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 that determine 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 bit mask for 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 subroutines or temporary strings (expressions) must NOT be used. **Instead, use global strings or strings local to the task**.

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 instantly when 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 the reload buffer.

Secondary addresses

Writing to the driver is possible with 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 for requesting parameters (Instruction GET, secondary address 0):

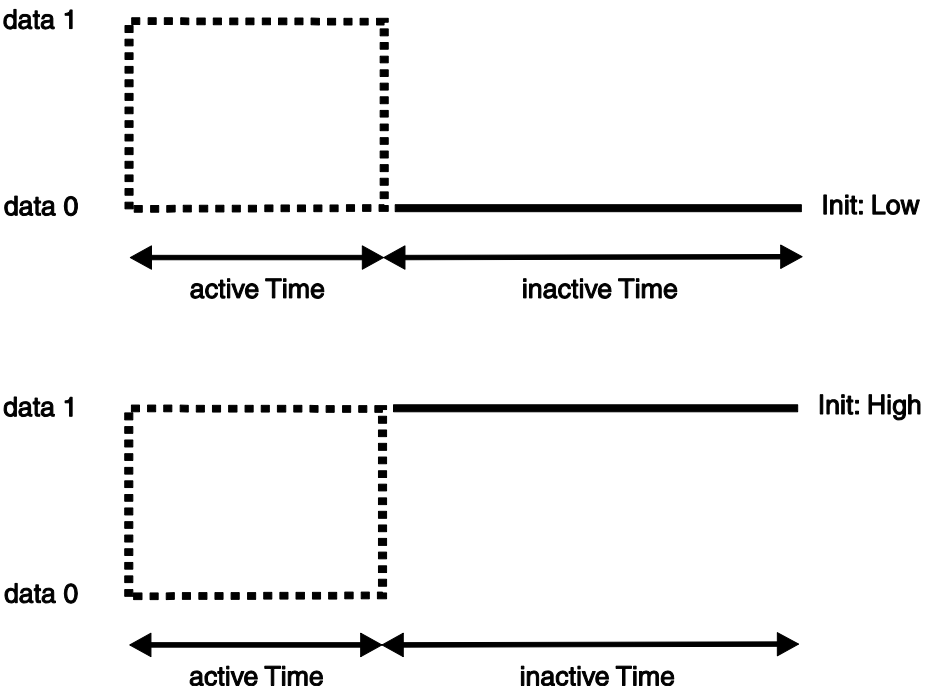
No.	Symbol Prefix UFCI_	Description
82H	PG_ACTIVE_PERIOD	Reads out active period in Timer-A ticks.
83H	PG_INACTIVE_PERIOD	Reads out active period in Timer-A ticks. 0: no inactive phase!!!
84H	PG_INITIAL	Reads out initial bit mask for port pins.
90H	PG_RESTLEN	No. of bytes to output in actual string.
91H	PG_ACT	Output active 0: Output is taking place ‹›0: idle
92H	PG_RELOAD	Reload buffer state 0: empty ‹›0: contains data
93H	PG_REPEAT	No. of remaining repetitions. 0 = infinite repetitions.

User-Function-Codes of PAT_GEN.TDD for setting of parameters (Instruction PUT, secondary address 0):

