Using day plans in control systems

Abstract

The application note deals with programming of day plans in control systems by means of the DetStudio development environment and their parametrization using the ViewDet environment.

Author: Zbyněk Říha
File: ap0031_en_04.pdf

Attachments

File content: ap0031_en_03.zip

dayplan_p1_en_03.dso	Day plans in a simplified mode (TPlan, GPlan, Feast, Holiday).
dayplan_p2_en_03.dso	Day plans in an extended mode (TPlan, Feast, Holiday).
dayplan_p3_en_01.mdb	Day plans in ViewDet.
dayplan_p4_en_01.dso	Working with various day plans on one screen.

amitomation.com

Contents

Revision history

Version	Date	Changes
001	16. 12. 2008	New document
002	17. 05. 2010	A note in Chapter 1; added a chapter on the structure of time shifts and value matrices added.
003	07. 05. 2012	Attached applications modified, applications created for DetStudio version 1.7.2, related documents modified, pictures modified according to the behaviour of DetStudio 1.7.2.
004	24. 03. 2014	Attached applications modified, applications created for DetStudio version 1.7.4, chapters 1, 1.1 and 1.2 modified, a new Chapter 6 added, a new sample application created.

Related documentation

1. Help tab for DetStudio development environment
   file: Psedet_xx.chm (Help for PseDet) and Tridet_xx.chm (Help for screen design)
2. Help tab in the ViewDet development environment
   file: ViewDet.xx-xx.chm
3. Application note AP0003 – Application Designed in Cyrillic
   file: ap0003_en_xx.pdf
4. Application note AP0023 – Scripting in DetStudio
   file: ap0023_en_xx.pdf

1 Day Plans

Day plans can be implemented in DetStudio in the programme section by means of modules **DayPlan**/**DayPlan2**. They can be parametrized in the ViewDet environment. This application note describes in detail how operation of time plans on terminals is programmed. This may then be implemented by means of elements

- **TPlan**, **Holiday**, **Feast** – all types of terminals/control systems with a screen.
- **GPlan** – terminals/control systems with graphic displays (AMiNi2D, ADOREG, …).
- **CPlan** – graphic terminals (APT3x00, APT3x00S, APT3xxxWT).

Modules **DayPlan**/**DayPlan2** allow for planning of values for individual days of the week. Using these modules, we can create special cycles for each day of the week, or more specifically for a group of days. In order to work properly, these modules require we set a time shift matrix, matrix of time shifts values, matrix for days off and an output day-plan variable. In case we will also use the holiday mode (using the **Holiday** module), it is also necessary to define the holiday matrix.

Set the parameter **Mode**/**PlanningType** in modules **DayPlan**/**DayPlan2** to set level planning or linear (analogue) planning. All examples in this application note have level planning set.

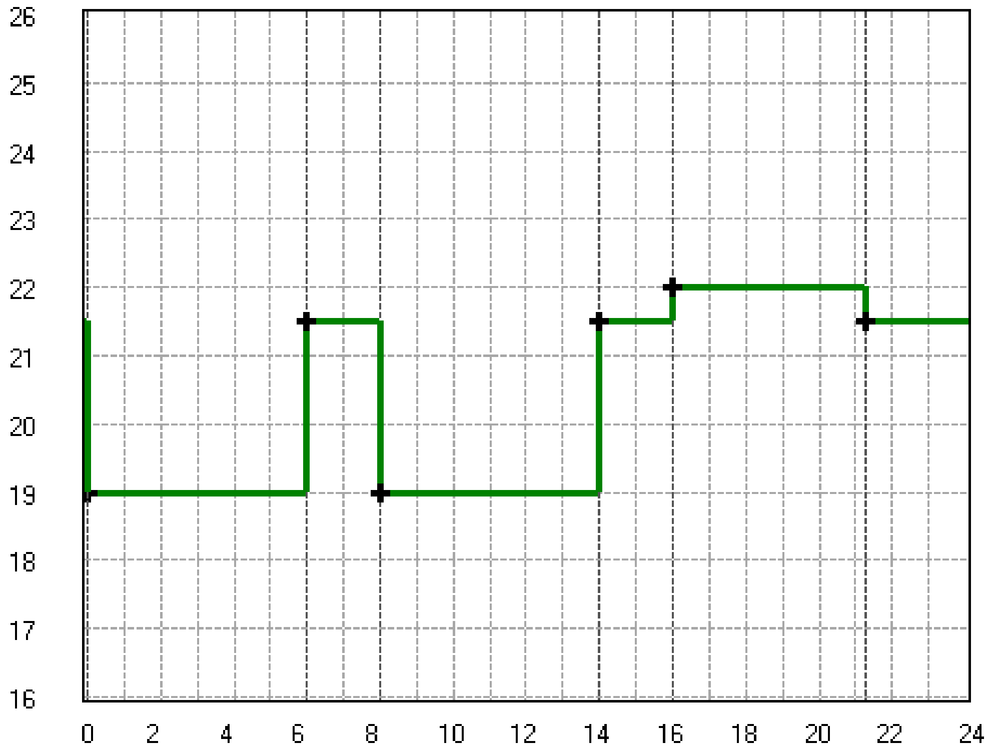


Fig. 1 – The parameter **Mode**/**PlanningType** equals zero (level planning)

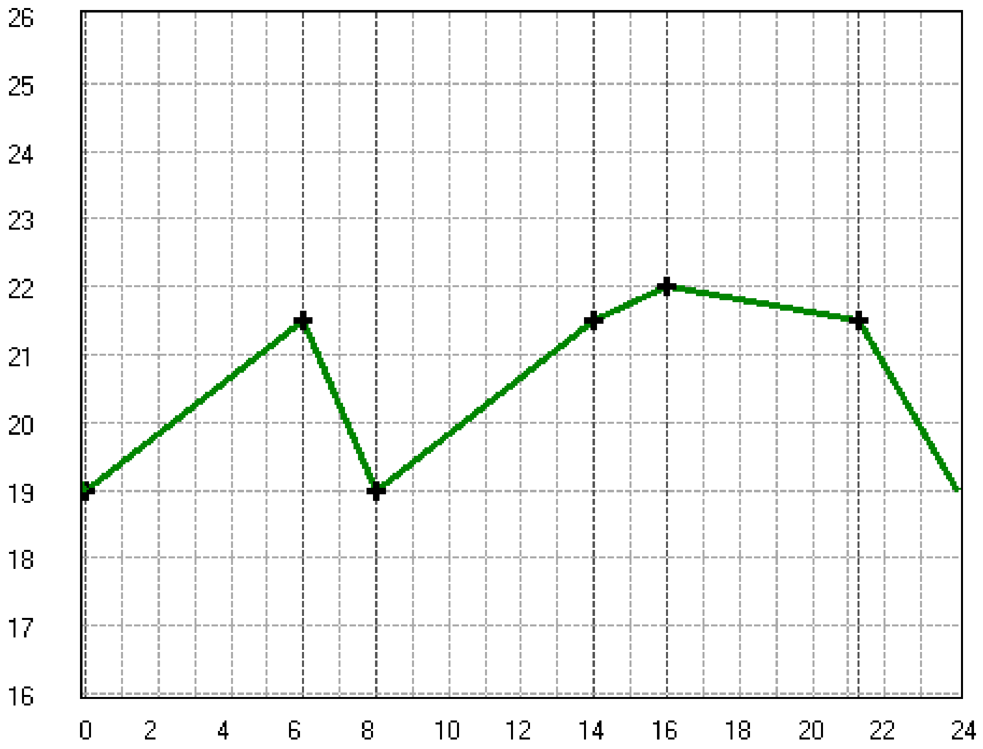


Fig. 2 – The parameter **`Mode`**/**`PlanningType`** equals one (linear/analogue planning)

Structure of the Time Shift Matrix and of the Matrix of Values for Time Shifts

Matrix of dimensions [**`n×m`**] where **`n`** must be higher than or equal to the number of plans set for individual days and **`m`** is the number of time shifts in the given day plan.
The matrix columns determine the individual points of shifts, i.e. how many times the planned variable may change in one day (dimension **`m`**). The matrix must always contain at least one shift (has to have at least two columns). The matrix lines determine day plans for individual days (dimension **`n`**). In the **`DayPlan`** module, each line of these matrices has a specific day code assigned according to which the module recognizes (according to the current date) which line should be active. The codes may not only determine individual days of the week, but also their possible combinations.

The matrices always have to be set.

Note:
*When setting the time shift values, we recommend you set the first time shift as the value 00:00:00. Controls **`GPlan`** and **`CPlan`** always enter the value 00:00:00 into the first time shift, regardless of the previous day plan setting.*
*If the first time shift is not set to the value 00:00:00, the module **`DayPlan`**/**`DayPlan2`** will set the value that has been set for the last time shift in the current day on its output (see Fig. 3 – Day plan with the first shift different from the time 00:00:00, where the dashed red line marks the level planning and the full green line marks the linear planning).*

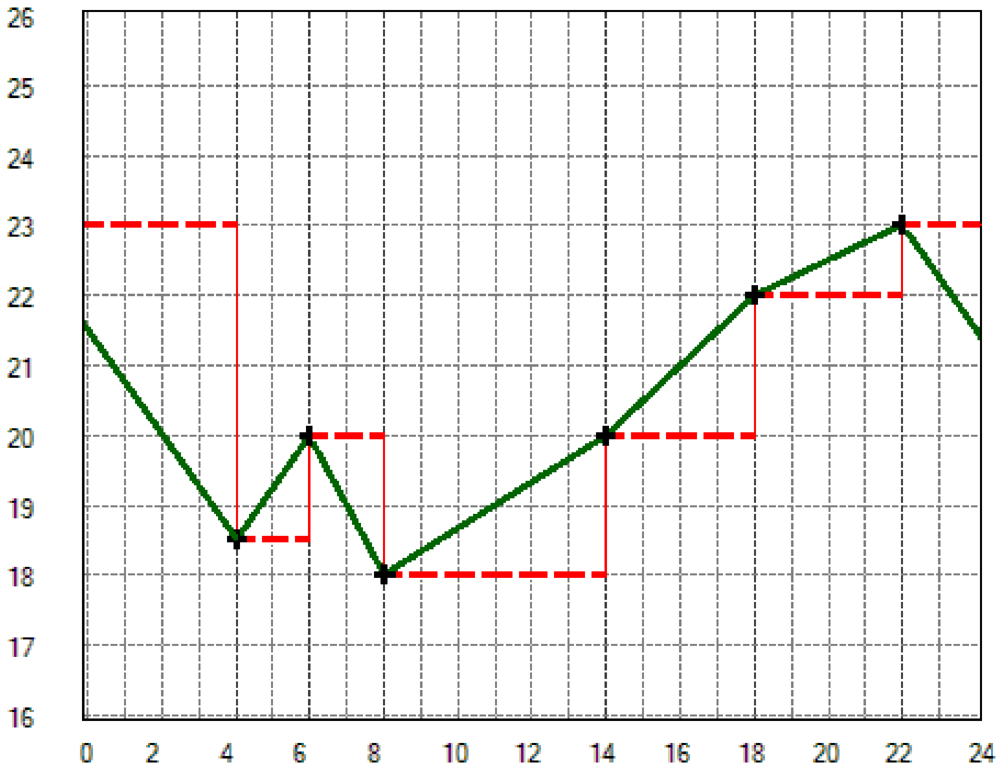


Fig. 3 – Day plan with the first shift different from the time 00:00:00

Day off Matrix Structure

The matrix with dimensions [3×**n**] where **n** is the number of days off we want to enter. The matrix lines have the following description:

- 0 – day,
- 1 – month,
- 2 – code of the day to be considered as a day off (e.g. Sunday).

The matrix always has to be set. If we do not work with Days off, it suffices to define the matrix [3×1].

Holiday Matrix Structure

The matrix with dimensions [5×**m**] for the holiday period where **m** is the number of holiday periods (sections) in the year. The matrix lines have the following description:

- 0 – first day of holiday,
- 1 – the month of the holiday beginning,
- 2 – the day of the holiday end,
- 3 – the month of the holiday end,
- 4 – code of the day to be considered as a holiday (e.g. Sunday).

The matrix only has to be set in cases we intend to work with holidays.

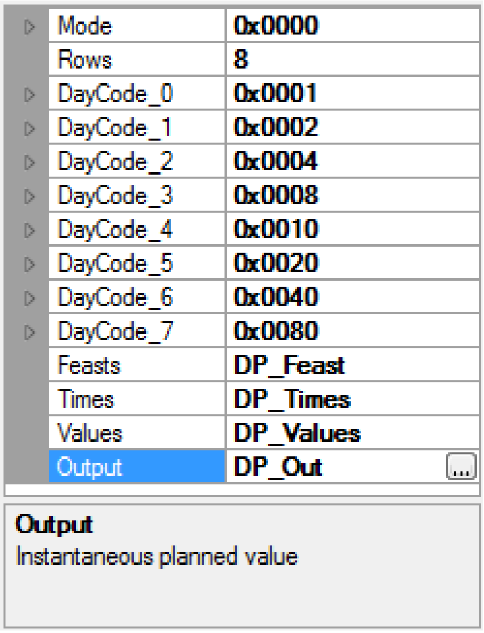
We can set and edit day plans on the display/control system terminal in two ways:

- Simplified mode,
- Extended mode.

1.1 Simplified mode

In this mode, the parameters of module **DayPlan** must be set in order to allow to plan for every day of the week (matrix of time shifts and matrix of values for time shifts must have 7 lines), or for each day of the week plus Day off (matrix of time shifts and matrix of values for time shifts must have 8 lines). The user sets the following in the **DayPlan** module:

- value 7 into the parameter **Rows** (7 day plans for individual days of the week) or value 8 (7 day plans for individual days of the week + Day off),
- codes of individual days (Mon to Sun, Day off),
- basic links to matrices and variables.



Mode	0x0000
Rows	8
DayCode_0	0x0001
DayCode_1	0x0002
DayCode_2	0x0004
DayCode_3	0x0008
DayCode_4	0x0010
DayCode_5	0x0020
DayCode_6	0x0040
DayCode_7	0x0080
Feasts	DP_Feast
Times	DP_Times
Values	DP_Values
Output	DP_Out

Output
Instantaneous planned value

Fig. 4 – Parametrization of the **DayPlan** module for the individual days of the week + Day off)

In this mode, we will be able to use control modules to enter day plans both for individual days of the week and for groups of days. They have the following order and definition when using feratures **TPlan**, **GPlan** and **CPlan**:

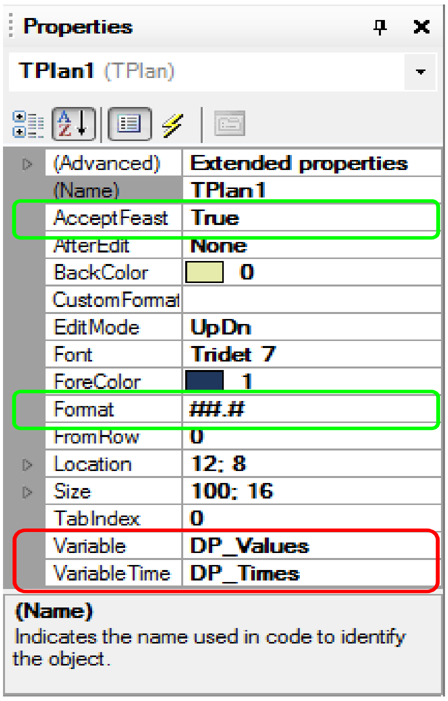
- Monday to Friday (work days),
- Saturday to Sunday (weekend),
- Monday to Sunday + Day off (all days),
- Monday,
- Tuesday,
- Wednesday,
- Thursday,
- Friday,
- Saturday,
- Sunday,
- Holiday.

The user then chooses the option that is most suitable for his editing: The texts are naturally multilingual (see AP0003 – Application Designed in Cyrillic).

1.1.1 Parametrization of the TPlan control

In the `TPlan` control, it is sufficient to set the matrix for values of time shifts into the parameter `Variable` and to set the matrix of time shift into the parameter `Variable_Time`. If necessary, we are also able to enter the display format of the required value in individual time shifts (parameter `Format`).

If we work with holidays, we also need to set the parameter `AcceptFeast` to the value True.



Properties

TPlan1 (TPlan)

(Advanced)	Extended properties
(Name)	TPlan1
AcceptFeast	True
AfterEdit	None
BackColor	0
CustomFormat	
EditMode	UpDn
Font	Tridet 7
ForeColor	1
Format	###.#
FromRow	0
Location	12; 8
Size	100; 16
TabIndex	0
Variable	DP_Values
VariableTime	DP_Times

(Name)
Indicates the name used in code to identify the object.

Fig. 5 – Example of parametrization of the `TPlan` control

Attention
The number of time shifts that can be defined using the `TPlan` control in the control system corresponds to the number of columns of the time shift matrix (the number of columns of the time shift matrix must be identical to the number of columns of the matrix for time shift values).

1.1.2 Parametrization of the GPlan control

In the `GPlan` control, it is sufficient to set the matrix for values of time shifts into the parameter `Variable` and to set the matrix of time shift into the parameter `Variable_Time`.

If we work with holidays, we also need to set the parameter `AcceptFeast` to the value True.

Attention
If the parameter `AcceptFeast` is set to the value True, the matrix entered into parameters `Variable` and `Variable_Time` must have at least 8 lines. If the parameter `AcceptFeast` is set to the value False, the matrix must have at least 7 lines. If this condition is not met, the application cannot be generated!

The `Gplan` control is able to work in two modes:

- Digital Mode,
- Level/analogue mode.

Digital Mode

In this mode, we may only enter the value 1/0 (On/Off) in the day plan. The mode is activated by setting the parameter **AnalogMode** to the value False.

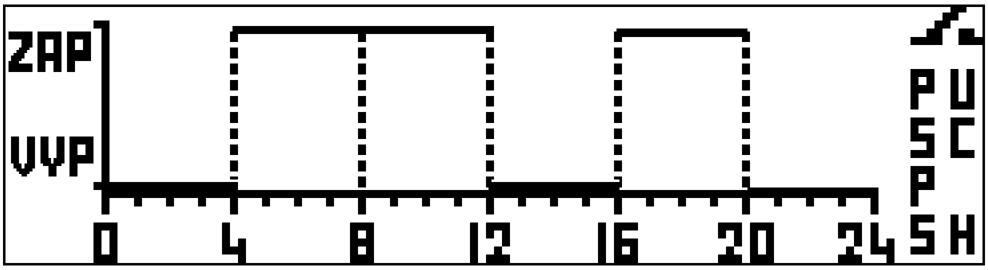


Fig. 6 – The **Gplan** control in digital mode

Level/analogue mode

In this mode, we can put in integer or decimal values in the day plan. The mode is activated by setting the parameter **AnalogMode** to the value True.

The **GPlan** control will always display level planning on the screen of the terminal/control system, even in case linear planning is set in the **DayPlan** module. Actual planning (level/linear) will always be performed in the control system according to the setting in the **DayPlan** module.

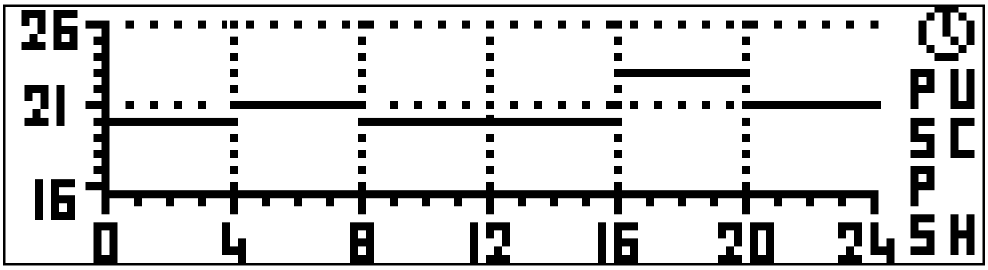
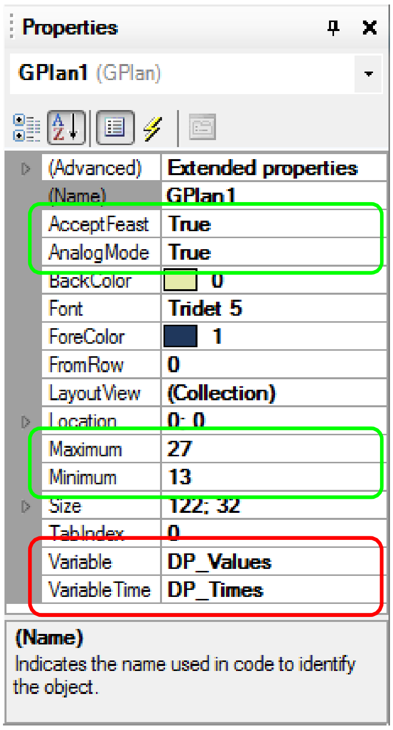


Fig. 7 – The **Gplan** control in analogue mode

The size of the required variable value in the given time shift can be limited by means of parameters **Maximum** and **Minimum**.



Properties

GPlan1 (GPlan)

(Advanced)	Extended properties
(Name)	GPlan1
AcceptFeast	True
AnalogMode	True
BackColor	🟨 0
Font	Tridet 5
ForeColor	■ 1
FromRow	0
LayoutView	(Collection)
Location	0; 0
Maximum	27
Minimum	13
Size	122; 32
TabIndex	0
Variable	DP_Values
VariableTime	DP_Times

(Name)
Indicates the name used in code to identify the object.

Fig. 8 – Example of parametrization of the **GPlan** control

Attention
The number of time shifts that can be defined using the ***GPlan*** *control in the control system corresponds to the number of columns of the time shift matrix decreased by one (the number of columns of the time shift matrix must be identical to the number of columns of the matrix for time shift values). This is determined by the character of the* ***GPlan*** *control when the control always starts planning from the time 00:00. The behaviour differs from the* ***TPlan*** *control and that is why we do not recommend using screen editing of day plans by means of the controls* ***TPlan*** *and* ***GPlan*** *at the same time.*

1.1.3 Parametrization of the CPlan control

In the **CPlan** module, it is sufficient to set the matrix for values of time shifts into the parameter **Variable** and to set the matrix of time shift into the parameter **Variable_Time**.

If we work with holidays, we also need to set the parameter **AcceptFeast** to the value True.

Attention
If the parameter ***AcceptFeast*** *is set to the value True, the matrix entered into parameters* ***Variable*** *and* ***Variable_Time*** *must have at least 8 lines. If the parameter* ***AcceptFeast*** *is set to the value False, the matrix must have at least 7 lines. If this condition is not met, the application cannot be generated!*

The **Cplan** control is able to work in two modes:

- Digital Mode,
- Level/analogue mode.

Digital Mode

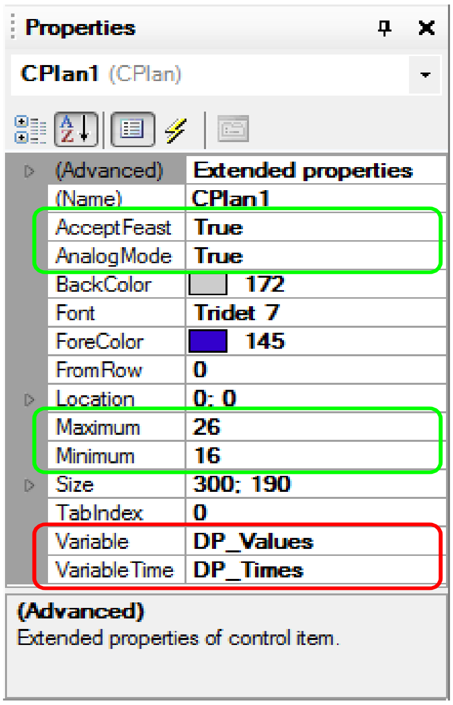
In this mode, we may only enter the value 1/0 (On/Off) in the day plan. The mode is activated by setting the parameter **`AnalogMode`** to the value False.

Level/analogue mode

In this mode, we can put in integer or decimal values in the day plan. The mode is activated by setting the parameter **`AnalogMode`** to the value True.

The **`CPlan`** control will always display level planning on the screen of the terminal/control system, even in case linear planning is set in the **`DayPlan`** module. Actual planning (level/linear) will always be performed in the control system according to the setting in the **`DayPlan`** module.

