Measure pulse lengths with TIMERA

This device driver measures the pulse length (high and low) at a pin. The file name given during installation of the driver specifies the pins at which the pulse length measurement is to take place. The resolution is determined by the TIMERA setting.

Installation of the driver

INSTALL DEVICE #*D*, "PLSI2_Pp.TDD"[, *P1, P2, P3*]

D	is a constant, variable or expression of the data type BYTE, WORD, LONG in the range from 0...63 and stands for the device number of the driver.
Pp	in the file name stands for: P: internal port p: measuring pin.
P1	is a parameter which sets the measurement to 16 or 32 bit 32: sets a 32-bit measurement any other or smaller value sets a 16-bit measurement.
P2	is a parameter which automatically extends the sign from WORD to LONG in a 16-bit measurement. 0: sign is extended 1: sign not extended
P3	is an optional parameter to determine the length of the brief starting pulse (valid values Tiger-1: 1...10; Tiger-2: 1...80)(leave unchanged: 0) (Default: 3@T1, 24@T2 = 5µs)

Secondary addresses

Reading out the results is possible from different secondary addresses:

Secondary address	Function	Instruction
0	Starts / Stops the measurement	PUT
1	Starts the measurement with an additional brief starting pulse	PUT

The measurement is started by transferring a value to the driver with a PUT instruction. The transferred value determines the action of the driver:

Value	Read operation
0	Stops the measurement
1	Starts the measurement with the next flank
2	Starts the measurement with the next rising flank
3	Starts the measurement with the next falling flank

Once the measurement has started PLSI2 waits for the matching flank in the cycled of the TIMERA ticks. Once the flank has arrived the measurement is carried out in time units specified by TIMERA. The 'high' part of the pulse is saved as positive number in the buffer, the 'low' part as a negative number. The measurement is stopped when the buffer is full or when a stop command is sent.

A 16-bit measurement has certain advantages:

- Lower load on the CPU.
- The buffer can hold more measured values.

Since WORD variables have no sign the measured values should be read out with a LONG variable. The driver automatically adds the appropriate sign from WORD to LONG. If the measured value were to be read out with WORD variables or the automatic sign extension deactivated all measured values for the 'low' part of the pulse 65536 would be minus the measured time.

! Note: If the TIMERA frequency is altered during measurements this produces values which are no longer reconstructable.

User Function Codes

User-Function-Codes of PLSI2_Pp.TDD for requesting parameters (Instruction GET, secondary address 0):

No	Symbol Prefix: UFCI_	Description
1	UFCI_IBU_FILL	Capacity of input buffer (Byte)
2	UFCI_IBU_FREE	Free space in input buffer (Byte)
3	UFCI_IBU_VOL	Size of input buffer (Byte)
65	UFCI_LAST_ERRC	Last Error-Code
99	UFCI_DEV_VERS	Driver version

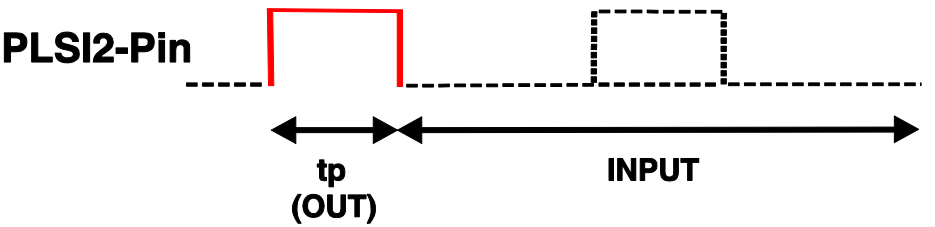
User-Function-Codes of PLSI2_Pp.TDD for setting of parameters (Instruction PUT, secondary address 0):

No	Symbol Prefix: UFCO_	Description
1	UFCO_IBU_ERASE	Delete input buffer
128	UFCO_PLS_SIGN	0: 16-bit values are evaluated with sign (-32767...+32768) 1: 16-bit values are evaluated without sign (0...65535)
133	UFCO_PLS_STOP	Stops the measurement

Generate brief starting pulse

For some devices it is necessary to pulse the pin as a short starting signal. Please start the pulse length measurement via secondary address 1 to pulse briefly. The length of the starting pulse can be adjusted with parameter P3 in the install device. Directly after calling the PUT procedure, the PLSI2 pin is switched to output and the pulse is generated. Immediately after the starting pulse, the PLSI2 line is switched to input again and the pulse length measurement starts as usual.

IMPORTANT: The PLSI2 line is switched to OUTPUT while sending the pulse. Consider this in your circuit!



	P3 = 1	P3 = 3	P3 = 10	P3 = 24	P3 = 80
tp (@T1)	3,3µs	5µs	14,8µs	n.a.	n.a.
tp (@T2)	590ns	870ns	2,4µs	5µs	16,2µs

Starts pulse length measurement including brief starting pulse:

```
PUT #D, #1, 2                          'start at next rising edge
```

Program example:

```
'-----------------------------------------------------------------
'Name: PLSI2.TIG
'-----------------------------------------------------------------
user_var_strict
#include define_a.inc
#include UFUNC3.INC
TASK Main                                'begin task MAIN
LONG FILLING, F
'install LCD-driver (BASIC-Tiger)
  INSTALL DEVICE #1, "LCD1.TDD"
'install LCD-driver (TINY-Tiger)
'INSTALL DEVICE #1, "LCD1.TDD", 0, 0, 0, 0, 0, 0, 80h, 8
  INSTALL_DEVICE #2, "TIMERA.TDD",1,250 'time base 10kHz
  INSTALL_DEVICE #3, "PLSI2_80.TDD",0   'install pulse length measurement
  INSTALL DEVICE #4, "SER1B_K1.TDD", BD_19_200, DP_8N, YES, BD_19_200,
DP_8N, YES
  USING "UD<8><1>   3,3,3,3.3"          'set decimal point at /1000

  PRINT #1, "<1>pulses on L80";
  run_task disp
'0 = stop measurement immediately
'1 = start measurement with next edge
'2 = start measurement with next rising edge
'3 = start measurement with next falling edge
  PUT #3, 2                              'start at next edge
  LOOP 999999999                         'many loops
    GET #3,#0,#UFCI_IBU_FILL,0,FILLING  'if results are in the buffer
    IF FILLING > 1 THEN
      GET #3, #0, 4, F                   'read result in mHz
      F = ABS(F)
      PRINT_USING #1, "<1BH>A<0><2><0F0H>pls10 L80:";F;
      PRINT_USING #4, "pls10:";F;"<9>";
      GET #3, #0, 4, F                   'read result in mHz
      F = ABS(F)
      PRINT_USING #1, "<1BH>A<0><3><0F0H>pls11 L80:";F;
      PRINT_USING #4, "pls11:";F
      PUT #3, #0, #UFCO_IBU_ERASE, 0     'ase buffer
      PUT #3, 2                          'ing edge
      WAIT_DURATION 10
    ENDIF
  ENDLOOP
END                                      'end task MAIN

TASK disp
  BYTE i
  LONG f

  for i = 0 to 0 step 0
    get #3, #0, #UFCI_IBU_FILL, 0, f
    print #1, "<1bh>A<0><1><0f0h>fill: ";f;"  ";
    wait_duration 100
  next
END
```