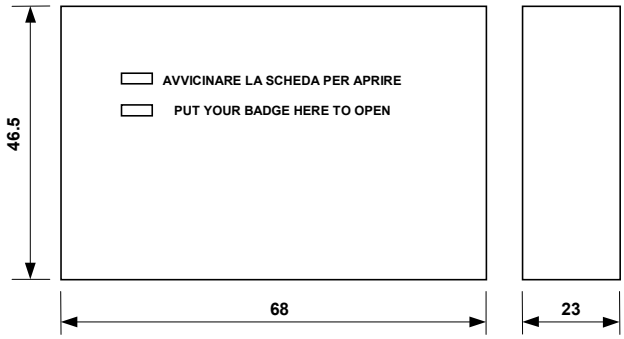
Note: the external antenna is an option; the cable length is about 2.5 meters.

Outline dimensions

ModTPD module



Reading unit TPR/T (shown dimensions change depending on the chosen frame):



Optional external antenna AE/T: