

## PAT\_GEN.TDD

This driver is a Pattern Generator at any Port. The resolution is determined by the TIMERA setting.

### Installation of the driver

INSTALL DEVICE #D, "PAT\_GEN.TDD" [, P1, ..., P5]

D is a constant, variable or an expression of data type BYTE, WORD, LONG in range of 0...63 and is the device number of the driver.

P1...P5 are more parameters, which changes the settings of the PAT\_GEN.TDD driver.

	Default	Description of parameters
P1	-	Port address
P2	-	Is always 255
P3		Minimum is 2!!!
P3a	-	active period in Timer-A ticks (low byte)
P3b	-	active period in Timer-A ticks (high byte)
P4		0, 2...n 0: no inactive phase!!!
P4a	-	inactive period in Timer-A ticks (low byte)
P4b	-	inactive period in Timer-A ticks (high byte)
P5	-	Initial Bitmask Port Pins

The Device Driver PAT\_GEN.TDD outputs the values from a String as a pattern signal. Output is synchronized using driver 'TIMERA.TDD', yielding high performance independent of the BASIC program. The driver is configured during installation, subsequent changes can be made through User Function Codes.

The output-data is contained byte-wise in the string.

The output-string must exist at all times ! Transient variables (e.g. local strings in sub-routines or temporary strings (expressions) must NOT be used. **Instead, use Global strings or strings local to the Task.**

## PAT\_GEN.TDD

The string is loaded into the PWM Driver using the PUT command with secondary address 0. The bytes appear in the selected output-speed.

**NOTE:** The output-port is determined by a parameter during driver-installation, not by the secondary address!

PAT\_GEN.TDD has the functionality of a Reload-Buffer, if output started yet. The reloaded String will be automatically output as soon as no other data is available for output. This ensures seamless transfer from one data-string to the next. The transfer from the Reload-Buffer into the output buffer happens at the instant that the output-buffer is empty.

Using User-Function-Code UFCI\_PG\_RELOAD, the BASIC-Program can query if the transfer already happened, and if the reload-buffer is available. If this is the case, the next data-string can be written to Reload Buffer.

## Secondary addresses

Write an option to the driver is possible to different secondary addresses:

Secondary address	Function	Instruction
0	defines Output Data and starts new Output	PUT

## User Function Codes

User-Function-Codes of PAT\_GEN.TDD to read out parameters (Instruction GET, secondary address 0):

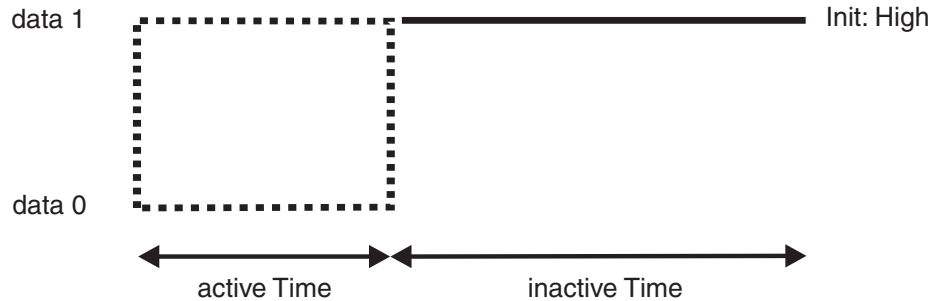
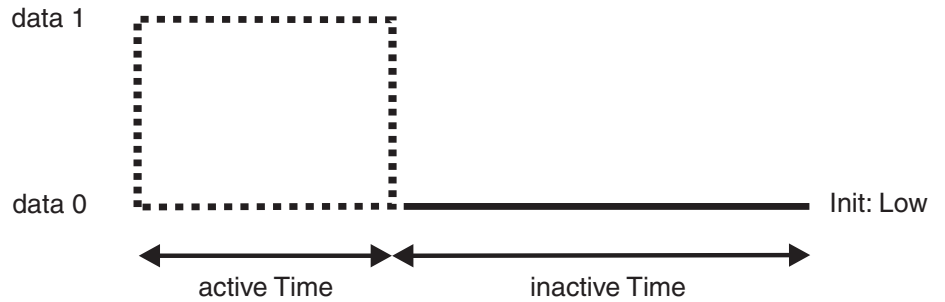
No.	Symbol Präfix UFCI_	Description
82H	PG_ACTIVE_PERIOD	Reads out active period in Timer-A ticks
83H	PG_INACTIVE_PERIOD	Reads out active period in Timer-A ticks 0: no inactive phase!!!
84H	PG_INITIAL	Reads out Initial Bitmask Port Pins
90H	PG_RESTLEN	No. of Bytes to Output in actual String
91H	PG_ACT	Output active 0: Output is taking place non-0: idle
92H	PG_RELOAD	Reload Buffer state 0: empty non-0: contains data
93H	PG_REPEAT	No. of remaining repetitions. Zero = infinite repetitions.