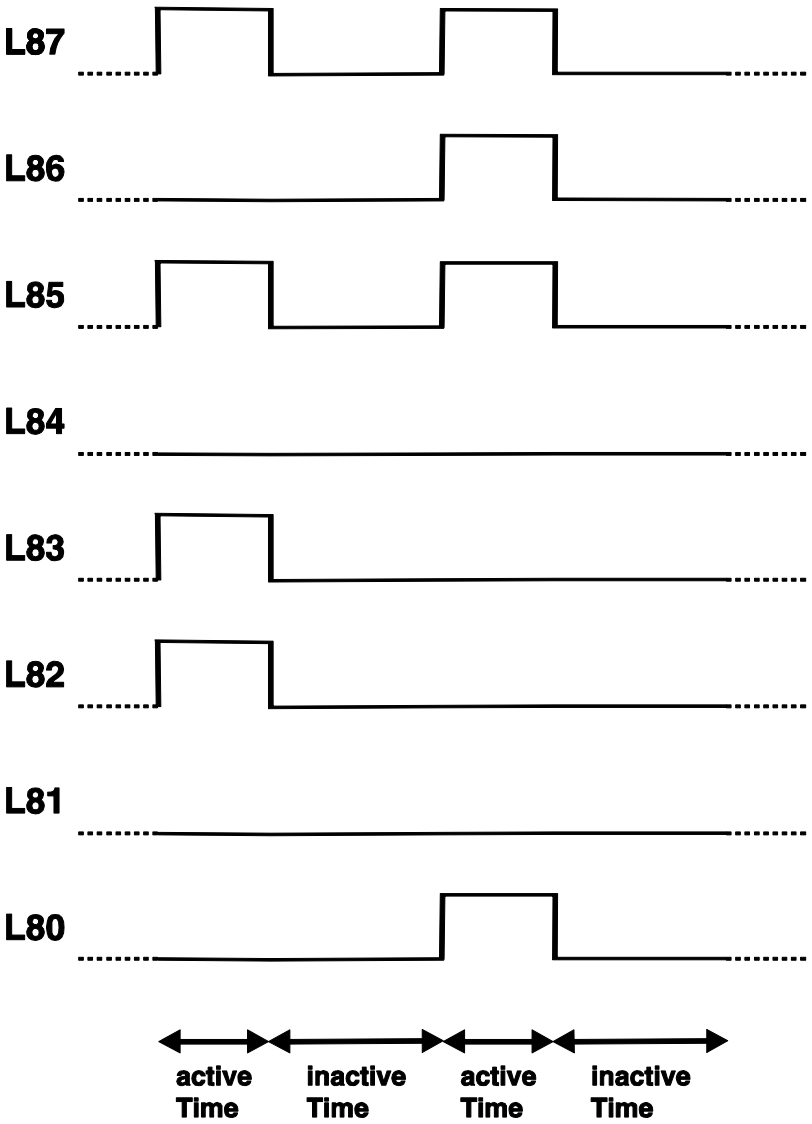
No.	Symbol Prefix: UFCO_	Description
80H	PG_STOP	Stops output
81H	PG_CONT	Continues output
82H	PG_ACTIVE_PERIOD	Sets active period in Timer-A ticks (2...65535). Driver must be idle to set active period!
83H	PG_INACTIVE_PERIOD	Sets active period in Timer-A ticks (0, 2...65535). 0: no inactive phase!!! Driver must be idle to set inactive period!
84H	PG_INITIAL	Sets initial bit mask for port pins. Driver must be idle to set initial state!

