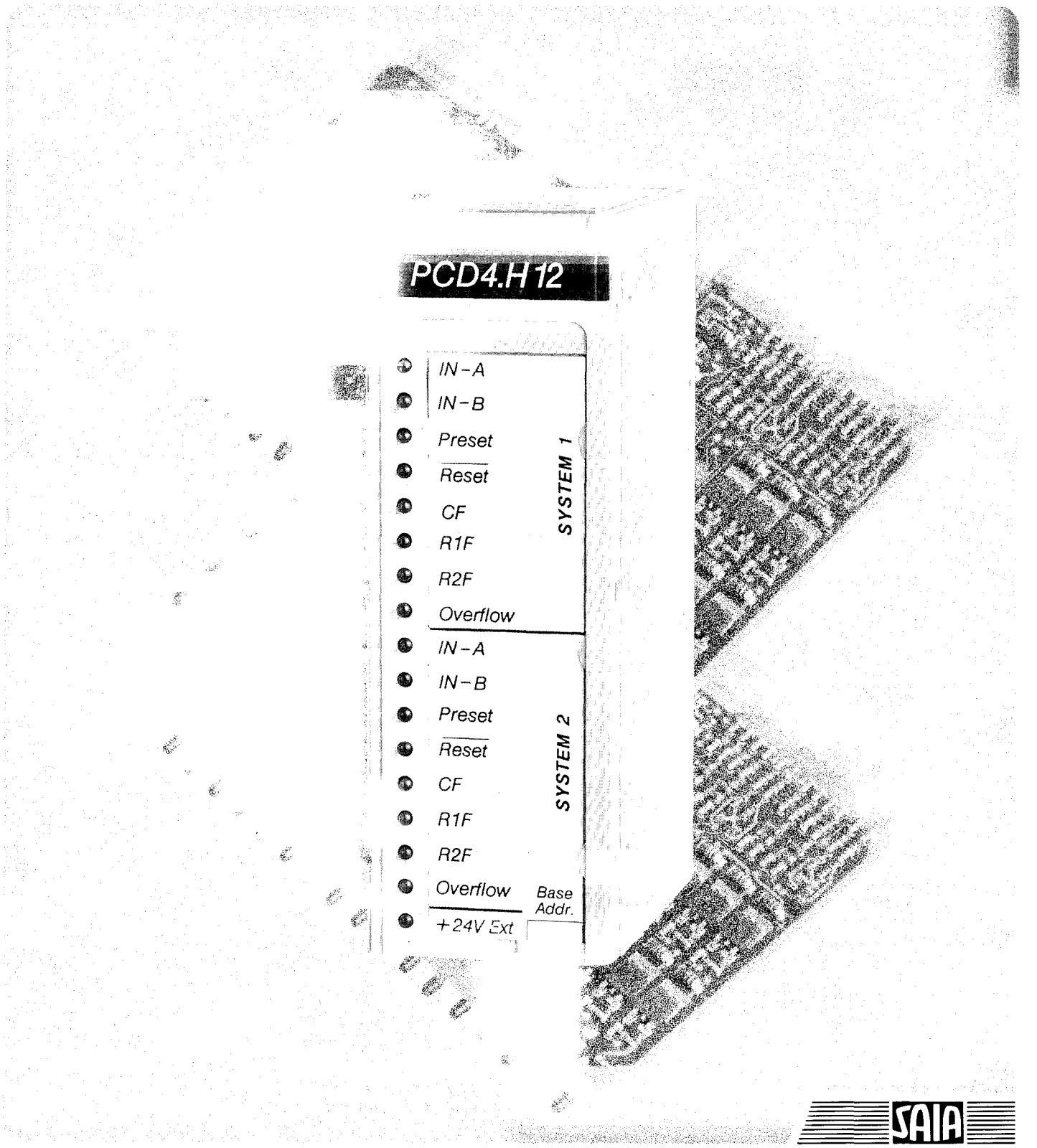
SAIA® PLC
Programmable controllers

Manual
General-purpose counting and measuring module
PCD 4.H 120



English edition 26/731 E1

SAIA® Programmable Controllers

General-Purpose Counting and Measuring Module

PCD4.H1..

Edition 26/731 E1
Issue 06/92

Subject to technical changes

Price SFr. 80.-

Contents

8. User program

Please note:

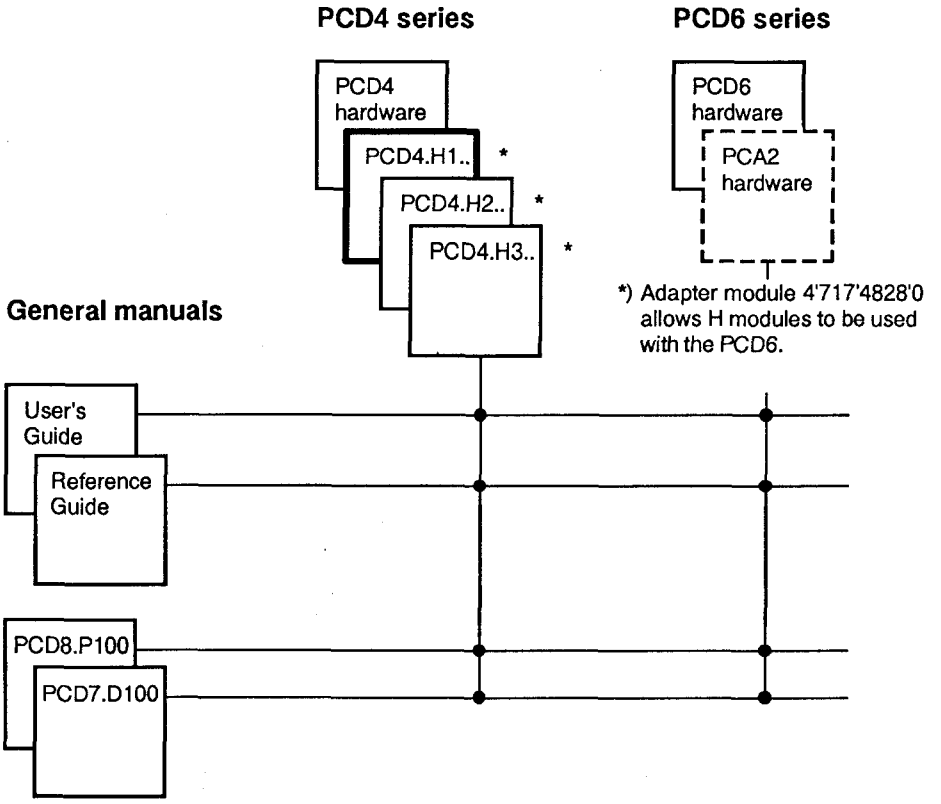
A number of detailed manuals are available to aid installation and operation of the SAIA PLC. These are for use by technically qualified staff, who may also have successfully completed one of our "workshops".

Each SAIA PLC is a high quality electronic device which has been subjected to strict quality control throughout all stages of its development, component selection and evaluation, and manufacture. Each has also passed stringent in-circuit and burn-in tests, and complete functional checks.

To obtain the best performance from your SAIA PLC, closely follow the guidelines for assembly, wiring, programming and commissioning given in these manuals. In this way, you will also become one of the many enthusiastic SAIA PLC users.

If you have any technical suggestions or recommendations for improvements to the manuals, please let us know. A form is provided on the last page of this manual for your comments.

Summary



1. Introduction

Functionality and applications

The maximum frequency for signals read via standard input modules is reached at about 50Hz. This is a physical limit imposed by the interference filters on the inputs (typically 8ms). Additionally, the CPU's cycle time, the length of the user program and the programming technique used, all affect the frequency limit. In order to process rapid sequences of pulses, with frequencies up to 166kHz, the PCD4/6.H1.. module should be used.

The PCD4.H120 has been designed primarily as a counting module. However, as shown in the following list, it can also measure frequencies or time intervals, or produce frequency or pulse outputs.

- rapid counting up to 166kHz
- 2 preset values, 6 digits each
- 3 programmable status messages or displays
- 2 counter inputs
- programmable digital input filters
- selectable input voltages
- up and/or down counting
- up/down counting to 0
- up/down counting to preset value
- up/down counting with direction control
- counting range -999'999 to +999'999
- overflow indicator
- time measurement resolution of 0.1 microseconds
- frequency measurement to 166kHz
- output of programmable frequencies to 166kHz
- 4 digit programmable frequency divider
- digital outputs, programmable for defined events
- outputs to PCA2.D14 display module
- input for entering preset values
- input for external reset

Each module contains two independent counting systems.

Typical areas of application:

- **Counting** pulses up to a frequency of 166kHz

  Example: Counting revolutions or movements (pulses) and reacting at one or two preset positions.

  Switching a rate of feed at the first position and stopping at the second.

- **Counting** with direction control connected to incremental shaft encoders

  Examples: Simpler (unregulated) positioning with motors having flanged shaft encoders for high-racking warehouses, simple lifts, general X-Y positioning with servo motors.

  For high-quality, regulated positioning the "PCD4/6.H3.. motion control module" should be used.

- **Output of pulse strings** with preselectable frequency, e.g. simple drive for stepper motors within the start-stop frequency, via suitable power electronics.

  Examples: Positioning X-Y tables, automatic palletizing machines, handling robots appropriate to motor velocity (start-stop), processing materials: cutting lengths of wire, sawing off wooden beams, processing paper, textiles, etc.

  More accurate drive for stepper motors with calculated acceleration and braking ramps is achieved by using the "PCD4/6.H2.. module for stepper motor drives" together with the appropriate drive electronics.

- **Measuring** pulse length, period duration or frequencies

  Examples: Quartz precision for the measurement of velocity, number of revolutions, flow rate, etc.

Programming

A library of standard routines is available for programming the many functions. The user program executes these routines, via the CPU and PCD bus, to set the module's operating modes and control actions. Status and counter values can also be read at any time.

Examples of common applications show how these commands are executed.

Summary of models

Model	Number of functional units
PCD4.H120	2 counting and measuring systems

2. Technical information

2.1 General

Number of systems	2 (all data listed below refers to a single system)
Counting range	-999'999 to +999'999
Counting frequency	up to 166kHz (up to 333kHz with restrictions)
Data security	All data stored in the H1.. module is volatile (non-volatile Registers are available in the PCD)

Digital inputs

IN-A and IN-B — Signal voltage, jumper selectable

Set voltage	Ranges Low	High
• 24V , 10mA selectable as source or sink operation	-3...+4V	15... 32V
• 12V , 5mA source operation only	-3...+3V	8,8... 32V
• 5V , 2mA sink operation, TTL compatible	0... +1V	2,4... 5V
• 5V , 2mA source operation	-3...+1V	3,9... 10V

RESET and PRESET

Set voltage	Range Low	High
24V , 10mA	-3... +4V	15... 32V

Jumper selectable source or sink operation

Input filter for IN-A and IN-B — Programmable between 16Hz and 166kHz (see section 2.2)

Digital outputs

Process outputs

CCO	Counter output relative to zero
R1CO	Register 1 output relative to counter
R2CO	Register 2 output relative to counter
Current range	5..500mA, maximum leakage 1mA (load resistance min. 48 Ohm for voltage range 16..24V DC)
Voltage range	16..32V DC smoothed *)
Type of circuit	Electrically connected, not shortcircuit protected, positive switching
Voltage drop	1V at 500mA
Output slippage	Typically 2 microseconds, longer with inductive load as consequence of recovery diode

Display outputs

For 2 PCA2.D14 display modules

Data	1 output	valid for the whole H1 module
Clock	1 output	
Enable	2 outputs	
Voltage range	19..32V DC smoothed *)	
Voltage drop	< 0.5V	
Current output	1... 100mA	
Type of circuit	Electrically connected, not shortcircuit protected, positive switching	

*) The supply for both process and display outputs is provided from the positive terminal on the bus module

Power supply

External (user)	+24V DC (19..32V) smoothed ripple max. 10%
Consumption from external 24V supply	50mA (for 2 x PCA2.D14 modules, excluding process outputs)
Internal from PCD4 bus	(+5V) max. 120mA (+15V) max. 10mA

Operating conditions

Ambient temperature	0°C..+50°C without forced ventilation
Interference resistance	1kV with capacitive coupling in accordance with IEC 801-4 for 24V inputs and outputs 0.5kV for 12V inputs 0.25kV for 5V inputs
Mechanical resistance	In accordance with IEC 65A
Storage conditions	Temperature: -20°C..+85°C Humidity: 0..95%

Programming	With function blocks, as PCD user program

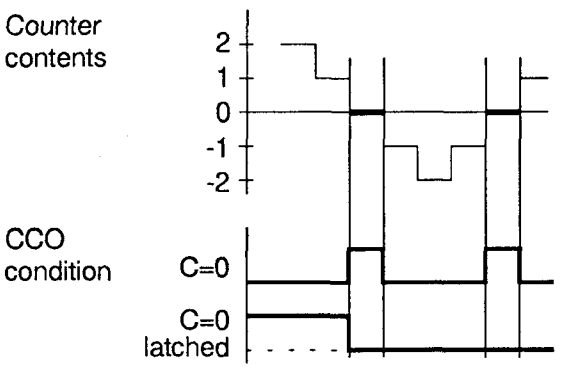
2.2 Counter

Number of counting systems	2 independent systems
Counting range	-999'999 to +999'999
Counting frequency	Standard: 166kHz Maximum: 333kHz (with restrictions)
Input filter	Programmable low-pass filter for frequencies of 333kHz (without filter), 166kHz, 16kHz, 1.6kHz, 166Hz and16Hz
Counting modes	7 modes - x4 mode - x2 mode - x1 mode } for incremental shaft encoder - dual count (+/-) - dual count (+) - single count - enabled count
Counting direction	- Primary counting direction "up" - Primary counting direction "down" - One direction only, "up" or "down" - Both "up" and "down" counting
Enabled counting (Counting activated by...)	- CCO = H - R1CO = H - R2CO = H - IN-B = L - IN-B = H - IN-B between positive edges - IN-B between negative edges
Use of digital counter outputs	Outputs CCO, R1CO and R2CO are set or reset by comparing values. They provide almost instantaneous (2µS) control of the process (e.g. closing a valve when a given count value has been reached). CCO, R1CO and R2CO can (like other PCD outputs) be polled by the user program to influence the course of the program.

- Output CCO	- C < 0	
	- C ≤ 0	
	- C = 0	
	- C ≥ 0	
	- C > 0	
	- C < 0	latched
	- C = 0	latched *
	- C > 0	latched
- Output R1CO	- C < R1	
	- C ≤ R1	
	- C = R1	
	- C ≥ R1	
	- C > R1	
	- C < R1	latched
	- C = R1	latched
	- C > R1	latched
- Output R2CO	- C < R2	
	- C ≤ R2	
	- C = R2	
	- C ≥ R2	
	- C > R2	
	- C < R2	latched
	- C = R2	latched
	- C > R2	latched

Digital outputs are switched almost instantaneously by counter module hardware when any of the abive conditions are reached.

*) Difference between "latched" and "not latched", example where C = 0:



The C = 0 condition sets output CCO for only as long as this condition is true. With high frequencies, this may last only a few micro-seconds.
The C = 0 **latched** condition requires the prior setting of the output by the user program. When the condition occurs, the output is reset and remains reset.
For further details see chapter 6.10.

2.3 Pulse generator

Frequency	10Hz to 166kHz *
Mark/space ratio	Standard: 50% Possible: 0.0001 to 99.9999%
Single pulse output	3 µs up to 3 years
Pulse resolution	3 µs up to 100s

*) For higher frequencies the output should have a load (100mA).

2.4 Measurements

Measuring modes	- Frequency measurement - Pulse length measurement - Period measurement - Multiple period measurement
Accuracy	Better than ±50ppm at 25°C
Temperature operation	150ppm between 0°C ... +70°C

2.4.1 Frequency measurement

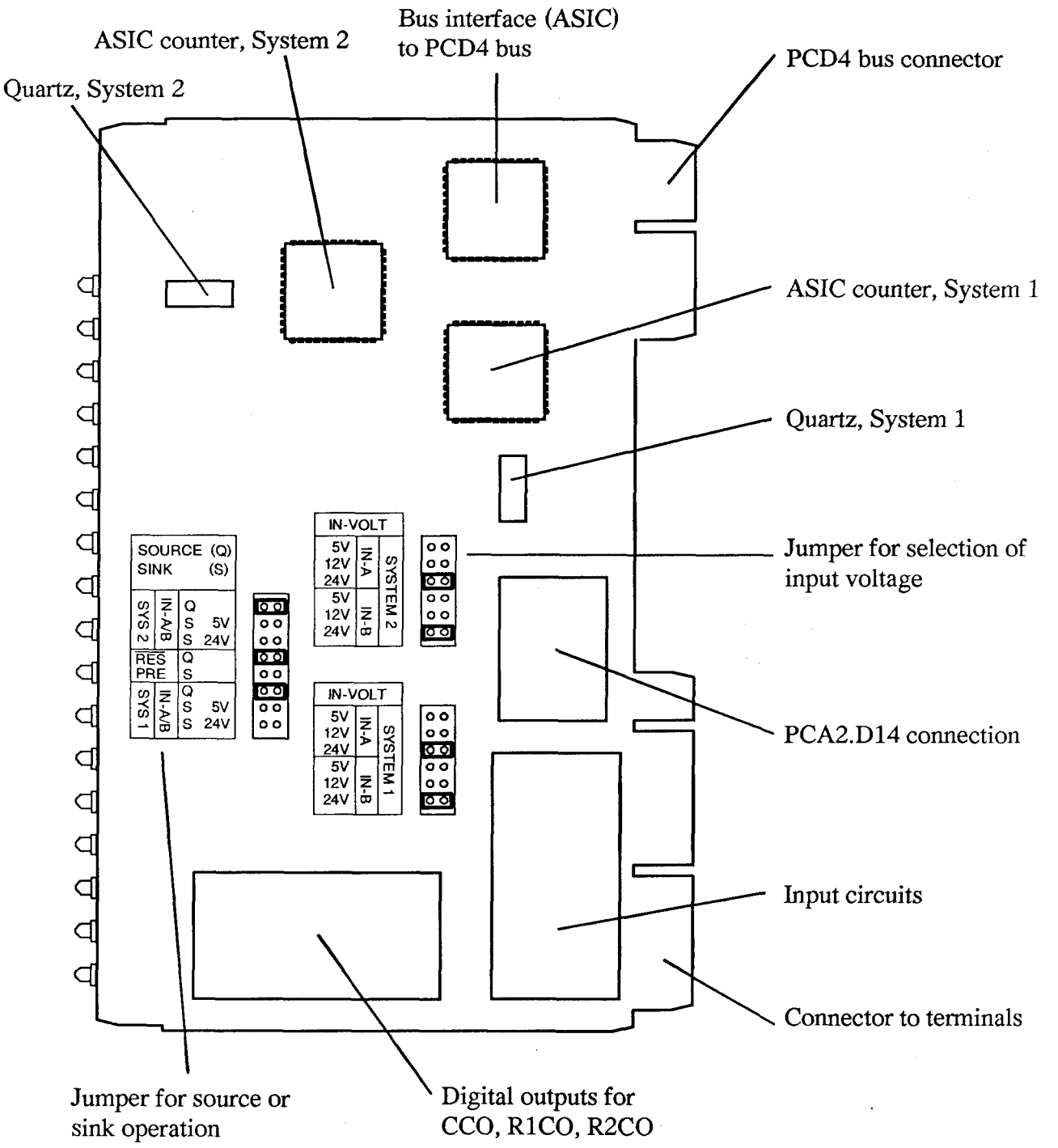
Time window	3 µs to 100s
Result	Total periods counted within window time window. If time widow is 1 sec, the result is in Hz.
Resolution	Dependent on time window size, 1 in 999'999
Max. frequency	166kHz

2.4.2 Period measurement

Result	Number of pulses produced by the internal quartz clock during the measuring time
Time resolution	0,1µs to 10ms
Absolute resolution	1 in 9'999'990'000
Max. period length	3 years

3. Presentation

3.1 Printed circuit, jumper positions



Jumper factory setting: all inputs in source operation, 24V

Jumper "IN-VOLT": input voltage

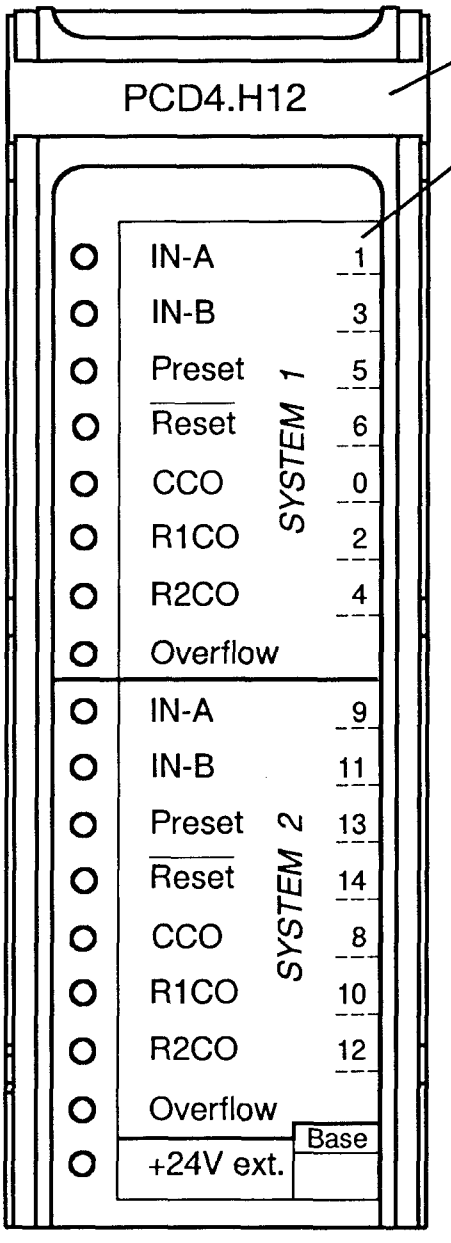
For each IN-A and IN-B input, a set voltage can be selected separately from 5, 12 or 24V.

The $\overline{\text{RESET}}$ and PRESET inputs are designed for 24V DC and are unchangeable.

Jumper "SOURCE (Q) - SINK (S)": source or sink operation

Jumper setting		Meaning	
IN-A/B	Q	Source operation 5 to 24V	selectable per system
	S 5V	Sink operation 5V (TTL)	
	S 24V	Sink operation 24V	
$\overline{\text{RES}}$ / PRE	Q	Source operation 24V	selectable per module
	S	Sink operation 24V	

3.2 Front panel



Model number

Terminal number

Input A
Input B
Preset input
Reset input

LED on if input is logic high

Counter output rel. to zero
Reg. 1 output rel. to counter
Reg. 2 output rel. to counter

LED on if output is set

Overflow display

on after an overflow

Input A
Input B
Preset input
Reset input

LED on if input is logic high

Counter output rel. to zero
Reg. 1 output rel. to counter
Reg. 2 output rel. to counter

LED on if output is set

Overflow display

on after an overflow

External +24V supply

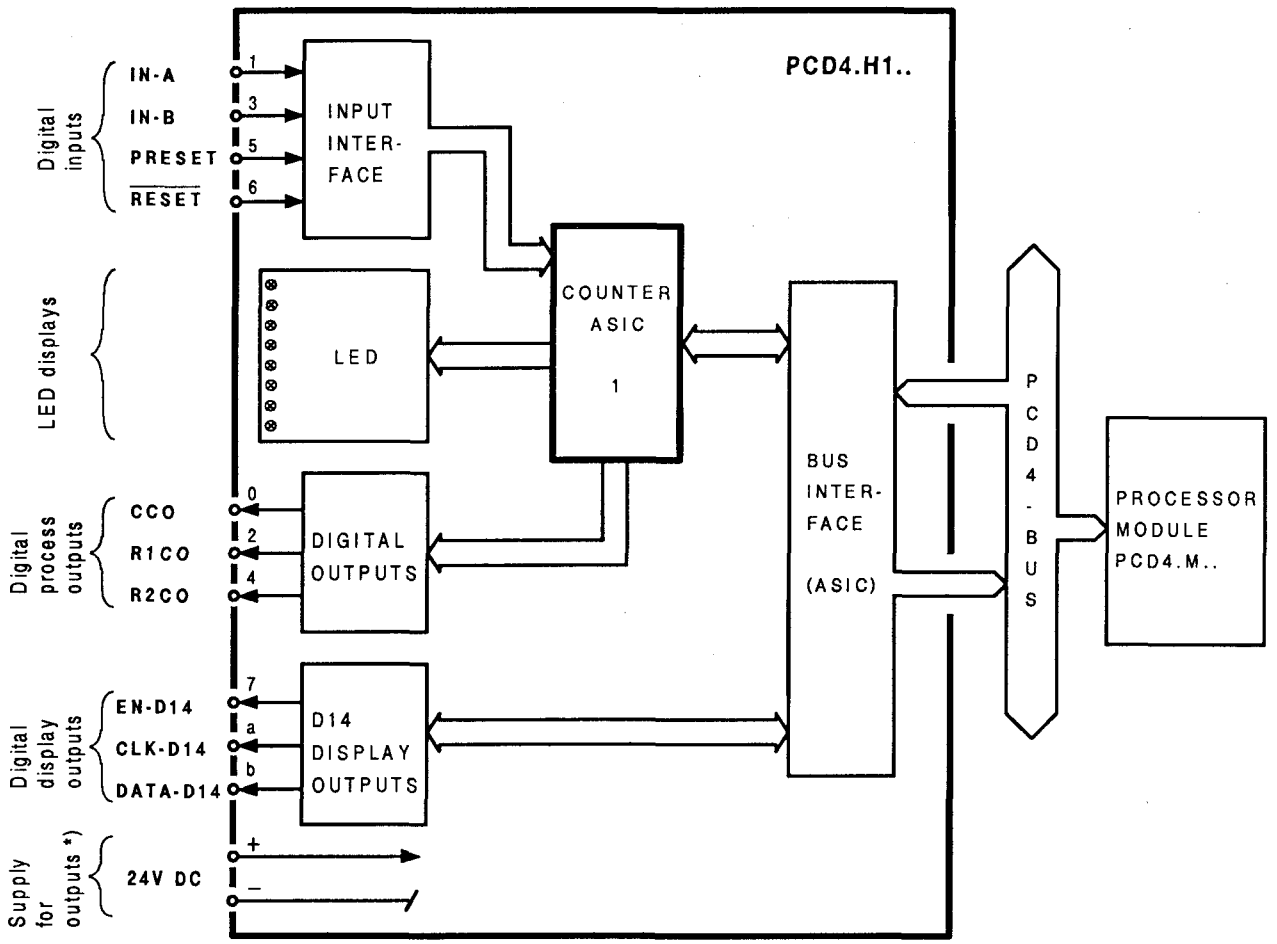
on if 24V DC present

CCO: Counter Controlled Output
R1CO: Register 1 Controlled Output
R2CO: Register 2 Controlled Output

4. Logic diagrams

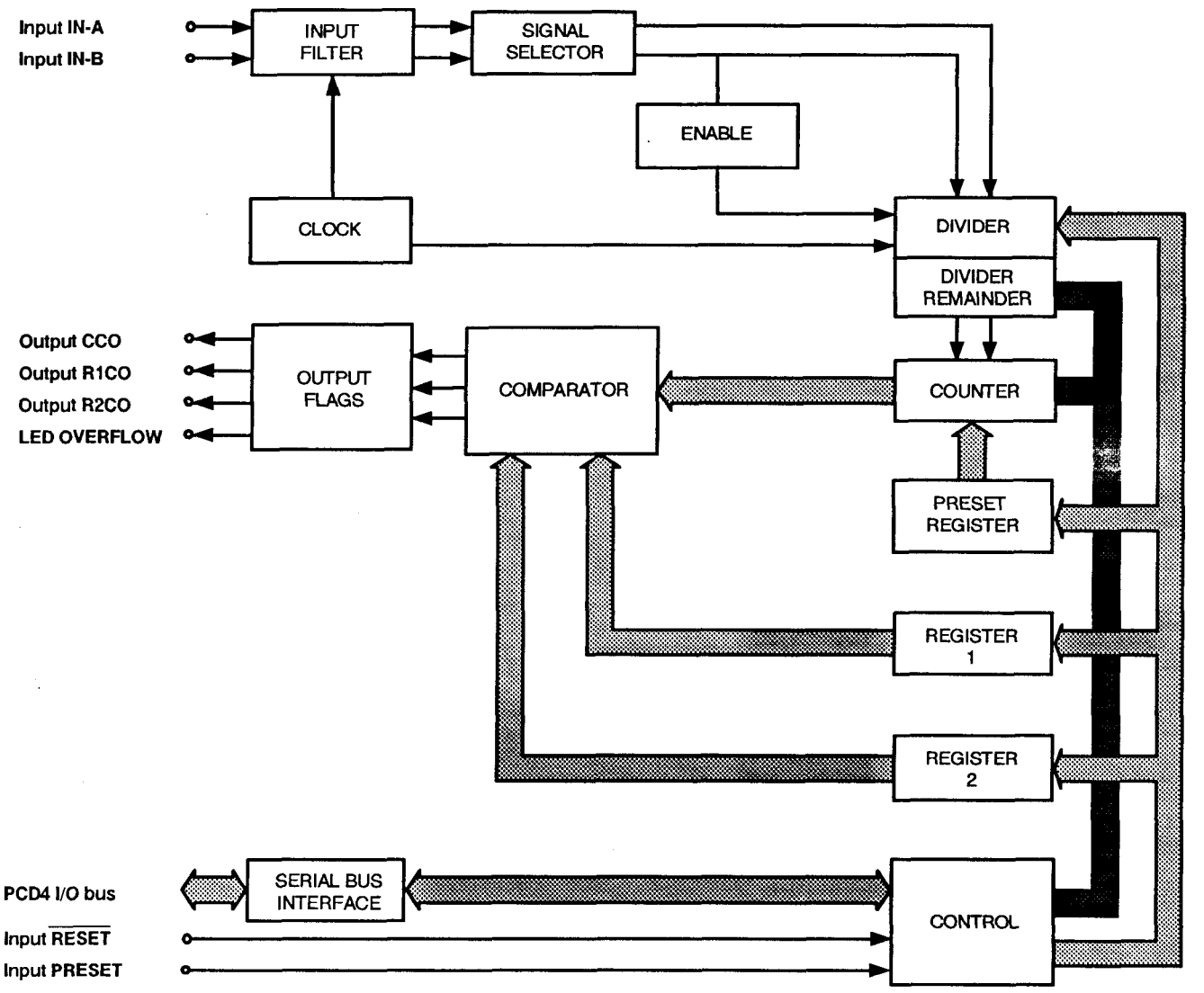
4.1 Module

Shows only System 1



*) The positive supply is shared for all outputs, i.e. CCO, R1CO, R2CO, EN, CLK and DATA.

4.2 ASIC counter



4.3 Functional overview

Inputs IN-A and IN-B are connected to an input circuit which has a jumper selectable input voltage of 5V, 12V or 24V. The signals are then passed to the programmable counter, a complex custom designed ASIC, which forms the heart of the counting system (ASIC = Application Specific Integrated Circuit). Inside the ASIC, both signals first pass through a programmable input filter. This lowpass filter circuit can be programmed for cut-off frequencies of 166kHz, 16kHz, 1.6kHz, 166Hz and 16Hz.

The two signals are then fed to the "Signal Selector". This is also programmable and determines the counting mode, e.g. the mode for an incremental shaft encoder "x 4", or a simple "Single Count" mode.

Next, the signals are routed to the programmable "Divider", with a divisor range of 0 to 9999. Unless specified otherwise, the divider is 0 and has no effect, i.e. incoming signals go directly to the counter. If the divider is loaded with, for example, the value 9, only the tenth signal will be passed to the counter. Values up to 9 (for this example) are sent to the "Divider Remainder" register.

The divider can be used to increase the counting range by a factor of up to 10,000. Counts can also be scaled by dividing incoming signals by any value between 1 and 9999.

The programmable "Clock" circuit produces a quartz-driven frequency which is used by the input filters, and for applications such as pulse generators. When connected to the "Enable" circuit it can be used for measurements of period, pulse length and frequency.

Frequencies of 10MHz, 1MHz, 100kHz, 10kHz, 1kHz and 100Hz are selectable by the user program.

The output of the divider is then routed to the actual counter. This has a range of -999'999 to +999'999 (6 digits with sign bit), and its contents can be read or reloaded at any time. A similar value can also be loaded into the "Preset Register". This preset value can be loaded directly into the counter at a given time, when the "Preset" input condition is activated.

Both the 6-digit registers, R1 and R2, can be loaded and compared with the counter value. The results of the comparison defined in the user program appear almost instantaneously at digital outputs R1CO and R2CO, which are controlled directly by the counting module.

Digital output CCO can also be defined and refers to the counter itself, i.e. counter value 0.

Module outputs CCO, R1CO und R2CO can be polled by the user program for the purposes of program control.

The fourth ASIC output is called "Overflow". If the counter overruns, i.e. at values greater than +999'999 or lower than -999'999, this flag is set. It can only be reset by the user program.

The logical states of the 4 flags or outputs indicated can be followed on the LEDs on the front panel of the module.

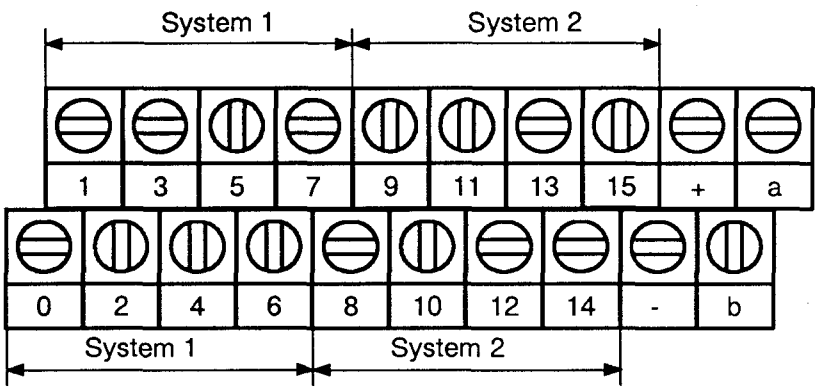
The "/RESET" input is used to reset and enable the counting module. The edge produced when a +24V signal is applied results in a reset. +24V must be present at the "/RESET" input to enable the module. The "/RESET" lamp must be on whenever the module is in use. The module can be reset by removing and restoring the +24V at the "/RESET" input. This interruption must last at least 25us.

Note: This reset is NOT automatically executed by a "Restart Cold". (see chapter 6.13)

All data transfer between the CPU and the H120 module takes place via the PCD bus. This means that all commands and definitions, including read/write values, are passed across the PCD bus. In contrast, counting and comparing values, and the appropriate responses to the process outputs, all take place purely inside the H120 module. In other words: the user program tells the module what to do, after which the module functions independently.

5. Connections and addressing

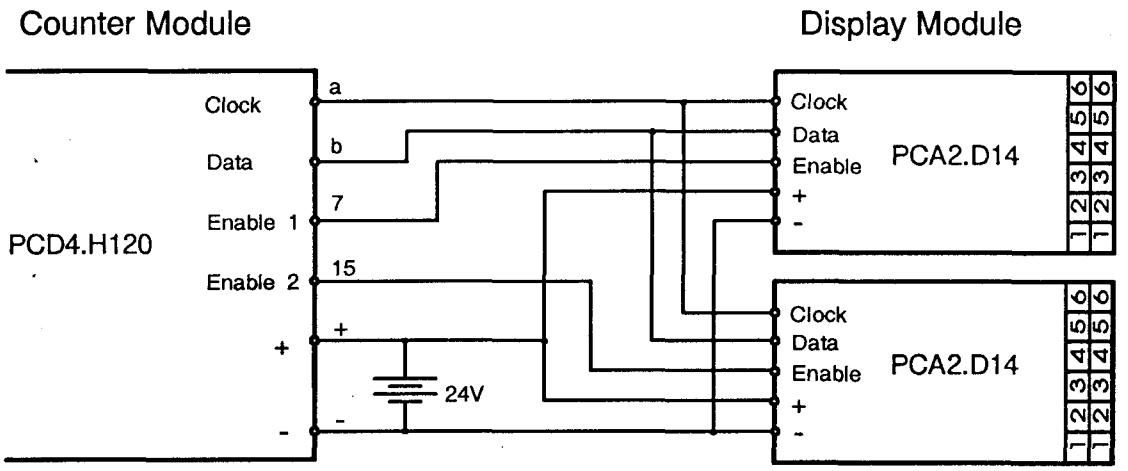
5.1 Terminal connectors (on bus module)



Terminal connectors for System 1 / System 2

Function		System 1	System 2
Input	IN-A	1	9
Input	IN-B	3	11
Input	Preset	5	13
Input	Reset	6	14
Output	CCO	0	8
Output	R1CO	2	10
Output	R2CO	4	12
Output Enable, Display Module D14		7	15
Shared connections			
Supply for all outputs		+	
0V connection, frame, ground		-	
Clock output, Display Module D14		a	
Data output, Display Module D14		b	

Outputs for Display Module PCA2.D14



5.2 Addressing

The module occupies 16 addresses on the PCD4 bus.

Unlike the usual input or output modules, the H1 module's input (read) addresses have different meanings according to the selected operating mode. The same addresses also have different meanings if used as output (write) addresses.

Read elements				Write elements	
0	--			0	D14-Clock
1	--			1	D14-Data
2	--			2	D14-Enable1
3	--			3	D14-Enable 2
4	--			4	--
5	--			5	--
6	Busy 2			6	Enter 2
7	Busy 1			7	Enter 1
	Outputs and Flags	Status Flags	Data		Data
8	R1CO	IN-B	DATA 0	8	DATA 0 (LSB)
9	R1<C	IN-A	DATA 1	9	DATA 1
10	R2CO	R2CO	DATA 2	10	DATA 2
11	R2<C	R2<C	DATA 3	11	DATA 3
12	CCO	CCO	DATA 4	12	DATA 4
13	C<O	C<O	DATA 5	13	DATA 5
14	OVERFLOW	OVERFLOW	DATA 6	14	DATA 6
15	ENABLE	PRESET	DATA 7	15	DATA 7 (MSB)

Addresses 0 to 15 are relative

Absolute address = Module base address + Relative address

Note: Watchdog addresses 255 and 511 should NOT be used as element addresses for digital inputs and outputs. In general, special modules such as analogue, motion control or fast counting modules, must not be used at addresses 240..255 and 496..511.

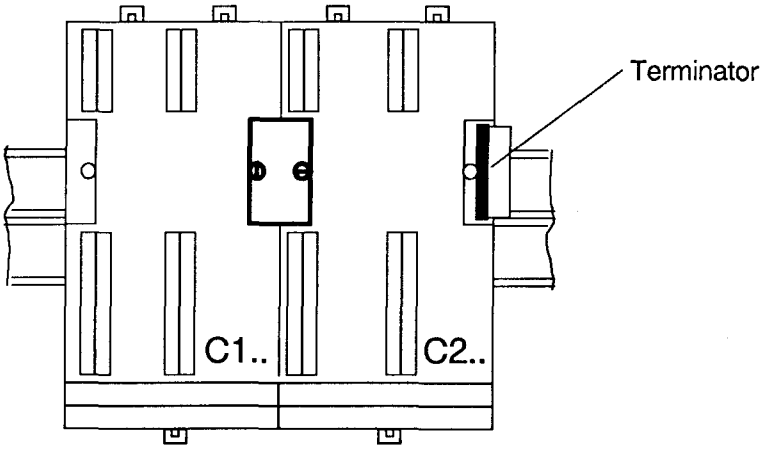
5.3 Bus terminator

Practical experience has shown that the presence of a long PCD bus and a high level of noise can have an unfavourable effect on the PCD4.H120 counting and measuring module, as it is more sensitive to noisy bus signals than other PCD modules.

Extensive investigations have demonstrated that the best results are achieved with a terminator network at the end of the PCD bus. This terminator is enclosed in every package. It should be plugged into the empty socket (connection to next bus module) on the last bus module.

It is recommended that, in principle, a terminator should be inserted in every system.

The terminator can be ordered separately under number 4 717 4830 0.



6. A brief introduction to programming the PCD4.H120

In practice, over 90% of all applications for fast counter modules are simple counting tasks, particularly for motion control. These only make use of a fraction of the PCD4.H120 module's powerful features.

Using a small example, the following sections therefore outline a procedure for creating a user program WITHOUT reference to complex details.

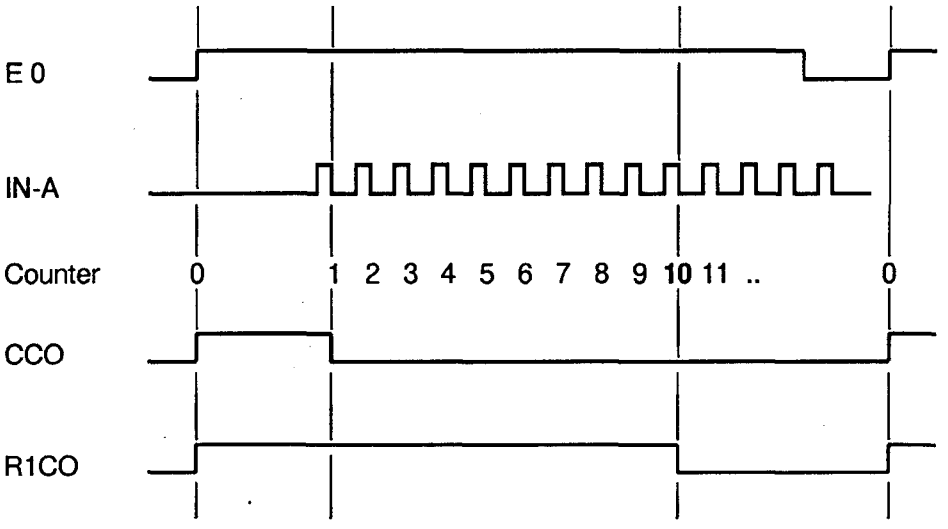
6.1 Example QUICK1

When digital PCD input 0 goes high, start counting the pulses received at input IN-A of the H120 module and set output R1CO high. Once a preselected number of pulses has been received, set output R1CO low (reset it).

The number of pulses to be counted is set by the 2-digit BCD switches on inputs 16 - 23. (The diagram below shows a setting of 10).

For control purposes, the CCO (counter output) should be defined so that it is high when the counter ≤ 0 (is less than or equal to zero). The maximum counting frequency is 100 Hz.

Timing Diagram:

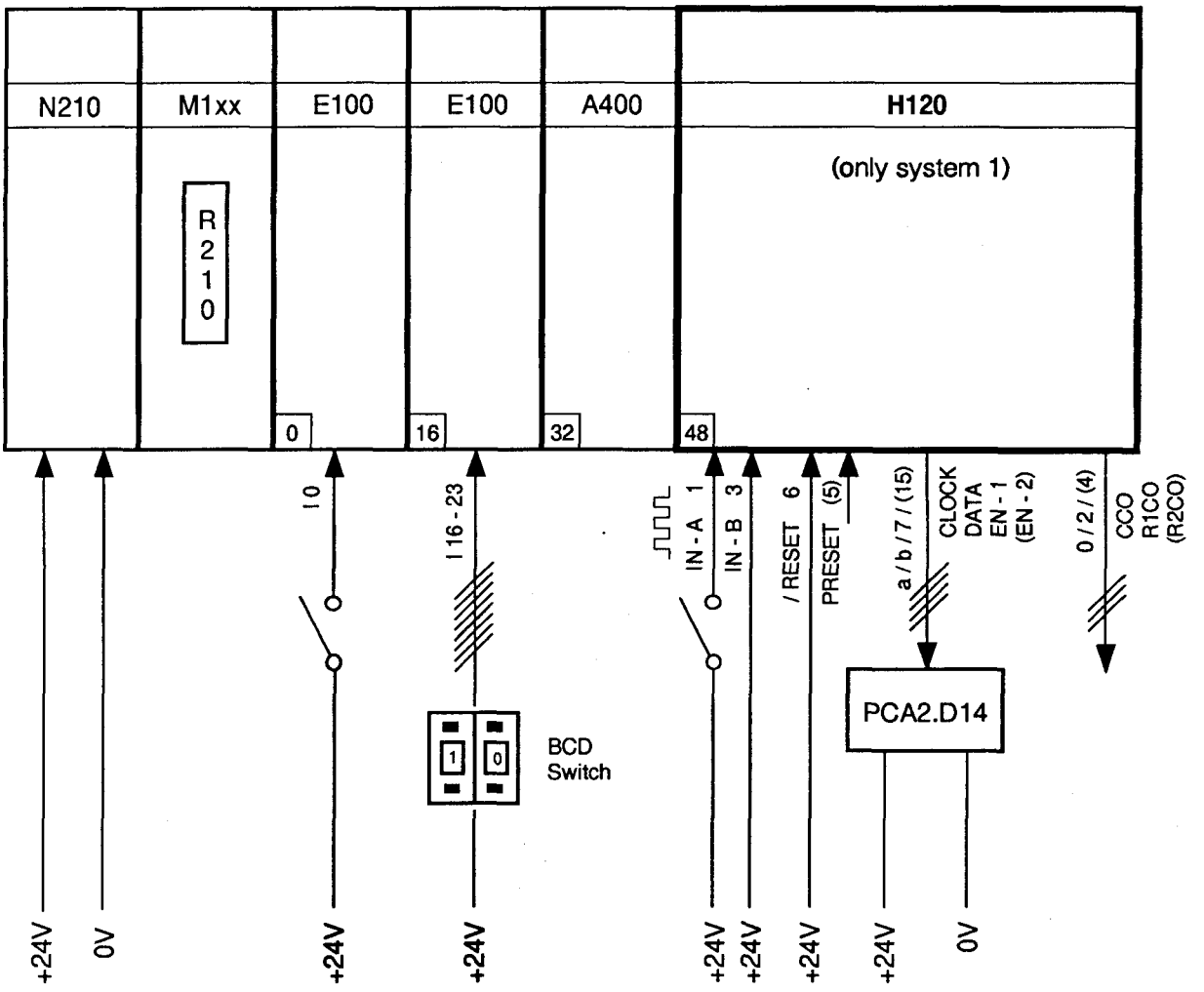


The PCD4.H120 has a base address of 48 (48 - 63).

Counting system 1 is used here. The first example does not display the counter status on the "D14" display module.

Arrangement of hardware:

The example is built around the PCD4 workshop model V-PCX 20.



Programming:

A user program for a counting task can be divided into three fundamental operations:

- Initialization of the H120 module
- Definition of the counting task
- Evaluation of the count

The following is true for the given example:

Intialization:	- Input Filter: 166 Hz
	- Output CCO: C $\leq$ 0 (not absolutely necessary)
	- Count Mode: Single Count
Definition of task:	- Reset Counter: Write to Counter, Value 0
	- Register 1 at preset value: Write to Register 1
	- Define output R1CO : C = R1, latched
	- Activate output R1CO : Enable Outputs
Evaluation of count:	- Counter = Register 1 —> R1CO = L ?

This description of the task also outlines the structure of the program. It is now only a matter of writing the source code.

Before writing the program, we'll summarize the main commands for programming the PCD4.H120.

Sub-set of commands, for brief introduction

Code	Command	FB name
0	Read Counter and Flags	RCF_x
32	Write To Counter, Positive	WCR_x
40	Write To Register 1, Positive	WCR_x
44	Write To Register 2, Positive	WCR_x
97	Disable Outputs; Reset Overflow Flag	WF_x
99	Enable Outputs; Reset Overflow Flag	WF_x
(129	Output CCO: $C \leq 0$	WF_x)
134	Output CCO: C = 0 Latched	WF_x
142	Output R1CO: C = R1 Latched	WF_x
150	Output R2CO: C = R2 Latched	WF_x
160	Input Filters: 16 Hz - 30 ms	WF_x
161	Input Filters: 166 Hz - 3 ms	WF_x
162	Input Filters: 1.6 kHz - 300 µs	WF_x
163	Input Filters: 16 kHz - 30 µs	WF_x
164	Input Filters: 166 kHz - 3 µs	WF_x
165	Input Filters: 333 kHz - 0.1 µs	WF_x
224	Count Mode: x4	WF_x
226	Count Mode: x2	WF_x
228	Count Mode: x1	WF_x
234	Count Mode: Single Count	WF_x

_x signifies: _1 for system 1, _2 for system 2

The comlete list of commands can be found in Section 7.1

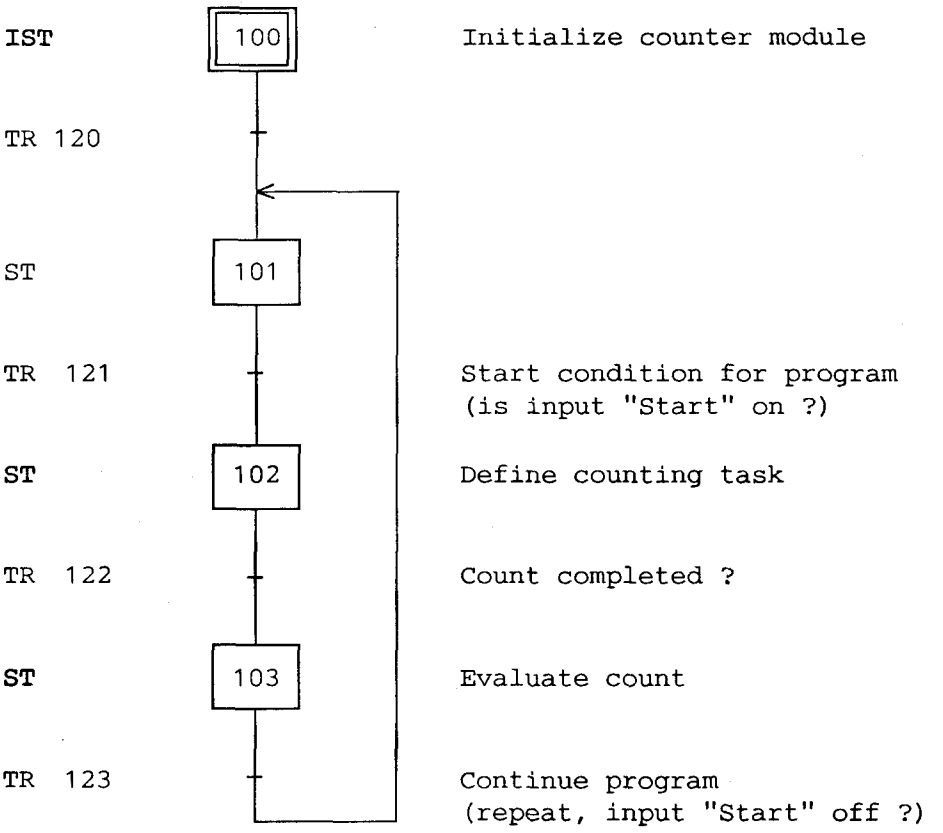
A function is activated by calling the appropriate function block in accordance with the following structure:

Call FB

```
CFB   WF_1
      161
```

name WF, system 1
constant 161 as parameter

Program flow is always sequential for counting tasks. When using the PCD, sequential programming should preferably be structured in GRAFTEC, assigning a GRAFTEC element to each routine.

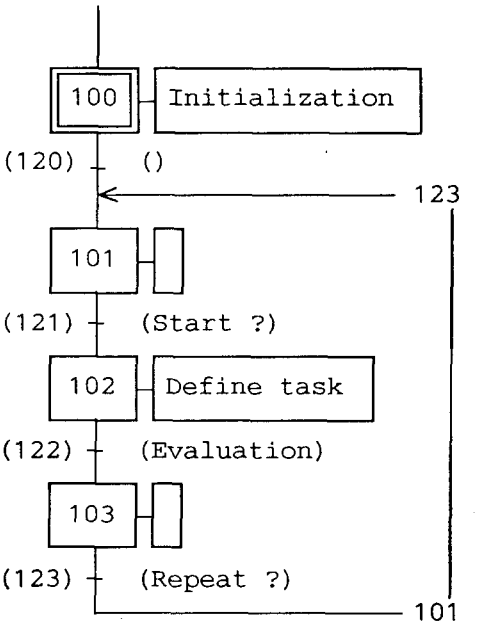


A detailed description of SAIA-GRAFTEC can be found in Section 3.3 of the user manual. Use of the GRAFTEC editor is explained in Section 6 of the user manual.

Source program: "QUICK1.SRC"

A sequential block (SB) is opened using the GRAFTEC editor.
The structure of the program is then drawn.

"QUICK1.GLS"
(extract)



Program code is entered into individual steps and transitions in the GRAFTEC editor, or using any pure ASCII editor (not SEDIT).

The SB must be called from a COB.

```
COB             0                       ;Cyclic program
                0

CSB             5                       ;Call of SB 5 (GRAFTEC)

ECOB

;---------------------------------------------------------------
```

```
SB      5               ; Sequential block 5 (GRAFTEC)

IST     100                 ;Initialization
        O    120

CFB         WF_1        ; FB Write Function, Syst. 1
            161         ;  Input Filters: 166 Hz

CFB         WF_1        ; FB Write Function, Syst. 1
            129         ;  Output CCO: C ≤ 0

CFB         WF_1        ; FB Write Function, Syst. 1
            234         ;  Count Mode: Single Count

EST

;--------------------------------

ST      101
        I    120
        I    123            ;Repeat ?
        O    121            ;Start ?
EST

;--------------------------------

ST      102                 ;Def. task
        I    121            ;Start ?
        O    122            ;Evaluation

LDL         LD_VAL_1    ;Scratch register
            0           ; value to load
CFB         WCR_1       ;FB Write Counter/Register, Syst.1
            32          ; Write to Counter, Positive
            LD_VAL_1    ; with value in scratch register

DIGI        2           ;Read BCD information
        I   16          ; at inputs 16 (- 23) and put
            LD_VAL_1    ; in scratch register
CFB         WCR_1       ;FB Write Counter/Register, Syst.1
            40          ; Write to Register 1. Positive
            LD_VAL_1    ; with value in scratch register

CFB         WF_1        ;FB Write Function, Syst. 1
            142         ; Output R1CO: C = R1, Latched

CFB         WF_1        ;FB Write Function, Syst. 1
            99          ; Enable Outputs

EST

;--------------------------------

ST      103
        I    122            ;Evaluation ?
        O    123            ;Repeat ?
EST

;--------------------------------
```

```
TR      120
        I    100          ;Initialization
        O    101
ETR

;---------------------------------

TR      121               ;Start ?
        I    101
        O    102          ;Def. task

STH     I    0       ; Check input "Start"

ETR

;---------------------------------

TR      122               ;Evaluation
        I    102          ;Def. task
        O    103

CFB          RCF_1    ; FB Read Counter and Flags, Syst.1
             0        ;  Read Counter and Flags
STL          R1CO_1   ; Check output R1CO, Syst. 1

ETR

;---------------------------------

TR      123               ;Repeat ?
        I    103
        O    101

STL     I    0       ; Check input "Start"

ETR

;---------------------------------

ESB
```

Comments on the GRAFTEC program "QUICK1"

COB 0 : Every PCD user program must have at least one COB. In the above example, SB 5 is called unconditionally from COB 0.

SB 5 : Sequential Block

The SB is the outer skin of the GRAFTEC structure. Every selfcontained GRAFTEC sequence must be enclosed in an SB.

IST 100 : This initializes the H120 module. If incorporated into the structure as shown, the IST is processed only when the SB is called for the first time.

TR 120 and ST 101 are empty and in this example are present only to satisfy the requirements of GRAFTEC structure (an empty transition is always assumed true).

TR 121 : This transition is left for the next step only if the ACCU is high at the end of the transition, i.e. if PCD input I 0 is switched on (high).

ST 102 : This step loads the counter (with zero) and loads register 1 with the preset value. It also defines and enables R1CO.

TR 122 : This transition contains the heart of the program and needs some explanation.

A novice GRAFTEC programmer might expect a GRAFTEC loop for the continuous processing of the "Read Counter and Flags" routine, which would loop until the preset value had been reached. However, a transition is left only when the ACCU is high at the end, and if not, the ENTIRE transition is executed again on the next pass of the program. This fact is exploited here. The "Read Counter and Flags" routine is processed on each pass, then R1CO is polled to determine the transition's state.

ST 103 : This step is empty in the present example and is for the purposes of GRAFTEC structure only. In practice, evaluation of the count would take place here.

TR 123 : Here input I 0 is checked to see if it's low. Step 103 and transition 123 could also be omitted by using AND to link the "I 0 = low" condition in TR 122 with R1CO. However, for the sake of clarity, some structure has been sacrificed.

The source program has now been written, but can't be assembled yet because it lacks (1) the definitions of symbols used, and (2) the FBs called. There are also no definitions of the base addresses of the counter module itself, nor of the other registers, flags and FBs which are not visible from the outside world.

Software package PCD9.H1E1 contains work files and examples for use with the PCD4.H120 counter module. (The above examples are also included in QUICKx.SRC). The following files are required by our example:

- EXTN.TXT
- H1DEF_12.SRC
- H1FB_12.SRC

"EXTN.TXT" contains the list of externally defined symbols. This list should be copied or $included at the start of each user program for the counter module.

```
extn    wf_1,wd_1,wcr_1,rdr_1,rcf_1
extn    ld_val_1,ct_val_1,div_val_1,f_reg_1,s_reg_1
extn    r1co_1,r2co_1,cco_1,ovflf_1,enab_1,inb_1,ina_1,pres_1

extn    wf_2,wd_2,wcr_2,rdr_2,rcf_2
extn    ld_val_2,ct_val_2,div_val_2,f_reg_2,s_reg_2
extn    r1co_2,r2co_2,cco_2,ovflf_2,enab_2,inb_2,ina_2,pres_2

extn    d14_12,enable_1,enable_2
;-----------------------------------------------------------------
```

Although it is rare for all the symbols listed to be used in the same module, the whole list is included for the sake of simplicity. It does NOT take up any space in the PCD's memory.

"H1DEF_12.SRC" contains the definitions of all the symbols used, the module base addresses, and some registers, flags and FBs. These 4 items MUST be re-defined, unless the default definitions are to be used. If using a "PCA2.D14" display module, this should also be defined.

```
************************************************************************

ba_12      equ   o   48   ;; Base address of the H120 module

bflag_12   equ   f   500  ;; Base address of 100 flags (F500-599)

br_12      equ   r   500  ;; Base register of 20 regist.(R500-519)

bfb_12     equ       800  ;; Base function block of 20 FBs (FB800-819)

d14_12     equ       999  ;; FB for display module PCA2.D14 ¦
f_d14_12   equ   f   400  ;; Base address of 48 flags       ¦ optional
c_d14_12   equ   c   999  ;; 1 scratch counter              ¦ for D14
r_d14_12   equ   r   999  ;; 1 scratch register             ¦

************************************************************************
```

"H1FB_12.SRC" contains all the FBs for controlling the counter module.

This file must NOT be altered !

The 3 files:

- user source program with "EXTN" list
- H1DEF_12.SRC with base addresses
- unaltered H1FB_12.SRC

should be assembled separately, then linked together. It is recommended that the user file is linked first, so that its ".LST" file contains the absolute addresses (see Section 8.1).

A few warnings are displayed after linking, because not all the FBs available have been called. These warnings are not significant in our program.

The program can now be loaded into the PCD. For initial tests, only this one program should be loaded.

The program can be viewed on-line from the "Graftec" menu.

Commissioning:

If installation matches the hardware arrangement shown above, and all the jumpers on the module are set to +24V source (Q) (standard setting), the program should run correctly after a "Run" command.

The debugger can be used to display the counter value. This is located in the defined base register + 1. If R 500 has been defined as the base, the counter value can be seen by displaying the contents of R 501 (refreshed).

After "Run", but before starting the program via input 0, the three front panel LEDs "IN-B", "/RESET" and "+24V ext" should be on.

After switching on input 0 (Start) and before pulses reach "IN-A", the "R1CO" and "CCO" lamps should also be on. After receipt of the first pulse at "IN-A" (falling edge), "CCO" should go off (this output is only high when $C \leq 0$).

After receipt of the number of pulses preset by the BCD switches, the "R1CO" lamp and the output are turned off.

When the start input "I 0" has been switched off and on again, the sequence is repeated from the beginning.

If more than the preset number of pulses is received, they are not shown by the debugger. This is because the "Read" sequence is exited once the preset number of pulses has been reached, and the register containing the counter value ceases to be updated. However, the counter does continue to count these pulses.

A continuous display of the counter value can be obtained by placing the "Read" sequence into each transition (see example QUICK2).

Possible problems:

- It does not work at all

  - Check wiring
  - Is the "Reset" lamp on? (+24V at Reset input)
  - Is the lamp "+24V ext" on? (ext. +24V supply)
    Is voltage sufficiently smoothed ?
  - Check H120 module base address in file H1DEF_12.SRC and in the PCD
  - Do all commands access the correct system(s) ?
  - Switch the PCD off and on again for hardware reset, or briefly remove the reset signal (+24V).
    (A "Restart Cold" does NOT produce a hardware reset)

- The "CCO" lamp does not go off after the first pulse, although it was correctly received (lamp IN-A).

  - Down counting: the +24V signal is missing from IN-B (common error)

- The R1CO lamp does not go off after the preset number of pulses have been received.

  - Check pulse voltage (+24V), (lamp IN-A) or adjust with jumper (+5V, +12V).
  - Has the preset value been read correctly ? (Debugger or D14. Format PCD - PCA, "DIGI" - "DIGIR")
  - Is the CPU in RUN ? - Is it the right CPU ?
  - Count direction "up" ? (+24V at IN-B); "CCO" off ?

- For more complex programs

  - Has the correct command sequence been followed ? (section 7.6)

6.2 Example QUICK2 in GRAFTEC

The preceding example QUICK1 is expanded so that current counter status and the preset value can be displayed on a PCA2.D14 display module at every stage in the user program.

Software package PCD9.H1E1 contains function block "D14_12", which can be called in the user program and then take over independent control of the display module. When invoking FB "D14-12" the following 3 parameters must be defined:

- Parameter 1: PCD register to be shown in upper display
- Parameter 2: PCD register to be shown in lower display
- Parameter 3: Enable System 1 or System 2

The connection diagram is given in section 5.1.

The easiest place from which to call FB "D14_12" is the COB, directly after the call to the counting task SB. The FB is constantly called by exploiting the fact that, in GRAFTEC, the program returns to the calling block (COB) after every step and every unfulfilled transition. For example, if the FB should only be called every 0.5 sec, to minimize its effect on processing time of the user program, the FB call is made timer dependent (see example QUICK2).

The aim of this display is to show current counter status at every stage of the program. It is for the user to ensure that, at all stages in the program, current counter status is available for display in a PCD register. The FB "RCF_12" (Read Counter and Flags) should therefore be called from all transitions in which the program might wait. In our example these would be transitions TR 121, 122 and 123.

For a continuous display, the timer should be omitted and FB "D14_12" invoked unconditionally.

Source program "QUICK2.SRC"

```
extn    wf_1,wd_1,wcr_1,rdr_1,rcf_1
extn    ld_val_1,ct_val_1,div_val_1,f_reg_1,s_reg_1
extn    r1co_1,r2co_1,cco_1,ovflf_1,enab_1,inb_1,ina_1,pres_1

extn    wf_2,wd_2,wcr_2,rdr_2,rcf_2
extn    ld_val_2,ct_val_2,div_val_2,f_reg_2,s_reg_2
extn    r1co_2,r2co_2,cco_2,ovflf_2,enab_2,inb_2,ina_2,pres_2

extn    d14_12,enable_1,enable_2
;-------------------------------------------------------------------

COB         0
            0

CSB         5

STL     T   0
LDL     T   0          ; Load timer with
            5          ;  0,5 sec

CFB     H   D14_12     ; FB D14 for System 1 and 2
            CT_VAL_1   ;  Upper Display (counter status Syst.1)
            LD_VAL_1   ;  Lower Display (last value loaded)
            ENABLE_1   ;  Output Enable 1

ECOB
;-------------------------------------------------------------------

COB         1          ; COB for other tasks
            0
NOP

ECOB
;-------------------------------------------------------------------
```

```
SB      5

; The steps are exactly the same as in example "QUICK1.SRC"
;        and are not reproduced here.

TR      120
        I    100            ;Initialization
        O    101
ETR

;---------------------------------

TR      121                 ;Start ?
        I    101
        O    102            ;Def. task

CFB          RCF_1      ; FB Read Counter and Flags, Syst.1
             0          ;  Read Counter and Flags

STH     I    0          ; Check input "Start"

ETR

;---------------------------------

TR      122                 ;Evaluation
        I    102            ;Def. task
        O    103

CFB          RCF_1      ; FB Read Counter and Flags, Syst.1
             0          ;  Read Counter and Flags

STL          R1CO_1     ; Check output R1CO, Syst. 1

ETR

;---------------------------------

TR      123                 ;Repeat ?
        I    103
        O    101

CFB          RCF_1      ; FB Read Counter and Flags, Syst.1
             0          ;  Read Counter and Flags

STL     I    0          ; Check input "Start"

ETR

;---------------------------------

ESB
```

6.3 Example QUICK3 in BLOCTEC

If a BLOCTEC example is required with the same functions as the preceding QUICK2, the following can be written:

```
extn      wf_1,wd_1,wcr_1,rdr_1,rcf_1
extn      ld_val_1,ct_val_1,div_val_1,f_reg_1,s_reg_1
extn      r1co_1,r2co_1,cco_1,ovflf_1,enab_1,inb_1,ina_1,pres_1

extn      wf_2,wd_2,wcr_2,rdr_2,rcf_2
extn      ld_val_2,ct_val_2,div_val_2,f_reg_2,s_reg_2
extn      r1co_2,r2co_2,cco_2,ovflf_2,enab_2,inb_2,ina_2,pres_2

extn      d14_12,enable_1,enable_2
;-------------------------------------------------------------------

Definit   EQU      PB    0
Start     EQU            0
End_Def   EQU      F     1
;-------------------------------------------------------------------

XOB          16            ; INITIALIZATION of H120 module

CFB          WF_1          ; FB Write Function, Syst. 1
             161           ;  Input Filters: 166 Hz

CFB          WF_1          ; FB Write Function, Syst. 1
             129           ;  Output CCO: C ≤ 0

CFB          WF_1          ; FB Write Function, Syst. 1
             234           ;  Count Mode: Single Count

EXOB
;-------------------------------------------------------------------
```

```
COB             0
                0

CFB             RCF_1       ; FB Read Counter and Flags, Syst.1
                0           ;    Read Counter and Flags

STH       I     Start
DYN       F     Start
CPB       H     Definit

STH       F     End_Def     ; Count ended ?
ANL             R1CO_1
RES       F     End_Def

STL       T     0
LDL       T     0           ; Load timer with
                5           ;    0,5 sec

CFB       H     D14_12      ; FB D14 for System 1 and 2
                CT_VAL_1    ;    Upper Display (counter status Syst.1)
                LD_VAL_1    ;    Lower Display (last value loaded)
                ENABLE_1    ;    Output Enable 1

ECOB
;-------------------------------------------------------------------

COB             1           ; COB for other tasks
                0
NOP

ECOB
;-------------------------------------------------------------------

PB              Definit     ; Definition of counting task

LD              LD_VAL_1    ; Scratch register
                0           ;    value to be loaded
CFB             WCR_1       ; FB Write Counter/Register, Syst.1
                32          ;    Write to Counter, Positive
                LD_VAL_1    ;    with value in scratch register

DIGI            2           ; Read BCD information
          I     16          ;    at inputs 16 (-23) and
                LD_VAL_1    ;    put into scratch register
CFB             WCR_1       ; FB Write Counter/Register, Syst.1
                40          ;    Write to Register 1. Positive
                LD_VAL_1    ;    with value in scratch register

CFB             WF_1        ; FB Write Function, Syst. 1
                142         ;    Output R1CO: C = R1, Latched

CFB             WF_1        ; FB Write Function, Syst. 1
                99          ;    Enable Outputs

SET       F     End_Def

EPB
```

The 3 fundamental operations

- Initialization of the H120 module
- Definition of the counting task
- Evaluation of count, or recognition of the end of counting

are exactly the same as with the GRAFTEC routines. Only the processing philosophy of the individual operations is different.

Whereas in GRAFTEC all routines are processed one after the other in the defined order (i.e. sequentially), a BLOCTEC structure must ensure that the appropriate PB or FB is called at the right moment for the task from the continuously running COB. Certain calls must therefore be latched with flags and unlatched again for any new pass. For simple tasks, such as this, it is quite obvious. However, for more complex applications a certain amount of caution should be exercised.

It is now left to the programmer to select the most appropriate of the two programming methods demonstrated.

7. Commands

Commands and data are transferred between the PCD's CPU and the H120 by the user program, as one or more 8-bit bytes.

A diskette is provided which contains pre-coded function blocks (FBs) to handle all the necessary procedures, so the user need not be concerned with control details. However, the user should be familiar with the major functions of the module, and the command sequences.

The command bytes are listed on the following pages. Each is shown with it's decimal command code and a short description. The name of an FB to use for each command is also shown.

A description of each individual command group is given in the following chapters. Standard examples for typical operations are shown in section 8. These can be incorporated directly into a program, and adapted for specific applications where necessary.

7.1 Command summary

Main function blocks

The 5 main function blocks, described in section 7.2, can be found in file H1FB_12 (for counter module 1, systems 1 and 2), of the PCD9.H1E1 software package. (Files for counter module 2, systems 3 and 4, are found in H1FB_34).

The FBs are called from the user program. The codes indicated on the following pages are given as parameters. Instead of "x", the number of the system to which the command is addressed should be used, e.g. 1 or 2. If more than one H120 module is present in the system, system numbers 3..8 or above can be used.

WF_x	Write Function
WD_x	Write Divider (Divider Remainder)
WCR_x	Write Counter/Register
RCF_x	Read Counter and Flags
RDR_x	Read Divider Remainder

Code	Command	FB Name
0	Read Counter and Flags	RCF_x
8	Read Divider Remainder	RDR_x
(16	Read Output Flags	ROF_x *)
(24	Read Status Flags	RSF_x *)
32	Write To Counter, Positive	WCR_x
33	Write To Counter, Negative	WCR_x
36	Write To Preset, Positive	WCR_x
37	Write To Preset, Negative	WCR_x
40	Write To Register 1, Positive	WCR_x
41	Write To Register 1, Negative	WCR_x
44	Write To Register 2, Positive	WCR_x
45	Write To Register 2, Negative	WCR_x
64	Write To Divider	WD_x
65	Write To Divider Remainder	WD_x

all commands listed below use FB "WF_x"

96	Bypass Divider; Disable Outputs	WF_x
97	Bypass Divider; Disable Outputs; Reset Overflow Flag	WF_x
98	Bypass Divider; Enable Outputs	WF_x
99	Bypass Divider; Enable Outputs; Reset Overflow Flag	WF_x
104	Divide IN-A; Disable Outputs	WF_x
105	Divide IN-A; Disable Outputs; Reset Overflow Flag	WF_x
106	Divide IN-A; Enable Outputs	WF_x
107	Divide IN-A; Enable Outputs; Reset Overflow Flag	WF_x
112	Divide IN-B; Disable Outputs	WF_x
113	Divide IN-B; Disable Outputs; Reset Overflow Flag	WF_x
114	Divide IN-B; Enable Outputs	WF_x
115	Divide IN-B; Enable Outputs; Reset Overflow Flag	WF_x
120	Divide IN-A and IN-B; Disable Outputs	WF_x
121	Divide IN-A and IN-B; Disable Outputs; Reset Overflow Flag	WF_x
122	Divide IN-A and IN-B; Enable Outputs	WF_x
123	Divide IN-A and IN-B; Enable Outputs; Reset Overflow Flag	WF_x

*) for internal use only in Routine "RCF"

Code	Command	FB Name
128	Output CCO: C < 0	WF_x
129	Output CCO: C ≤ 0	WF_x
130	Output CCO: C = 0	WF_x
131	Output CCO: C ≥ 0	WF_x
132	Output CCO: C > 0	WF_x
133	Output CCO: C < 0 Latched	WF_x
134	Output CCO: C = 0 Latched	WF_x
135	Output CCO: C > 0 Latched	WF_x
136	Output R1CO: C < R1	WF_x
137	Output R1CO: C ≤ R1	WF_x
138	Output R1CO: C = R1	WF_x
139	Output R1CO: C ≥ R1	WF_x
140	Output R1CO: C > R1	WF_x
141	Output R1CO: C < R1 Latched	WF_x
142	Output R1CO: C = R1 Latched	WF_x
143	Output R1CO: C > R1 Latched	WF_x
144	Output R2CO: C < R2	WF_x
145	Output R2CO: C ≤ R2	WF_x
146	Output R2CO: C = R2	WF_x
147	Output R2CO: C ≥ R2	WF_x
148	Output R2CO: C > R2	WF_x
149	Output R2CO: C < R2 Latched	WF_x
150	Output R2CO: C = R2 Latched	WF_x
151	Output R2CO: C > R2 Latched	WF_x
160	Input Filters: 16 Hz - 30 ms	WF_x
161	Input Filters: 166 Hz - 3 ms	WF_x
162	Input Filters: 1.6 kHz - 300 µs	WF_x
163	Input Filters: 16 kHz - 30 µs	WF_x
164	Input Filters: 166 kHz - 3 µs	WF_x
165	Input Filters: 333 kHz - 0.1 µs	WF_x
168	Divider IN-A: Single Shot; 100 Hz	WF_x
169	Divider IN-A: Single Shot; 1 kHz	WF_x
170	Divider IN-A: Single Shot; 10 kHz	WF_x
171	Divider IN-A: Single Shot; 100 kHz	WF_x
172	Divider IN-A: Single Shot; 1 MHz	WF_x
173	Divider IN-A: Single Shot; 10 MHz	WF_x
176	Divider IN-B: Single Shot; 100 Hz	WF_x
177	Divider IN-B: Single Shot; 1 kHz	WF_x
178	Divider IN-B: Single Shot; 10 kHz	WF_x
179	Divider IN-B: Single Shot; 100 kHz	WF_x
180	Divider IN-B: Single Shot; 1 MHz	WF_x
181	Divider IN-B: Single Shot; 10 MHz	WF_x

Code	Command	FB Name
184	Divider IN-A: Continuous; 100 Hz	WF_x
185	Divider IN-A: Continuous; 1 kHz	WF_x
186	Divider IN-A: Continuous; 10 kHz	WF_x
187	Divider IN-A: Continuous; 100 kHz	WF_x
188	Divider IN-A: Continuous; 1 MHz	WF_x
189	Divider IN-A: Continuous; 10 MHz	WF_x
192	S.C.D. is Up; Up / Down Counting; No Preset	WF_x
193	S.C.D. is Up; Up / Down Counting; Preset if Overflow	WF_x
194	S.C.D. is Up; Up / Down Counting; Preset if C=R1	WF_x
195	S.C.D. is Up; Up / Down Counting; Preset if C=0	WF_x
196	S.C.D. is Up; Up / Down Counting; Preset if Preset-IN=H	WF_x
197	S.C.D. is Up; Up / Down Counting; Preset on positive Edge	WF_x *)
199	S.C.D. is Up; Up / Down Counting	WF_x *)
200	S.C.D. is Up; Up Counting only; No Preset	WF_x
201	S.C.D. is Up; Up Counting only; Preset if Overflow	WF_x
202	S.C.D. is Up; Up Counting only; Preset if C=R1	WF_x
203	S.C.D. is Up; Up Counting only; Preset if C=0	WF_x
204	S.C.D. is Up; Up Counting only; Preset if Preset-IN=H	WF_x
205	S.C.D. is Up; Up Counting only; Preset on positive Edge	WF_x *)
207	S.C.D. is Up; Up Counting only;	WF_x *)
208	S.C.D. is Down; Up / Down Counting; No Preset	WF_x
209	S.C.D. is Down; Up / Down Counting; Preset if Overflow	WF_x
210	S.C.D. is Down; Up / Down Counting; Preset if C=R1	WF_x
211	S.C.D. is Down; Up / Down Counting; Preset if C=0	WF_x
212	S.C.D. is Down; Up / Down Counting; Preset if Preset-IN=H	WF_x
213	S.C.D. is Down; Up / Down Counting; Preset on positive Edge	WF_x *)
215	S.C.D. is Down; Up / Down Counting	WF_x *)
216	S.C.D. is Down; Down Counting only; No Preset	WF_x
217	S.C.D. is Down; Down Counting only; Preset if Overflow	WF_x
218	S.C.D. is Down; Down Counting only; Preset if C=R1	WF_x
219	S.C.D. is Down; Down Counting only; Preset if C=0	WF_x
220	S.C.D. is Down; Down Counting only; Preset if Preset-IN=H	WF_x
221	S.C.D. is Down; Down Counting only; Preset on pos. Edge	WF_x *)
223	S.C.D. is Down; Down Counting only	WF_x *)

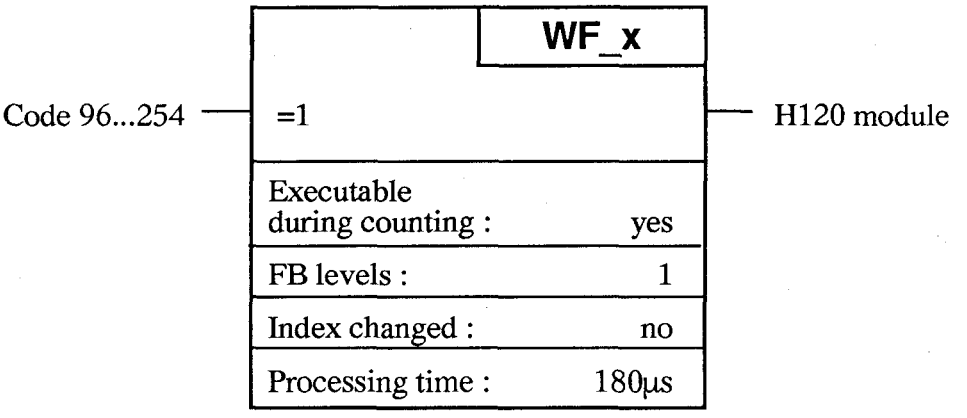
*) From revision "C" only

Code	Command	FB Name
224	Count Mode: x4	WF_x
226	Count Mode: x2	WF_x
228	Count Mode: x1	WF_x
230	Count Mode: Dual Count (+/-)	WF_x
232	Count Mode: Dual Count (+)	WF_x
234	Count Mode: Single Count	WF_x
236	Count Mode: Disabled Count	WF_x
240	Count Mode: Enabled Count if Output CCO = H	WF_x
242	Count Mode: Enabled Count if Output R1CO = H	WF_x
244	Count Mode: Enabled Count if Output R2CO = H	WF_x
248	Count Mode: Enabled Count between positive Edges on IN-B	WF_x
250	Count Mode: Enabled Count if IN-B = H	WF_x
252	Count Mode: Enabled Count between negative Edges on IN-B	WF_x
254	Count Mode: Enabled Count if IN-B = L	WF_x

7.2 Function blocks

WF Function : - Write Function **WF**

Software package : PCD9.H1E1 File : H1FB_xy.SRC



This FB writes commands to the H120 module. The 8-bit command code (96..254 decimal) is written to the H120 module's inputs 8..15.

When calling the FB, indicate which system (_1, _2 ...) the FB is to address.

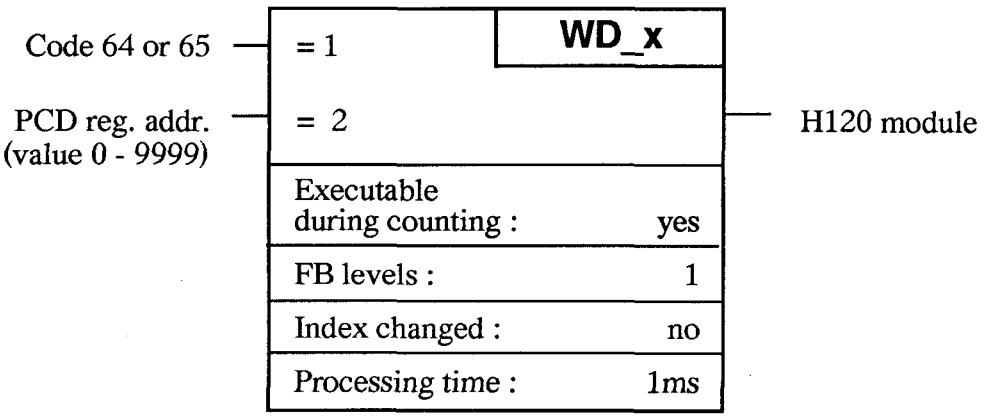
e.g.: **CFB WF_2**
163 ; Input Filters: 16kHz - 30µs

sets the input filter of system 2 to ≤ 16kHz

WD

Function : - Write Divider, Write Divider Remainder **WD**

Software package : PCD9.H1E1 File : H1FB_xy.SRC



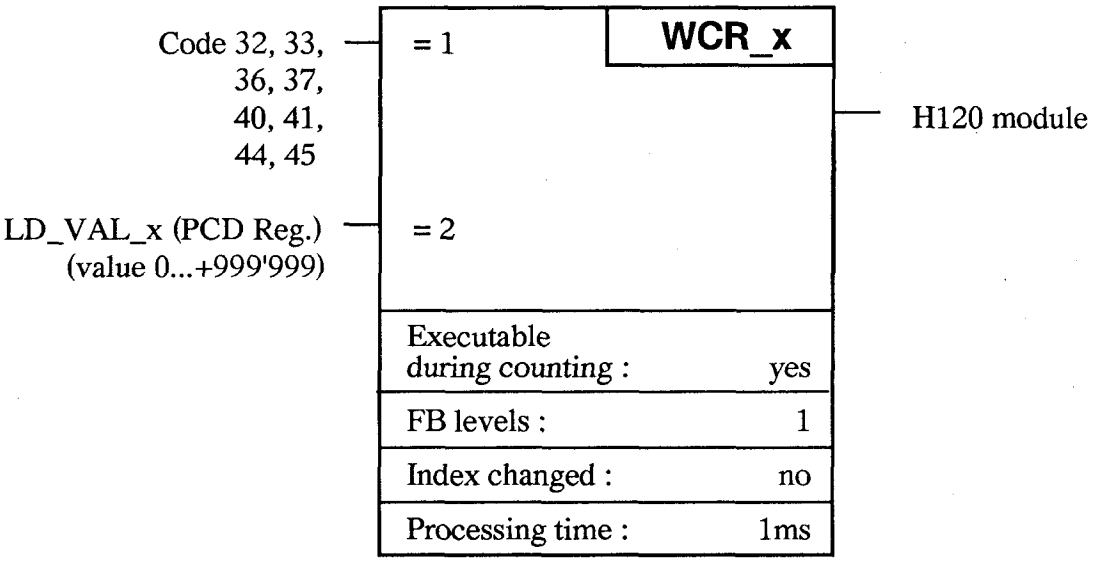
This FB loads the divider or divider remainder with a value between 0..9999 into the H120 module.