The size of the required variable value in the given time shift can be limited by means of parameters **`Maximum`** and **`Minimum`**.



Properties

CPlan1 (CPlan)

(Advanced)	Extended properties
(Name)	CPlan1
AcceptFeast	True
AnalogMode	True
BackColor	□ 172
Font	Tridet 7
ForeColor	■ 145
FromRow	0
Location	0; 0
Maximum	26
Minimum	16
Size	300; 190
TabIndex	0
Variable	DP_Values
VariableTime	DP_Times

(Advanced)
Extended properties of control item.

Fig. 9 – Example of parametrization of the **`CPlan`** control

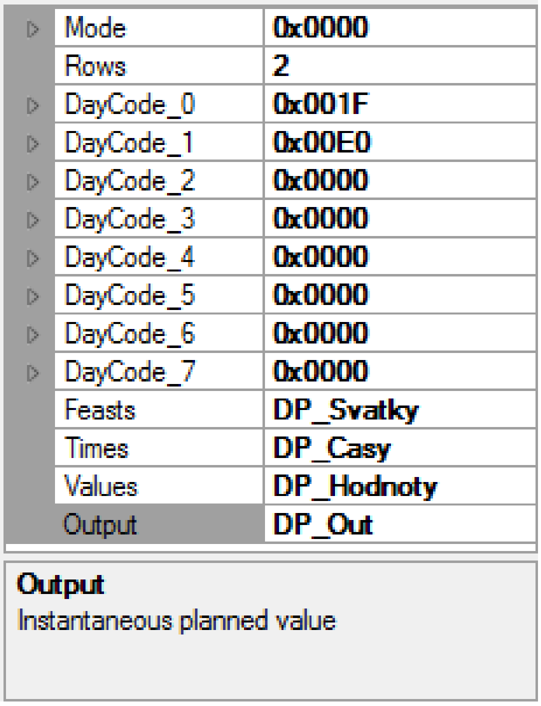
Attention
*The number of time shifts that can be defined using the **`CPlan`** control in the control system corresponds to the number of columns of the time shift matrix decreased by one (the number of columns of the time shift matrix must be identical to the number of columns of the matrix for time shift values). This is determined by the character of the **`CPlan`** control when the control always starts planning from the time 00:00. The behaviour differs from the **`CPlan`** control and that is why we do not recommend using screen editing of day plans by means of the controls **`TPlan`** and **`CPlan`** at the same time.*

1.2 Extended mode

It is necessary to use this mode in cases when using the simplified mode is not suitable, e.g. because of a request for editing of user groups of days of the week or a smaller amount of groups of days in the week (workdays, weekend, Day off, etc.). In such cases, it is necessary for time shift matrices and matrices of time shift values have as many rows as there are day groups required.

In case of the extended mode, we cannot use control controls `Gplan`, `CPlan` and there are no multilingual texts. We can only use the `TPlan` control or controls `Holiday` and `Feast`.

Then we need to define texts for individual day groups for the `TPlan` control. These are entered into matrix type MI (we need to maintain the matrix dimensions, see below).



Mode	0x0000
Rows	2
DayCode_0	0x001F
DayCode_1	0x00E0
DayCode_2	0x0000
DayCode_3	0x0000
DayCode_4	0x0000
DayCode_5	0x0000
DayCode_6	0x0000
DayCode_7	0x0000
Feasts	DP_Svatky
Times	DP_Casy
Values	DP_Hodnoty
Output	DP_Out

Output
Instantaneous planned value

Fig. 10 – Parametrization of the `DayPlan` module for two day groups

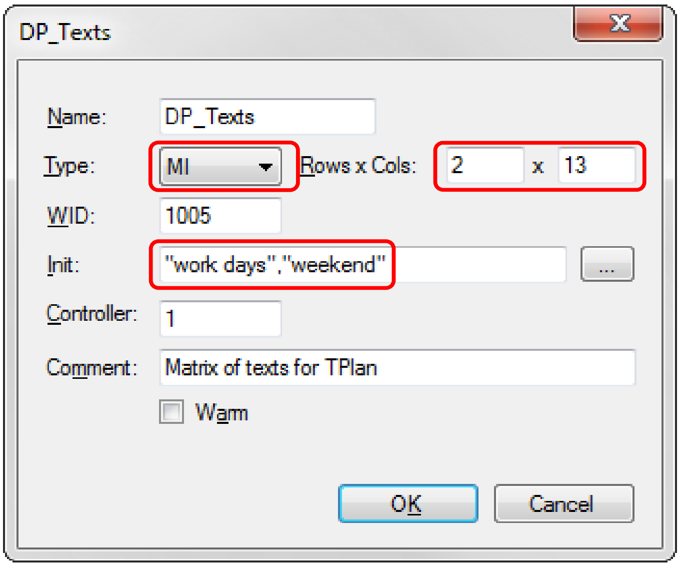
Figure 9 shows the parametrization of the `DayPlan` module for two day groups:

- Work days,
- Weekends and Days off.

In this type of parametrization, it is necessary for the time shift matrix and matrix of time shift values to only have 2 rows (row line for each group). We also need to set value 2 in the parameter `Rows` (two day plans).

1.2.1 Parametrization of the TPlan control

If we require that the user is able to edit only day groups (work days/weekends and days off), it is necessary to assign a data matrix to the `TPlan` control that contains texts to be displayed in the `TPlan` control (for individual day groups). This matrix has to have the same number of lines as the time shift matrix and the matrix of time shift values.



DP_Texts

Name: DP_Texts

Type: MI

Rows x Cols: 2 x 13

WID: 1005

Init: "work days","weekend"

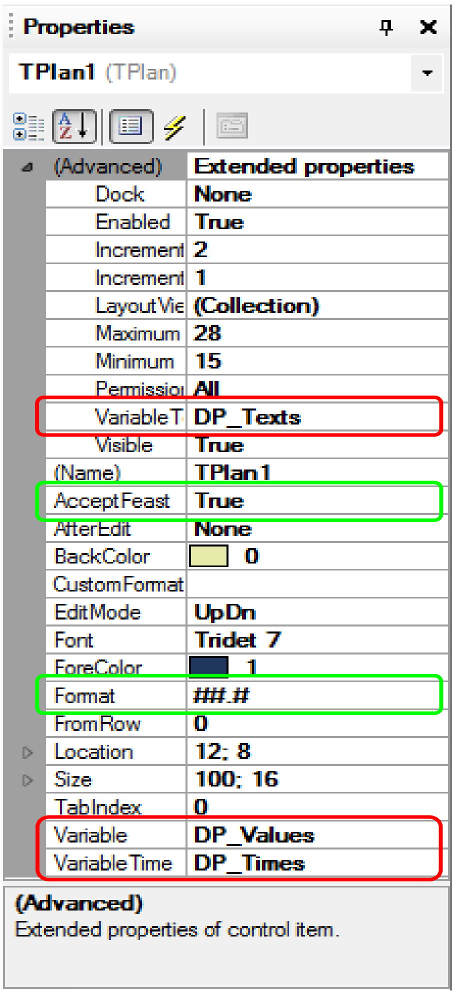
Controller: 1

Comment: Matrix of texts for TPlan

□ Warm

Fig. 11 – Definition of text matrix for day groups

Subsequently, we assign the text matrix to the parameter **`VariableText`** found in the Properties of the control **`TPlan`** in the section **`Advanced`**.