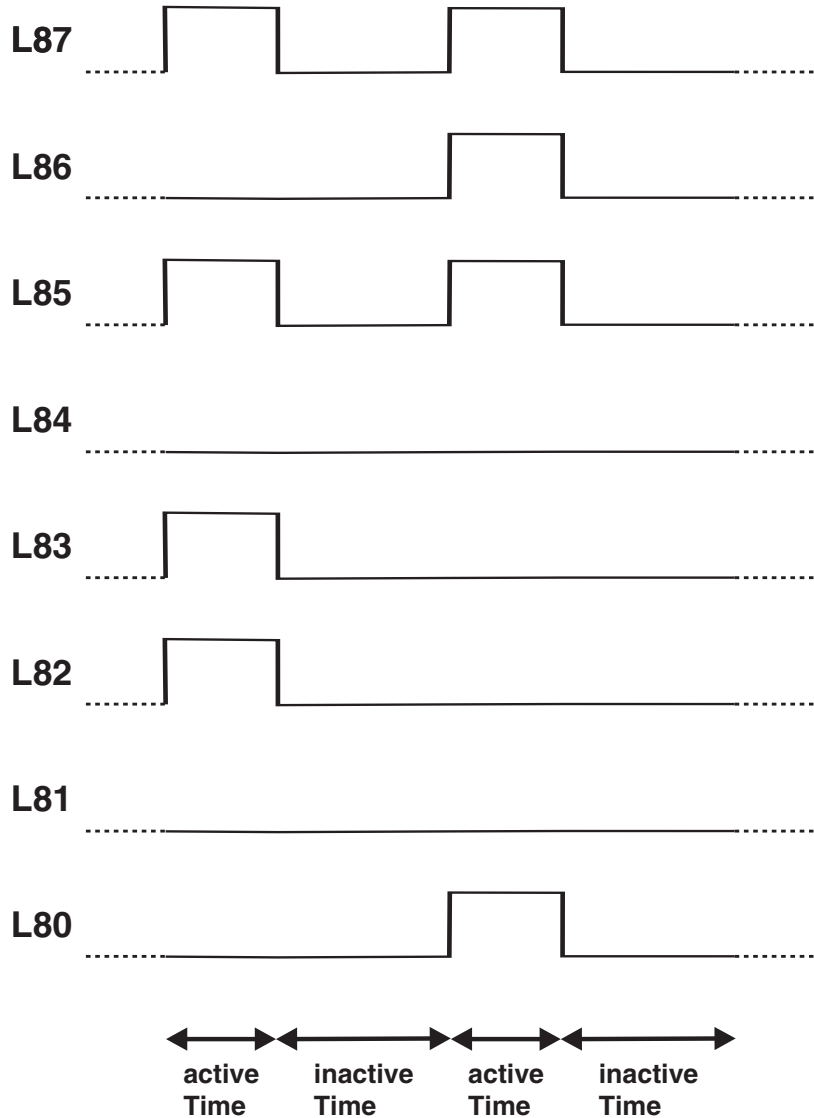
User-Function-Codes of PAT\_GEN.TDD to set parameters (Instruction PUT, secondary address 0):

No.	Symbol Präfix: UFCO_	Description
80H	PG_STOP	Stops Output
81H	PG_CONT	Continues Output
82H	PG_ACTIVE_PERIOD	Sets active period in Timer-A ticks (2...65535); Driver must be idle to set active period!
83H	PG_INACTIVE_PERIOD	Sets active period in Timer-A ticks (0, 2...65535) 0: no inactive phase!!! Driver must be idle to set inactive period!
84H	PG_INITIAL	Sets Initial Bitmask Port Pins; Driver must be idle to set initial state!