Pattern Generation with active and inactive Period

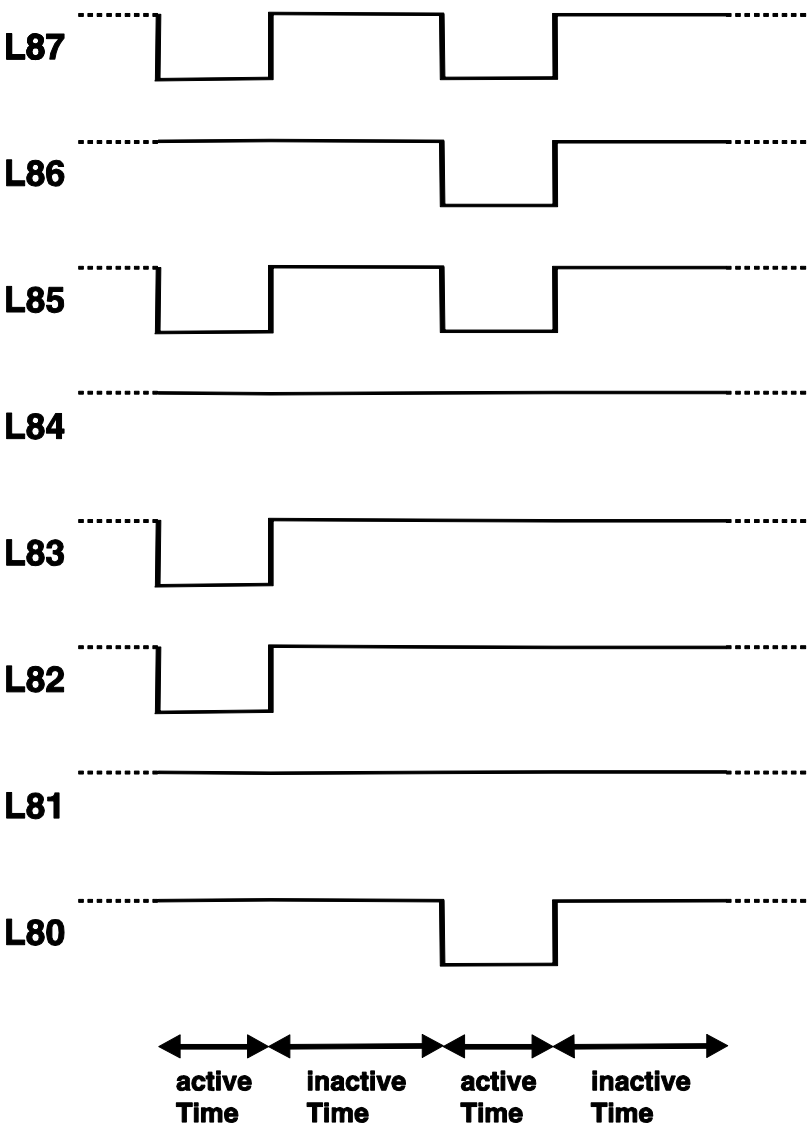
If output data is 1, the initial state of the pin is toggled for the active period, otherwise the pin stays in initial state. During the inactive period all pins are in initial state.



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

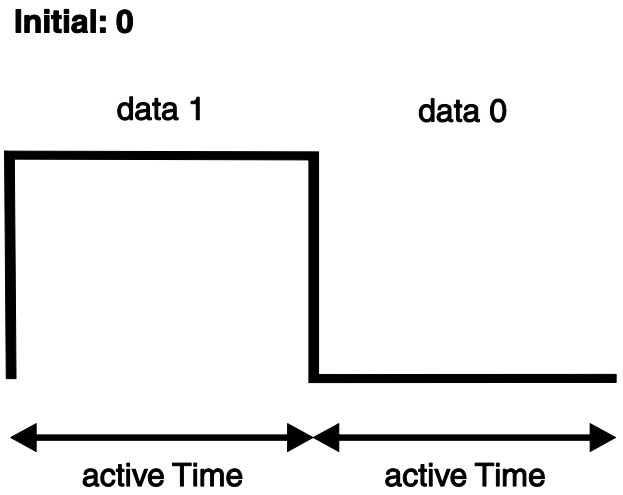


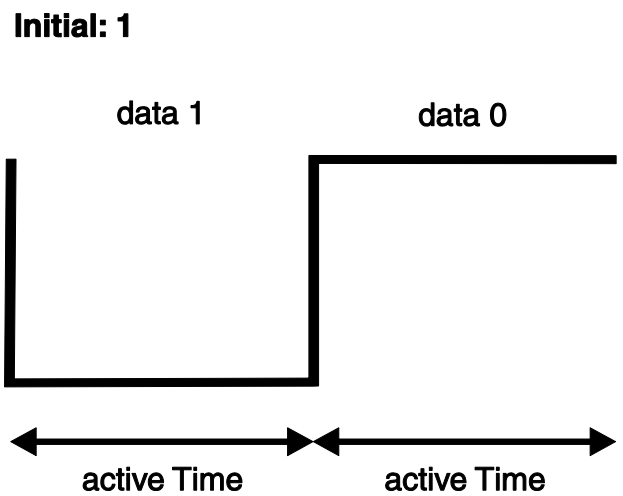
Port: 8
Output String: “<0ACH><0E1H>”
Binary: <10101100B><11100001B>
active & inactive Phase
Initial: 11111111B