Properties

TPlan1 (TPlan)

(Advanced)	Extended properties
Dock	None
Enabled	True
Increment	2
Increment	1
LayoutVie	(Collection)
Maximum	28
Minimum	15
Permissio	All
VariableT	DP_Texts
Visible	True
(Name)	TPlan1
AcceptFeast	True
AfterEdit	None
BackColor	□ 0
CustomFormat	
EditMode	UpDn
Font	Tridet 7
ForeColor	■ 1
Format	###.#
FromRow	0
Location	12; 8
Size	100; 16
TabIndex	0
Variable	DP_Values
VariableTime	DP_Times

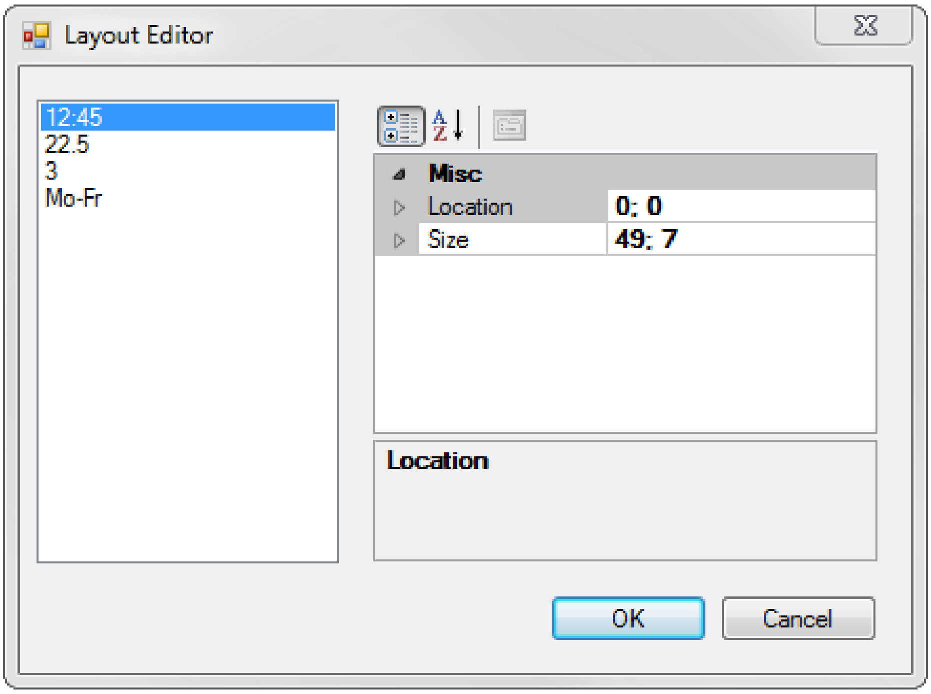
(Advanced)
Extended properties of control item.

Fig. 12 – Example of parametrization of the **`TPlan`** control

In the **TPlan** control, it is further necessary to set the matrix for values of time shifts into the parameter **Variable** and to set the matrix of time shift into the parameter **Variable_Time**. If necessary, we are also able to enter the display format of the required value in individual time shifts (parameter **Format**).

If we work with holidays, we also need to set the parameter **AcceptFeast** to the value True.

Attention
*Depending on the length of user texts entered in the text matrix, it is necessary to modify the position and dimensions of individual texts of the **TPlan** control by means of its parameter **Advanced/LayoutView**.*



12:45
22.5
3
Mo-Fr

Misc
Location 0; 0
Size 49; 7

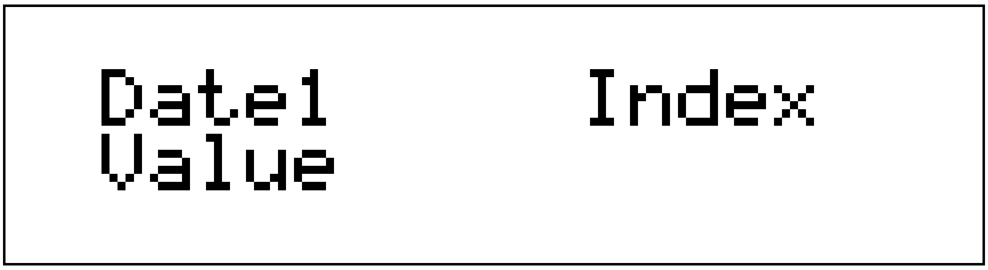
Location

Fig. 13 – Setting text disposition in the control **TPlan**

2 Days off

Days off are defined in the control system by means of the day off matrix entered in the **`DayPlan`** module for the parameter **`Feast`** (the matrix must always be defined, even if we do not use day off days). The day off matrix structure is described in Chapter 1. Day Plans.

Entering Days off by means of a control system terminal is then possible by means of the **`Feast`** control.



```
Date1        Index
Value
```

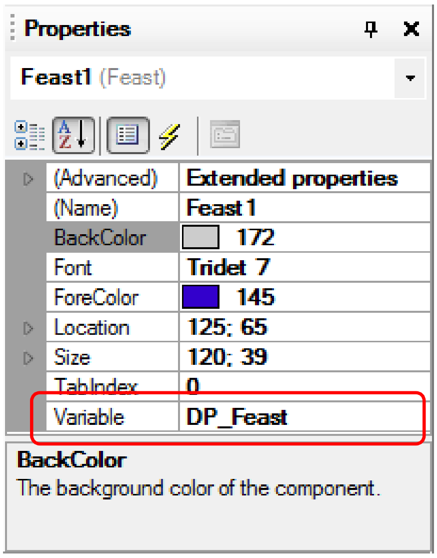
Fig. 14 – The **`Feast`** control after being entered on the screen

We can enter Days off in two ways:

- Simplified mode,
- Extended mode.

2.1 Simplified mode

In the simplified mode, we only put the **`Feast`** control on the screen and we enter the Day off matrix into the parameter **`Variable`**.



Feast1 (Feast)

(Advanced)	Extended properties
(Name)	Feast1
BackColor	□ 172
Font	Tridet 7
ForeColor	■ 145
Location	125; 65
Size	120; 39
TabIndex	0
Variable	DP_Feast

BackColor
The background color of the component.

Fig. 14 – Example of parametrization of the **`Feast`** control

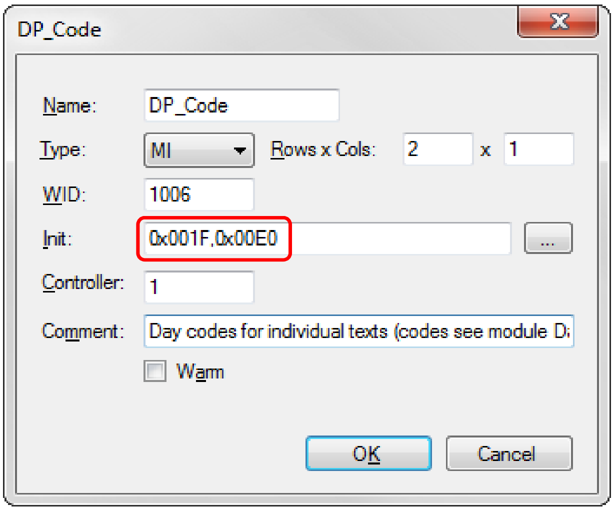
Then we set the date of the day off and the day according to which the day off is to be planned on the control system terminal.

This mode can use multilingual texts (see AP0003 – Application Designed in Cyrillic).

2.2 Extended mode

If we work with day plan in extended mode (or we want to display other than default texts by means of the **Feast** control) and we require that the **Feast** control displays the same range of days as in the **TPlan** control, we must work with the **Feast** control in the extended mode as well.

The extended mode of the **Feast** control requires setting of the text matrix (see Chapter 1.2.1 Parametrization of the TPlan) and of the relevant day codes to individual plans



DP_Code

Name: DP_Code

Type: MI

Rows x Cols: 2 x 1

WID: 1006

Init: 0x001F,0x00E0

Controller: 1

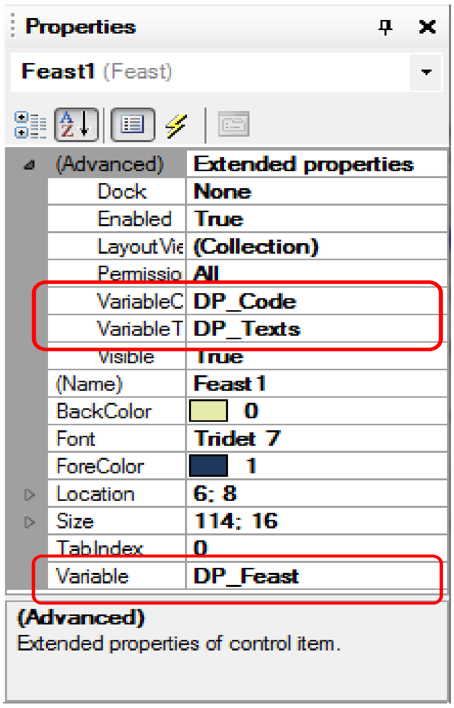
Comment: Day codes for individual texts (codes see module D

□ Warm

Fig. 15 – The definition of day codes matrix for individual plans

Attention
The day codes matrix must have the same number of lines as the text matrix for individual plans.