## Pattern Generation with active and inactive Period

If output data is 1, the initial state of the pin is toggled for the active period, otherwise the pin stays in initial state. During the inactive period all pins are in initial state.



**Port: 8****Output String: "<0ACH><0E1H>"****Binary: <10101100B><11100001B>****active & inactive Phase****Initial: 00000000B**

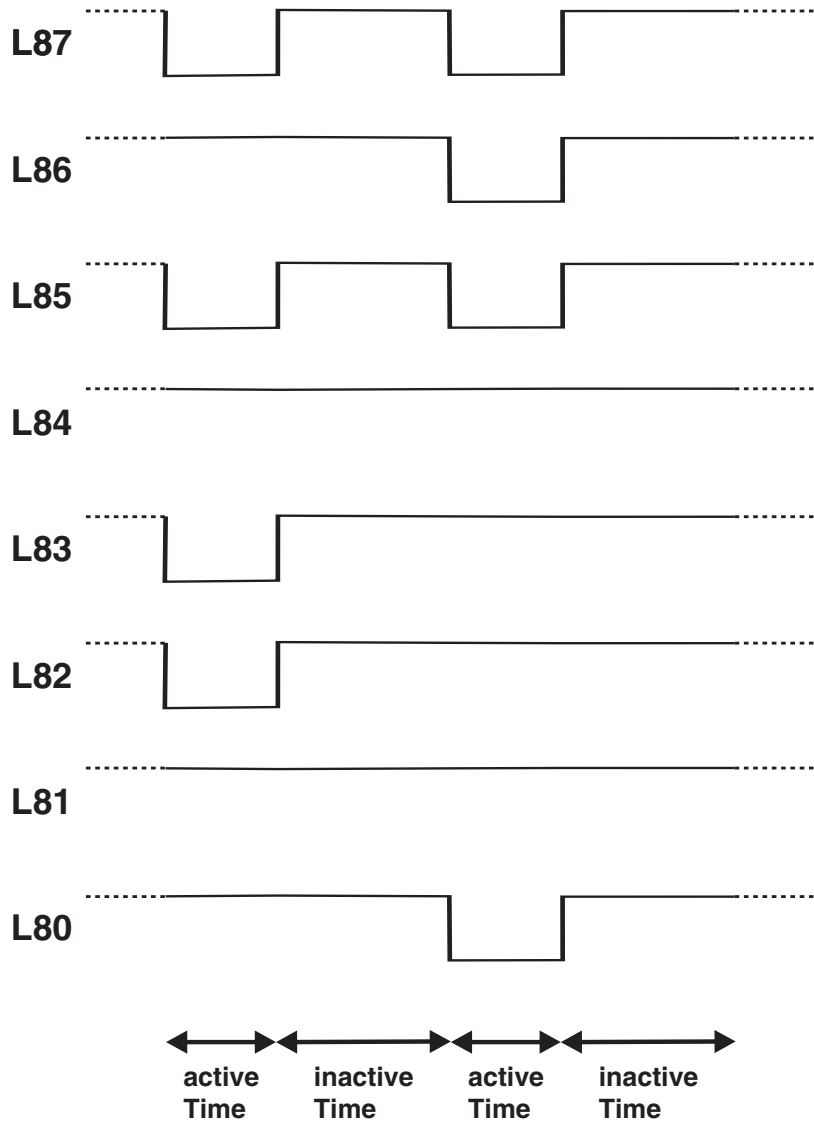
**Port: 8**

**Output String: "<0ACH><0E1H>"**

**Binary: <10101100B><11100001B>**

**active & inactive Phase**

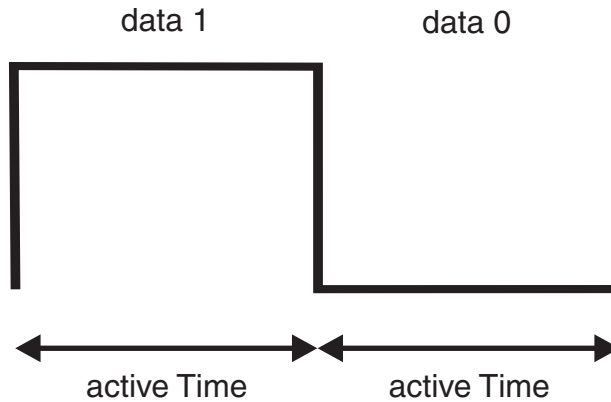
**Initial: 11111111B**



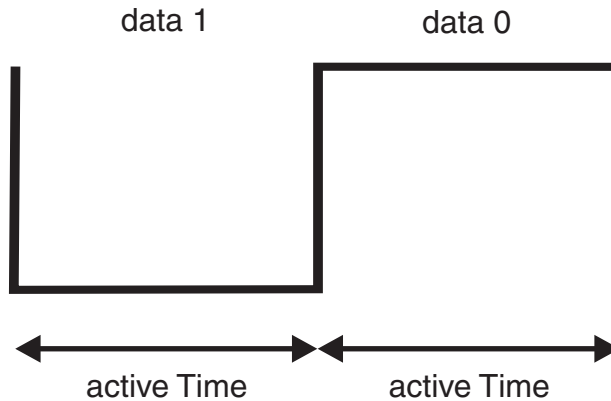
## Pattern Generation with active period only

If no inactive period is chosen, data is XORed with the initial state. The next byte directly follows after the active period without any pause. Constant levels can be generated this way. PWM output is also possible at a complete Port or X-Port.

**Initial: 0**



**Initial: 1**





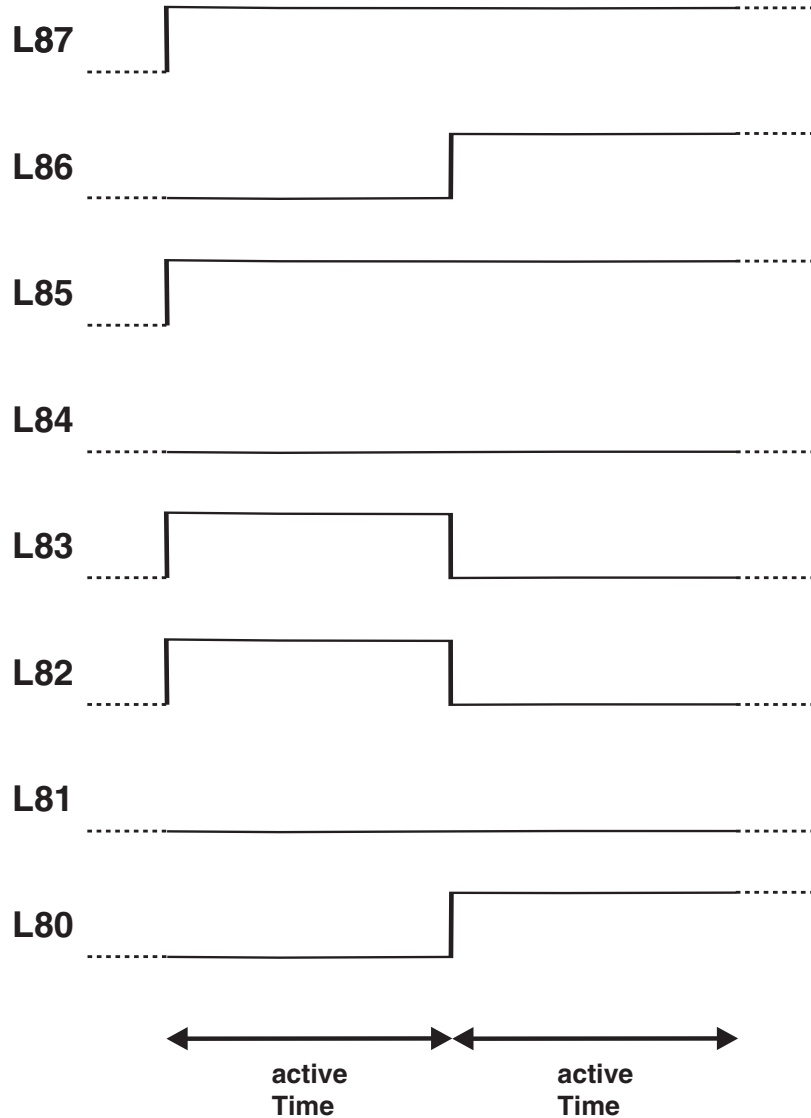
Port: 8

Output String: "<0ACH><0E1H>"

Binary: <10101100B><11100001B>

active Phase only

Initial: 00000000B



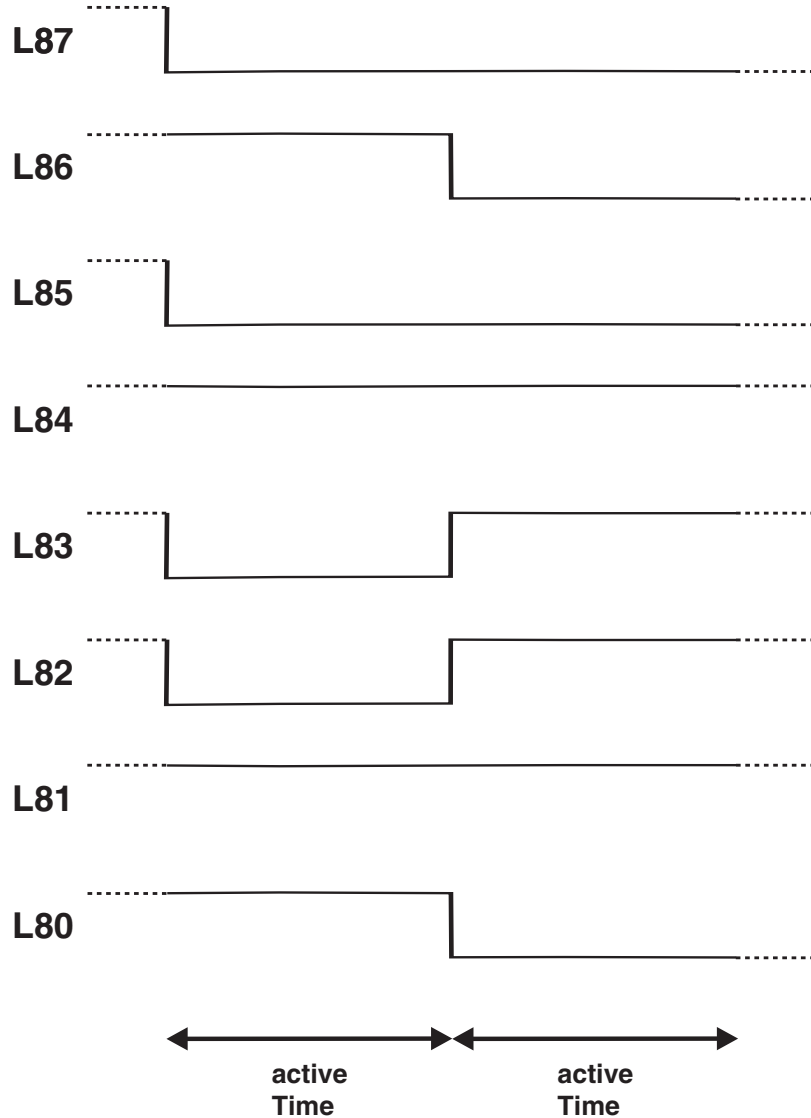
**Port: 8**

**Output String: "<0ACH><0E1H>"**

**Binary: <10101100B><11100001B>**

**active Phase only**

**Initial: 11111111B**



## Starting the Pattern Generator

Output for strings is started using the following command.