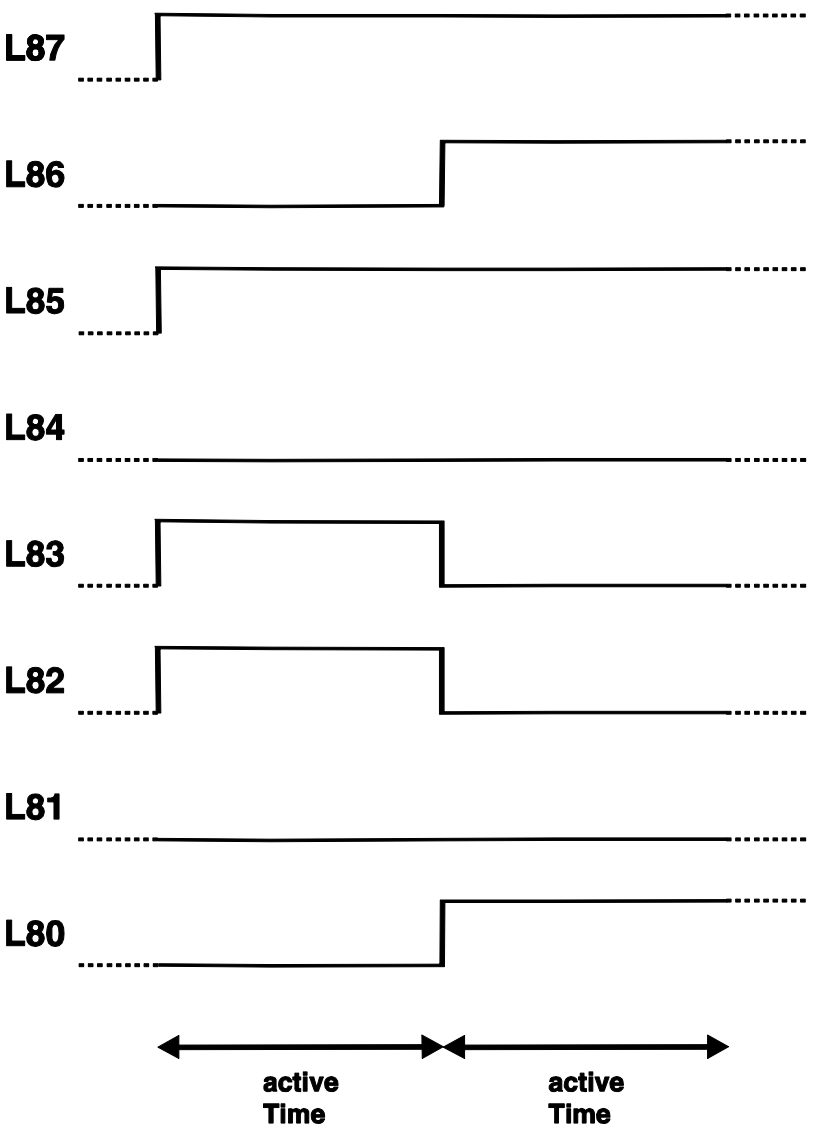
Pattern Generation with active period only

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.