Text matrix will be assigned to the parameter **Advanced/VariableText**, the day codes matrix will be assigned to the parameter **Advanced/VariableCode** of the **Feast** control.



Properties

Feast1 (Feast)

(Advanced)	Extended properties
Dock	None
Enabled	True
LayoutVie	(Collection)
Permissio	All
VariableC	DP_Code
VariableT	DP_Texts
Visible	True
(Name)	Feast1
BackColor	□ 0
Font	Tridet 7
ForeColor	■ 1
Location	6; 8
Size	114; 16
TabIndex	0
Variable	DP_Feast

(Advanced)
Extended properties of control item.

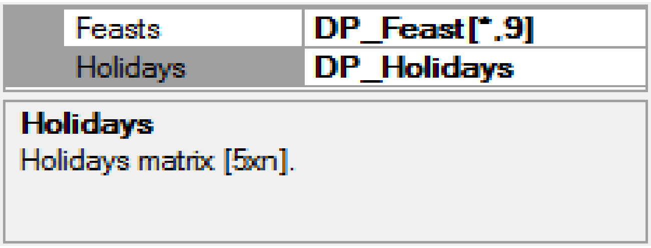
Fig. 16 – Example of parametrization of the **Feast** control

Attention
When entering longer user texts for individual plan types, it is necessary to modify the position and dimensions of individual texts of the ***`Feast`*** *control on the screen by means of its parameter* ***`Advanced`****/****`LayoutView`****.*

This mode does not allow the use of multilingual texts.

3 Holidays

It is necessary to use the **Holiday** module in the programme part in order to define holidays in the control system. This module determines a day code for each day of the holiday period to set how the given day of the holiday should be interpreted (e.g. as a Saturday, Sunday...). In the variable set in the **DayPlan** module for the parameter **Feasts** (holiday matrix), it is then necessary to assigned one extra column into with the **Holiday** module writes a value every day during the holiday period. We initiate this column with zeros, which means "unused".



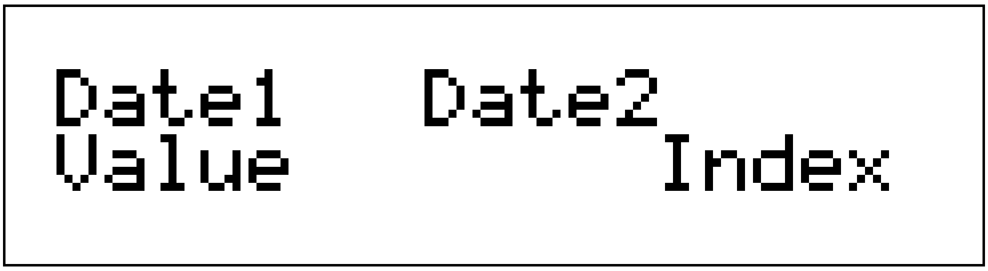
Feasts	DP_Feast[*,9]
Holidays	DP_Holidays

Holidays
Holidays matrix [5xn].

Fig. 17 – Example of **Holiday** module parametrization (matrix DP_Svatky has 10 columns)

The holiday matrix structure is described in Chapter 1. Day Plans. In the setting described above, the column No. 9 of the holiday matrix was selected as the column the module **DayPlan** uses to get information that it is holiday.

Entering holidays by means of the control system terminal is then done by means of the **Holiday** control and there are two ways to do it (simplified/extended).

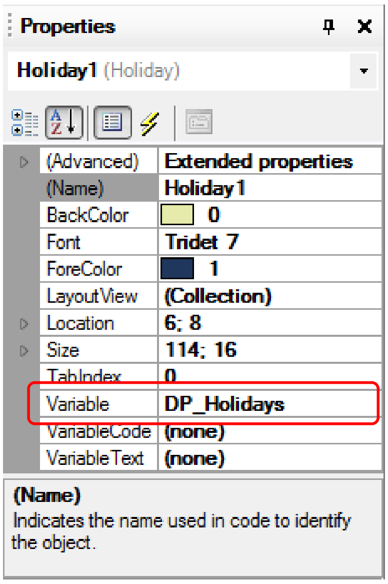


```
Date1   Date2
Value          Index
```

Fig. 18 – The **Holiday** control after being entered on the screen

3.1 Simplified mode

In the simplified mode, we only put the **Holiday** control on the screen and we enter the holiday matrix into the parameter **Variable**.



Properties

Holiday1 (Holiday)

(Advanced)	Extended properties
(Name)	Holiday1
BackColor	0
Font	Tridet 7
ForeColor	1
LayoutView	(Collection)
Location	6; 8
Size	114; 16
TabIndex	0
Variable	DP_Holidays
VariableCode	(none)
VariableText	(none)

(Name)
Indicates the name used in code to identify the object.

Fig. 19 – Example of parametrization of the **Holiday** control

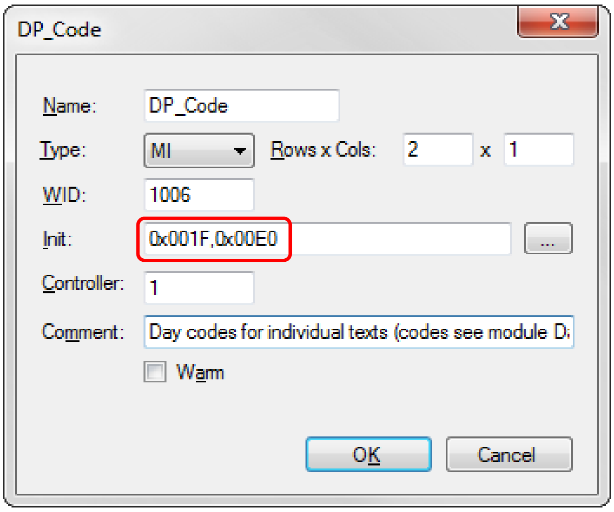
We select the date of holiday start and end on the control system terminal as well as the code of the day according to which the holiday is to be planned.

This mode can use multilingual texts (see AP0003 – Application Designed in Cyrillic).

3.2 Extended mode

If we work with day plan in extended mode (or we want to display other than default texts by means of the **Holiday** control) and we require that the **Holiday** control displays the same range of days as in the **TPlan** control, we must work with the **Holiday** control in the extended mode as well.

The extended mode of the **Holiday** control requires setting of the text matrix (see Chapter 1.2.1 Parametrization of the TPlan) and of the relevant day codes to individual plans



DP_Code

Name: DP_Code

Type: MI   Rows x Cols: 2 x 1

WID: 1006

Init: 0x001F,0x00E0

Controller: 1

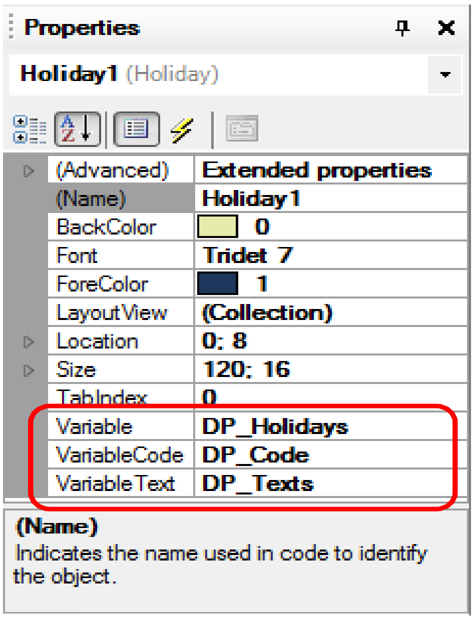
Comment: Day codes for individual texts (codes see module D

□ Warm

Fig. 20 – The definition of day codes matrix for individual plans

Attention
The day codes matrix must have the same number of lines as the text matrix for individual plans.

Text matrix will be assigned to the parameter **Advanced**/**VariableText**, the day codes matrix will be assigned to the parameter **Advanced**/**VariableCode** of the **Holiday** control.



Properties

Holiday1 (Holiday)

(Advanced)	**Extended properties**
(Name)	**Holiday1**
BackColor	□ **0**
Font	**Tridet 7**
ForeColor	■ **1**
LayoutView	**(Collection)**
Location	**0; 8**
Size	**120; 16**
TabIndex	**0**
Variable	**DP_Holidays**
VariableCode	**DP_Code**
VariableText	**DP_Texts**

(Name)
Indicates the name used in code to identify the object.

