Using dimmer modules

This application note applies to Contatto dimmer modules MOD2DM and MOD2DV. These two models differ only for the output type (power for MOD2DM and voltage for MOD2DV); for this reason, the following application note may be applied to both modules.
In addition, all the description in this paragraph will be referred to one channel only of the dimmer module (because the mode of operation of both channels is the same).

Operation from the bus of dimmer module
Assigning address **n** to one channel of dimmer module, the executed function depends on the value of the data byte written to output address **On** as here bottom listed:

Data written to On	Function
0÷100	Set brightness to 0÷100%
101÷109	Save current brightness to setup 1÷9
111÷119	Recall brightness from setup 1÷9
128	No operation
129	Up command
130	Down command
131	Single command
133	Set brightness to 0 but store last value
134	Set brightness to last level value
135	Set current brightness as MIN value
136	Set current brightness as MAX value
137	Reset MIN level to default
138	Reset MAX level to default

In other words, values lower or equal to 100 will be managed as percentage of the brightness to be written to the output and values higher or equal to 101 will be managed as commands.

To send commands and values from MCP to the dimmer module, the counter registers have to be used as in the following example:

```
V100 = C0=1 P[129]I1.1 & P[130]I1.2 & \
       P[50]I1.3 & P[128]V1 & O1


V1 = !(I1.1 | I1.2 | I1.3)
```

where `I1.1` and `I1.2` are Up and Down inputs (pushbuttons) and `O1` is output address of channel 1 of the dimmer module; button connected to `I1.3` will cause the setting of brightness to 50%.

Thanks to the second equation defining `V1`, at the releasing of any pushbutton, the counter will be always set to 128 (no operation condition); ***be sure to place inside the round brackets all the inputs (real and virtual) used in the counter equation controlling the dimmer***.

The value of the counter, for each variation, will be transferred to the dimmer output (`O1`) and the performed function will be as specified in above table. To drive more channels or modules, simply add more "`& Ox`" to the equation in above example.

Note that an 8-bit counter was used and that `C0=1` has not any meaning but is required by the counter function syntax; also V100 is required by the counter syntax even if it is not used.
If a supervisor is connected to MCP module, it may set any value in the range 0 to 100 simply writing this value in the counter register.

Each dimmer code will be described in the following of this application note.

Set brightness to 0÷100%
Sending to dimmer module a value in the range 0 to 100, the related brightness level will be set. For MOD2DM power dimmer module values in the range 85 to 100 will set the brightness to 85%, because this is the maximum allowed value.

Save and recall (codes 101÷109 and codes 111÷119)
Codes 101÷109 force the dimmer module to store the current brightness value into one of the 9 "dynamic" presets; since the module has a non volatile-memory, these value will be retained even if the power supply is removed.

Codes 111÷119 force the dimmer module to recall the "dynamic" preset previously stored and to adjust the brightness at the related value. These functions are useful to create custom lighting scenes without re-programming MCP module.

Saving and recalling may be easily implemented as in the following example:

```
V1 = !(I1.1 | I1.2 | V11 | V12 | V13 | V21 | V22 | V23)


V100 = C0=1 P[129]I1.1 & P[130]I1.2 & \   // Up-Down
       P[101]V11 & P[111]V21 & \          // Save-recall 1
       P[102]V12 & P[112]V22 & \          // Save-recall 2
       P[103]V13 & P[113]V23 & \          // Save-recall 3
       P[128]V1 & O1
```

`V11-V13` points perform the saving and `V21-V23` points perform the recalling of related brightness value. These points may be controlled both by a supervisor and by real inputs (e.g. connected to pushbuttons or to a remote control receiver).

As option, the supervisor can write directly in the counter the code of save-recall functions; as example, some buttons performing the sending of saving codes and some others performing the sending of recalling codes may be placed on the screen of the PC. In this way, the virtual points in above equation can be removed as follows:

```
V1 = !(I1.1 | I1.2)
V100 = C0=1 P[129]I1.1 & P[130]I1.2 & \   // Up-Down
       P[128]V1 & O1
```

In this last example, the PC must send the “no operation” code (128) at the releasing of buttons on the screen.

No operation (code 128)
Code 128 must be always loaded in the counter controlling the dimmer when no active commands are pending.

Up and Down commands (codes 129 and 130)
Codes 129 and 130 allow controlling of dimmer output through the bus from two inputs (increase and decrease).

Single command (code 131)
Code 131 allows controlling of dimmer output through the bus from a single input (increase/decrease).

Set brightness to 0 but store last value (code 133)
Code 133 forces the brightness to zero (lamp completely off) but the last value will be stored into non-volatile memory of the dimmer module. This command is useful in all cases where a centralized management of lighting is needed by the supervisor (PC) or by a main switching off pushbutton.

Suppose to have many dimmer modules installed in some offices in a building: each lamp may be controlled by local pushbuttons in order to have the freedom to adjust the brightness of each office as desired. When people leave the offices at the end of working day, each lamp may be lighted at different brightness values (and other lamps may be already off).
Sending code 133 to all dimmer modules (by supervisor or by a main pushbutton), all the lamps will be completely turned off. The day after, when people returns in the offices, a short touch on UP or DOWN local pushbuttons will cause the turning-on of the related lamp at the same brightness value of last adjustment.

Set brightness to last level value (code 134)
Code 134 forces the brightness to last stored value; this command is useful in all cases where a centralized management of lighting is needed by the supervisor (PC) or by a main switching-on pushbutton.

In the same example of previous paragraph, suppose that in the morning, at the opening of the offices, the service operator wants to turn-on all the lamps at the same brightness value of the evening before; this is possible sending code 134 to all dimmer modules.

Set current brightness as MIN value and Set current brightness as MAX value (codes 135 and 136)
These commands are useful to limit the excursion of the brightness level; default values are 10% to 100% for MOD2DV and 15 to 85% for MOD2DM.

After having adjusted the brightness as desired, send code 135 to set it as minimum value. Adjust now the brightness to another value (greater than the previous one) and then send code 136 to set it as maximum value. At this point the brightness adjustment is allowed inside the two defined levels. These values will be stored into non-volatile memory of the dimmer module.

Reset MIN level to default and Reset MAX level to default (codes 137 and 138)
These commands allow the restoring of MIN and MAX brightness level to default values.

Using* `#define` *directive to simplify programs when using dimmer codes
MCP-Tools 3.1.0 (or higher releases) provides a powerful version of **`#define`** directive, which result very helpful when programming dimmer modules. The **`#define`** directive allows to assign a name to part of an equation; during the compiling process, the occurrence of the name will be replaced with its true meaning. The name must be made by alphanumeric characters and included inside two % symbols; no spaces or other % characters are allowed inside the name.
The following example describes the use of this directive.

```
#define     %NOP%       P[128]      // no operation
#define     %UP%        P[129]      // Up command
#define     %DW%        P[130]      // Down command
#define     %P50%       P[50]       // Preset 50%
#define     %P70%       P[70]       // Preset 70%
#define     %Main_Off%  P[133]I41.2 // Centralized switching off from I41.2

V1 = !(I1.1 | I1.2 | I1.3 | I1.4 | I41.2)
V100 = C0=1 %UP%I1.1   & %DW%I1.2  & \
            %P50%I1.3  & %P70%I1.4 & \
            %Main_Off% & %NOP%V1 & O1

V2 = !(I2.1 | I2.2 | I2.3 | I2.4 | I41.2)
V101 = C1=1 %UP%I2.1   & %DW%I2.2  & \
            %P50%I2.3  & %P70%I2.4 & \
            %Main_Off% & %NOP%V2 & O2
```