**PUT #D, STR\$ [, Offs, No, Rpt]**

<b>D</b>	is a constant, a variable or expression of the data type BYTE, WORD, LONG in the range from 0→63 and stands for the device number of the drivers.
<b>STR\$</b>	is the data-string. <b>STR\$ must be static, i.e. global or local to task.</b>
<b>Offs</b>	is a constant, a variable or expression of the data type BYTE, WORD, LONG and determines the offset into the string from which data is to be output. (Default is 0)
<b>No</b>	is a constant, a variable or expression of the data type BYTE, WORD, LONG and determines the number of Bytes that is to be output. If zero is used, data is output until the end of string.
<b>Rpt</b>	is a constant, a variable or expression of the data type BYTE, WORD, LONG and determines the number of repetitions (0...255). Zero = infinite repetitions.

If output is started yet and not finished and the Reload buffer is empty, the next string is loaded into the Reload Buffer. This ensures seamless transfer from one data-string to the next. The transfer from the Reload-Buffer into the output buffer happens at the instant that the output-buffer is empty. If Reload Buffer is not empty, the string will be ignored

Output for flash data is started using the following command.

**PUT #D, Pos, Offs, No [, Rpt]**

<b>D</b>	is a constant, a variable or expression of the data type BYTE, WORD, LONG in the range from 0→63 and stands for the device number of the drivers.
<b>Pos</b>	address in flash memory, where output data is saved.
<b>Offs</b>	is a constant, a variable or expression of the data type BYTE, WORD, LONG and determines the offset into the string from which data is to be output. (Default is 0)
<b>No</b>	is a constant, a variable or expression of the data type BYTE, WORD, LONG and determines the number of Bytes that is to be output. If zero is used, data is output until the end of string.
<b>Rpt</b>	is a constant, a variable or expression of the data type BYTE, WORD, LONG and determines the number of repetitions (0...255). Zero = infinite repetitions.

If output is started yet and not finished and the Reload buffer is empty, the next string is loaded into the Reload Buffer. This ensures seamless transfer from one data-string to the next. The transfer from the Reload-Buffer into the output buffer happens at the instant that the output-buffer is empty. If Reload Buffer is not empty, the data will be ignored.

## Checking activity of driver

### GET #D, #0, # PG\_ACT, Number, Variable

**D** is a constant, a variable or expression of the data type BYTE, WORD, LONG in the range from 0→63 and stands for the device number of the drivers.

**Number** is a constant, a variable or expression of the data type BYTE, WORD, LONG and specifies the length of output.

**Variable** is a variable of the data type BYTE, WORD, LONG or STRING. If result is 0, output is taking place, otherwise the driver is idle

### GET #D, #0, # PG\_RESTLEN, Number, Variable

**D** is a constant, a variable or expression of the data type BYTE, WORD, LONG in the range from 0→63 and stands for the device number of the drivers.

**Number** is a constant, a variable or expression of the data type BYTE, WORD, LONG and specifies the length of output.

**Variable** is a variable of the data type BYTE, WORD, LONG or STRING. If result is 0, the driver is idle, otherwise the number specifies the number of Bytes to Output in actual String (Reload string is NOT included)

## **Stop and continue output**

### **PUT #D, #0, # PG\_STOP, dummy**

**D** is a constant, a variable or expression of the data type BYTE, WORD, LONG in the range from 0→63 and stands for the device number of the drivers.

**dummy** is a constant, a variable or expression of the data type BYTE, WORD or LONG. This is only a dummy.

This command stops output of the driver PAT\_GEN.TDD. The actual Byte is written complete to Port, then the output stops. The driver is idle now. Afterwards a complete output can be started with secondary address 0 or the actual output can be continued with USER FUNCTION CODE PG\_CONT while no new output is started.

### **PUT #D, #0, # PG\_CONT, dummy**

**D** is a constant, a variable or expression of the data type BYTE, WORD, LONG in the range from 0→63 and stands for the device number of the drivers.

**dummy** is a constant, a variable or expression of the data type BYTE, WORD or LONG. This is only a dummy.

An Output can only be continued, if it was stopped before and no other Output started yet. The Output continues exactly, where it stopped.