Fig. 21 – Example of parametrization of the **Holiday** control

Attention

When entering longer user texts for individual plan types, it is necessary to modify the position and dimensions of individual texts of the ***Holiday*** *control on the screen by means of its parameter* ***Advanced****/****LayoutView****.*

This mode does not allow the use of multilingual texts.

4 Day Plan Definitions

The order of day plan definitions:

- Define the time shift matrix.
- Define the matrix for time shift values (must have the same dimensions as the time shifts matrix).
- If the aforementioned matrices have the number of line lower than 8, define the text matrix and days code matrix (if you intend to use the controls **`Feast`** or **`Holiday`**). These must have the same number of lines as the time shift matrix.
- Define the day off matrix.
- Define the output variable from the day plan.
- Enter the **`DayPlan`** module into the periodic process, select its mode (parameter **`Mode`**) and enter the necessary matrices and the output variable for its parameters.
- Enter the required control for day plan parametrization on the screen (**`TPlan`**, **`GPlan`**, **`CPlan`**) and enter the necessary matrices for its parameters.
- If you intend to use day off planning, put the **`Feast`** control on the screen and enter the necessary matrices for its parameters.
- If you intend to use holiday planning, define the holiday matrix. Then put the **`Holiday`** control on the screen and enter the necessary matrices for its parameters.

5 Parametrization of day plans from PC

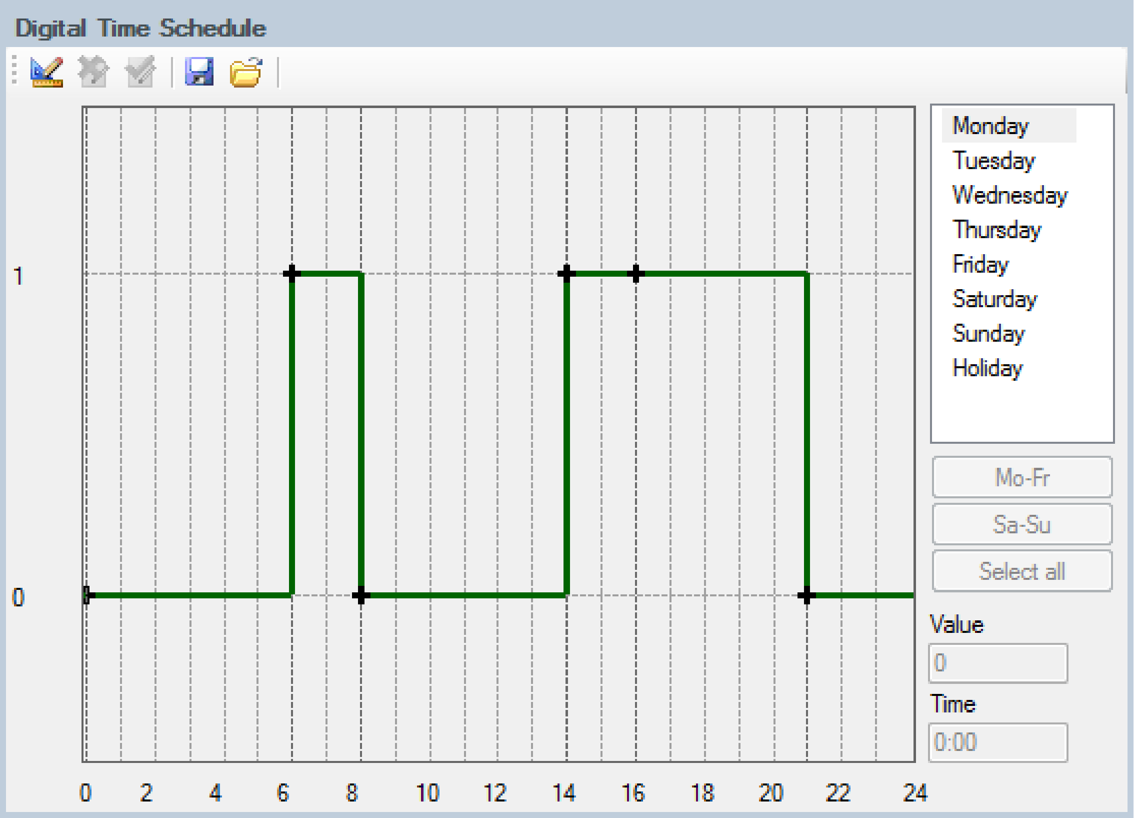
Parametrization of day plans in the control systems made by AMiT can not only be performed from the control system terminal/screen, but also from PC (using the ViewDet environment or various visualisation SW). If you use `GPlan` or `CPlan` controls to parametrize day plans in the control system, keep in mind that these controls always enter the value 00:00:00 of the first time shift into the day plan. When parametrizing day plans using visualisation SW, we need to make sure this time value cannot be changed. It would lead to discrepancies in the day plan setting from the control system and disrupted visualisation.

5.1 Day plans in ViewDet

The ViewDet environment includes a control for day plan parametrization in control systems made by AMiT. This control enables us to set day plans in various modes comfortably.

5.1.1 Digital Mode

The digital mode can be used to plan values Off/On (0/1). The time is displayed for individual days as a rectangular broken line that allows us to enter values 0/1 into the plan.



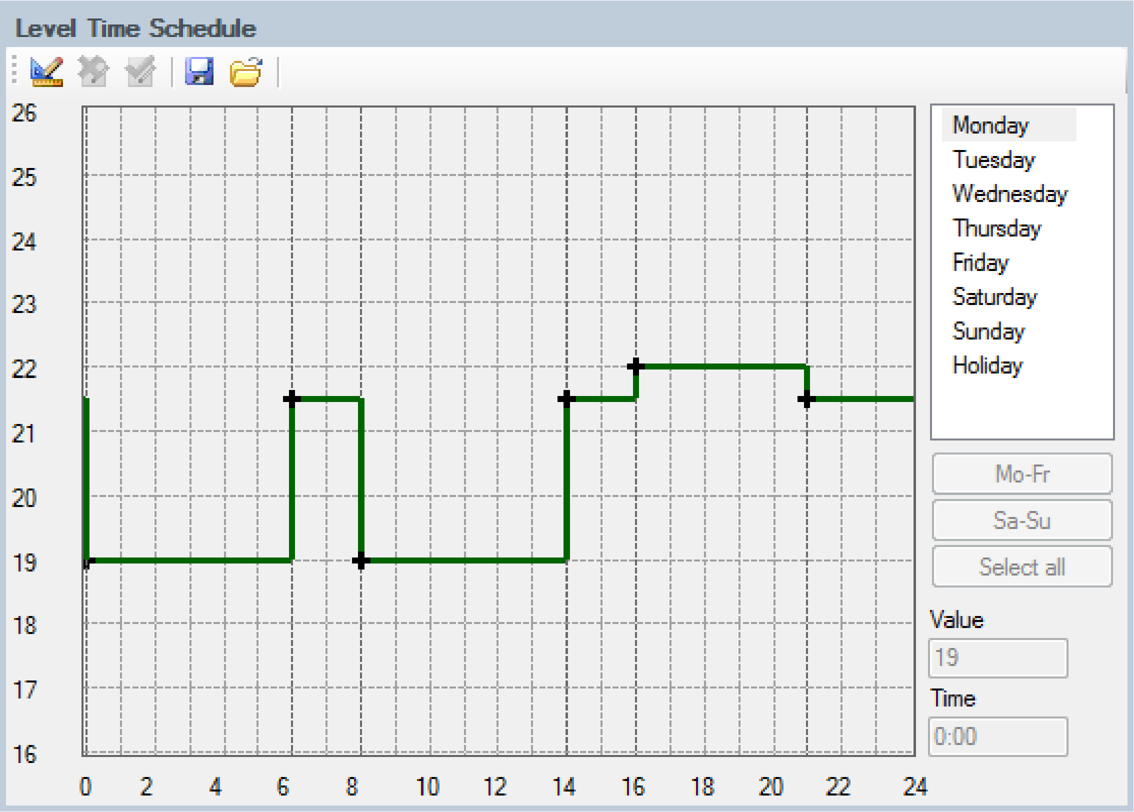
Value: 0

Time: 0:00

Fig. 22 – Entering a time schedule in the digital mode

5.1.2 Level mode

The level mode can be used for planning values on a specific required value in a selected period. Individual plans are displayed as a rectangular broken line that allows us to enter values of the plan within the range of values entered in the level day plan draft into parameters axis X – min and Axis Y – max.