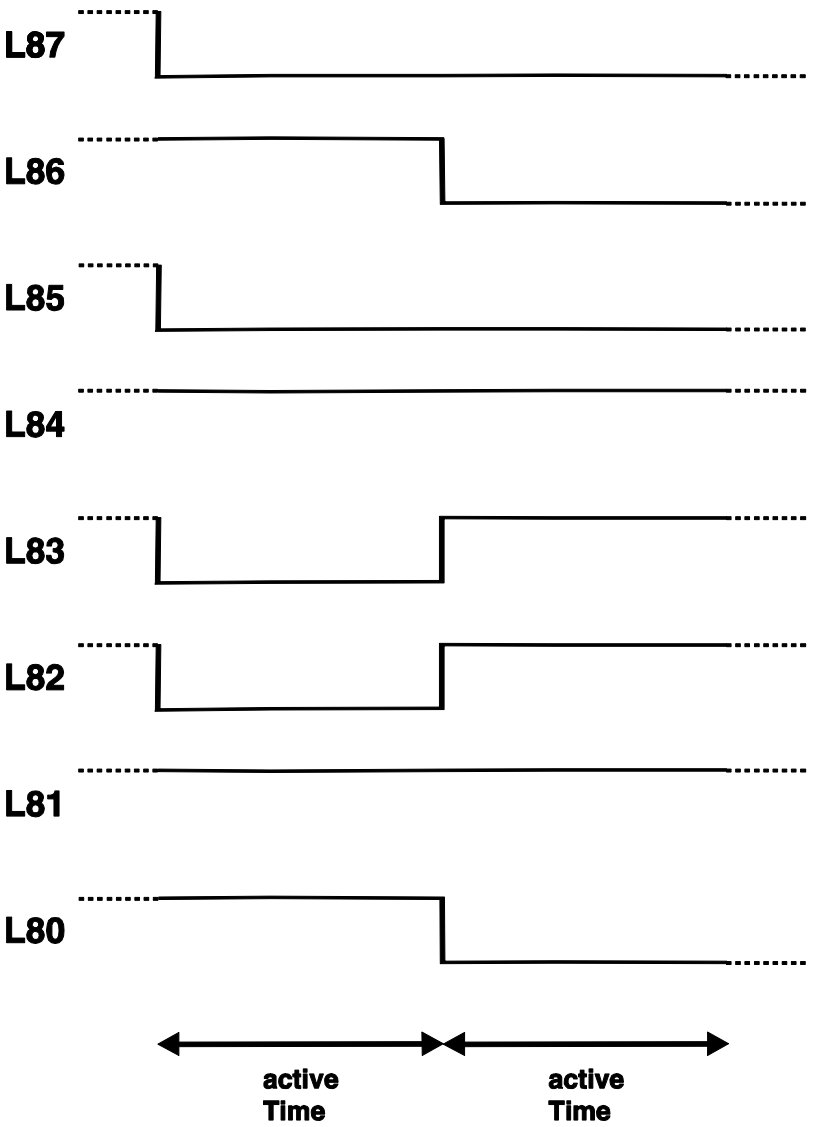




Port: 8
Output String: “<0ACH><0E1H>”
Binary: <10101100B><11100001B>
active Phase only
Initial: 00000000B



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]

D	is a constant, a variable or expression of the data type BYTE, WORD, LONG in the range from 0...63 and determines the device number of the driver.
STR$	is the data-string. **STR$ must be static, i.e. global or local to task.**
Offs	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).
No	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.
Rpt	is a constant, a variable or expression of the data type BYTE, WORD, LONG and determines the number of repetitions (0...255). 0 = 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 instantly when the output buffer is empty. If the reload buffer is not empty, the string will be ignored.

Output of flash data is started using the following command:

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

D	is a constant, a variable or expression of the data type BYTE, WORD, LONG in the range from 0...63 and determines the device number of the driver.
Pos	is the start address in flash memory, where output data is saved.
Offs	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).
No	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.
Rpt	is a constant, a variable or expression of the data type BYTE, WORD, LONG and determines the number of repetitions (0...255). 0 = 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 instantly when the output buffer is empty. If the 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 determines the device number of the driver.

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 determines the device number of the driver.

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 determines the device number of the driver.

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 completely to the port, after that 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 determines the device number of the driver.

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 was stopped.