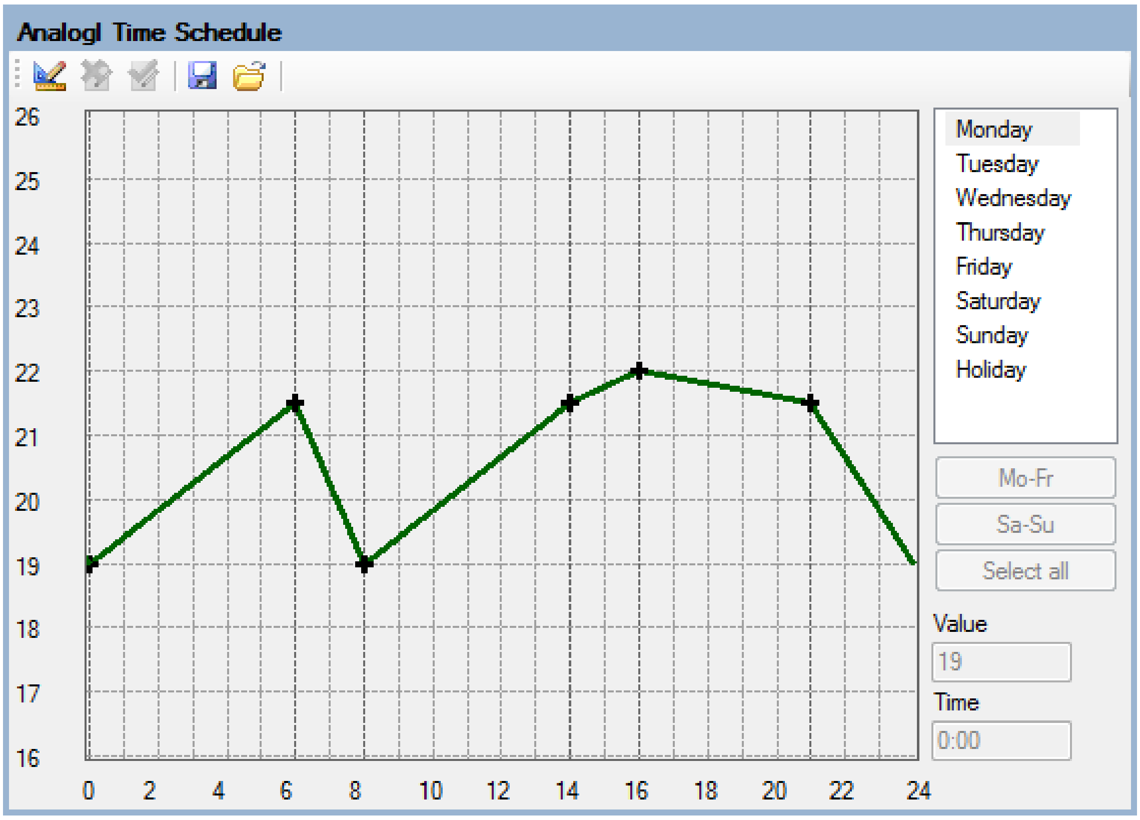
Value
19
Time
0:00

Fig. 23 – Entering a day plan in the level mode

The choice of display mode does not influence the actual behaviour of the day plan programmed by the **`DayPlan`** module in the programme part. If we want the day plan to be level, we must set this choice in the parameter **`Mode`** of the **`DayPlan`** module.

5.1.3 Analogue mode

The analogue mode can be used for planning values on a specific required value in a selected period. Individual plans are displayed as a broken line connecting individual shifts that allows us to enter values of the plan within the range of values entered in the level day plan draft into parameters axis X – min and Axis Y – max.



Value: 19

Time: 0:00

Fig. 24 – Entering a day plan in the analogue mode

Figure 24 shows the analogue mode of the same day plan that is depicted in Fig. 23 (day plan in the level mode).

The choice of display mode does not influence the actual behaviour of the day plan programmed by the **`DayPlan`** module in the programme part. If we want the day plan to be analogue, we must set this choice in the parameter **`Mode`** of the **`DayPlan`** module.

6 APPENDIX A

6.1 Working with various day plans on one screen

When using multiple various day plans, we can select certain parameters and programme the display/editing of day plans on one single screen. For that, we need:

- Putting all day plans into common matrices,
- using the parameter **`FromRow`** of modules **`DayPlan/DayPlan2`**,
- using the script on the screens.

6.1.1 Day plans in common matrices

We need to define all day plans by means of a single time matrix and a single values matrix. That means that in all modules **`DayPlan/DayPlan2`** matrices will be used with the same name and the same dimensions.

The number of lines **`n`** of both matrices must be higher than or equal to the required number of all plans entered in all modules **`DayPlan/DayPlan2.`**

The number of columns **`m`** of all matrices must be the maximum of the considered number of time shifts within the individual modules **`DayPlan/DayPlan2`**.

Example
There is a requirement for implementation of three day plans that are supposed to work in the simplified mode (the option of independent setting of the plan for each day of the week) with the use of days off. For the first time plan, it is sufficient to set 4 time shifts within the day, the other two require 6 time shifts.

The resulting matrix dimension is therefore:
n = 8 × 3 = 24 lines

The value 8 is given by the number of days of the week (7) increased by one line for day off plan. The value 4 is the number of day plans.

m = max(4, 6, 6) = 6 columns

6.1.2 Using the parameter FromRow of modules DayPlan/DayPlan2

The parameter **`FromRow`** us used to determine the line of the value matrix and time matrix, where dates for the relevant module **`DayPlan`**/**`DayPlan2`** start.

Example
When entering the example from Chapter 6.1.1, parametrization of modules ***`DayPlan`****, for instance, would be as follows:*

```
//Data for the value matrix and time matrix start on the line 0
DayPlan 0x0000, 8, 0x0001, 0x0002, 0x0004, 0x0008, 0x0010, 0x0020, 0x0040, 0x0080,
DP_Feast, DP_Times, DP_Values, DP_Out_0

//Data for the value matrix and time matrix start on the line 8
DayPlan 0x0800, 8, 0x0001, 0x0002, 0x0004, 0x0008, 0x0010, 0x0020, 0x0040, 0x0080,
DP_Feast, DP_Times, DP_Values, DP_Out_1

//Data for the value matrix and time matrix start on the line 16
```

```
DayPlan 0x1000, 8, 0x0001, 0x0002, 0x0004, 0x0008, 0x0010, 0x0020, 0x0040, 0x0080,
DP_Feast, DP_Times, DP_Values, DP_Out_2
```

The parameter related to the line number is highlighted.

6.1.3 Using the script on the screens

Selected controls for display/editing of day plans allow us to work with the parameter **FromRow**, same as modules **DayPlan**/**DayPlan2**. The parameter **FromRow** available on the screen by means of a script. It can be used e.g. in the event **OnOpen** of the screen on which the control for display/editing of day plans is placed. In this event, we choose what data is to be displayed at the moment, e.g. on the basis of a selected item (e.g. 0, 1 and 2) from the menu (that is placed on the screen) in the following manner:

```
GPlan1.FromRow = Menu.MenuScreen1.SelectedIndex * 8;
GPlan1.Refresh();
```

7 Technical Support

All information on the use of day plans will be provided by the technical support department of the company AMiT. Do not hesitate to contact the technical support via e-mail using the following address: **support@amit.cz**.

8 Warning

The company AMiT, spol. s r.o. does not provide any guarantee concerning the contents of this publication and reserves the right to change the document with no obligation to inform any person or authority.

This document can be copied and redistributed under following conditions:

1. The whole text (all pages) must be copied without making any modifications.
2. All copies must include the AMiT, spol. s r.o. copyright and any other notices stated in the document.
3. This document must not be distributed for profit.

The names of products and companies used herein may be trademarks orregistered trademarks of their respective owners.