When calling the FB, indicate which system (_1, _2 ...) the FB is to address.

```
e.g.:  LD     LD_VAL_1    ; predefined work register
       400                ; value to be loaded
       CFB    WD_1
       64                 ; Write to Divider
       LD_VAL_1
```

WCR Function : - Write Counter/Register WCR

Software package : PCD9.H1E1 File : H1FB_xy.SRC



This FB writes a value 0..999'999, positive or negative, into counter register 1 or 2, or to the preset register in the H120 module.

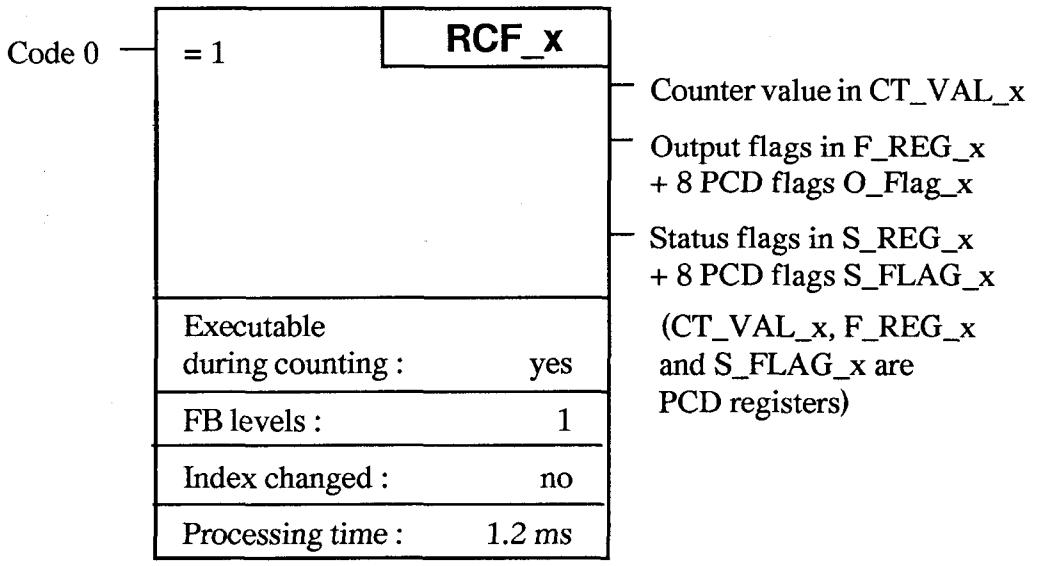
While loading the counter, the counting process must be stopped.

When calling the FB, indicate which system (_1, _2 ...) the FB is to address.

```
e.g.: LD      LD_VAL_2      ; predefined work register
      224466                ; value to be loaded
      CFB     WCR_2
      40                    ; Write to Register 1, pos.
      LD_VAL_2
```

RCF Function : - Read Counter and Flags RCF

Software package : PCD9.H1E1 File : H1FB_xy.SRC



This FB reads the current counter value, the output status, and status flags from the H120 module.

It is designed so that the most important output and status flags can be polled directly by the user program:

R1CO_x, R2CO_x, CCO_x, OVFLF_x, ENAB_x,
INB_x, INA_x und PRES_x

When calling the FB, indicate which system (_1, _2 ...) the FB is to address.

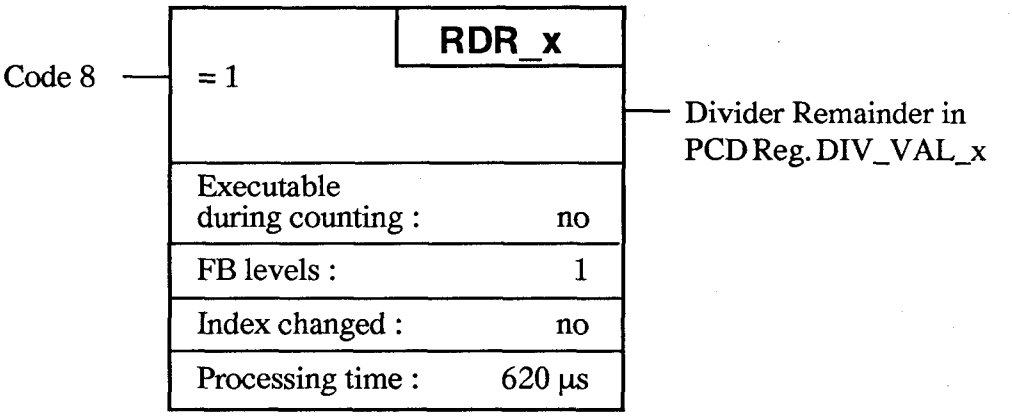
```
e.g.:   TR      120
        CFB     RCF_1
                0       ; Read Counter and Flags
        STL     R1CO_1  ; this TR will only be left
                        ; for the next ST when
        ETR             ; R1CO_1 = L .
```

RDR

Function : - Read Divider Remainder

RDR

Software package : PCD9.H1E1 File : H1FB_xy.SRC



This FB reads the value of the divider remainder from the H120 module into a PCD register. This FB can only be called when counting is inactive.

When calling the FB, indicate which system (_1, _2 ...) the FB is to address.

```
e.g.: CFB     RDR_3
      8                ; Read Divider Remainder

      CMP     DIV_VAL_3  ; Compare divider remainder
              RESULT     ; with a value

      CPB  Z  RES_OK
```

7.3 Write commands

As the commands listed on the preceding pages show, the H120 module's command set consistst of mostly "write" commands, and two "read" commands: "Read Counter and Flags" and "Read Divider Remainder".

The "write" commands can be further sub-divided into groups:

a) Commands for writing 6 digit values

- Write to Counter, pos/neg
- Write to Preset, pos/neg
- Write Register 1/2, pos/neg

This group uses the "WCR_x" function

b) Commands for writing 4 digit values

- Write to Divider
- Write to Divider Remainder

This group uses the "WD_x" function

c) Commands for writing individual functions.
This group can be further divided into sub-groups:

- Input filter commands	(7.3.1)
- Commands for signal selection (count mode)	(7.3.2)
- Commands for output function	(7.3.3)
- Commands for counting direction	(7.3.4)
- Commands for the frequency divider	(7.3.5)
- Enable count commands	(7.3.6)
- Preset register commands	(7.3.7)
- Reset	(7.3.7)

This group uses the "WF_x" function

7.3.1 Input filter commands

Commands of the type "Input Filter: 166Hz - 3mS" set the clock frequency of the IP1 and IP2 filters. (IP1 and IP2 are the ASIC inputs of the corresponding IN-A and IN-B module inputs). Filter outputs only switch over if a high/low signal is present at the inputs for at least 3 consecutive internal clock cycles.

If the command "Input Filter: 333kHz - 0.1µS" (internal clock frequency = 10MHz) is entered, the input filters are bypassed.

Input filter command limiting times and frequencies are given in the following table.

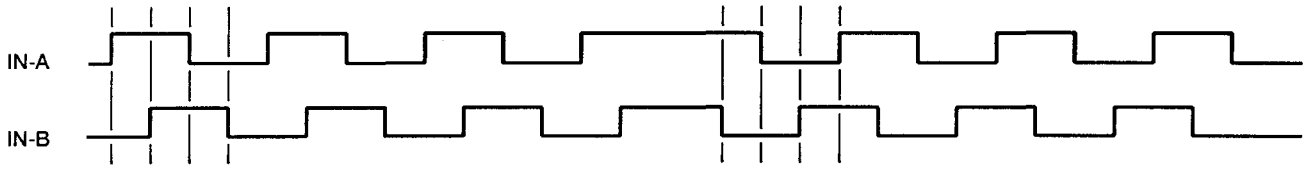
Command	Internal Clock Frequency	Limit at : Minimum Level Period	Limit at : Maximum Frequency
Input Filter : 16 Hz	100 Hz	30 ms	16 Hz
Input Filter : 166 Hz	1 kHz	3 ms	166 Hz
Input Filter : 1.6 kHz	10 kHz	300 µs	1,6 kHz
Input Filter : 16 kHz	100 kHz	30 µs	16 kHz
Input Filter : 166 kHz	1 MHz	3 µs	166 kHz
Input Filter : 333 kHz	10 MHz	0,1 µs *	333 kHz *

* The frequency limit derives from the interference suppression filters at inputs IN-A and IN-B.

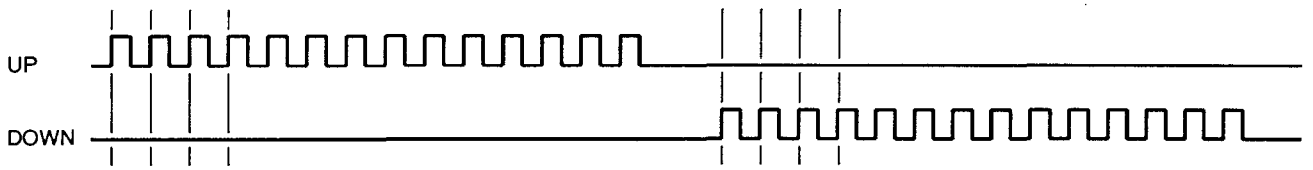
7.3.2 Commands for signal selection, "count mode" commands

These commands select the input signal type, or define how the input signals are to be processed. The following diagrams show the up/down counting for each command.

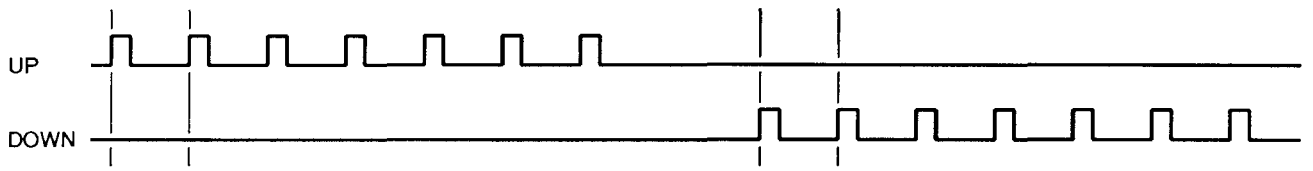
INPUT SIGNALS



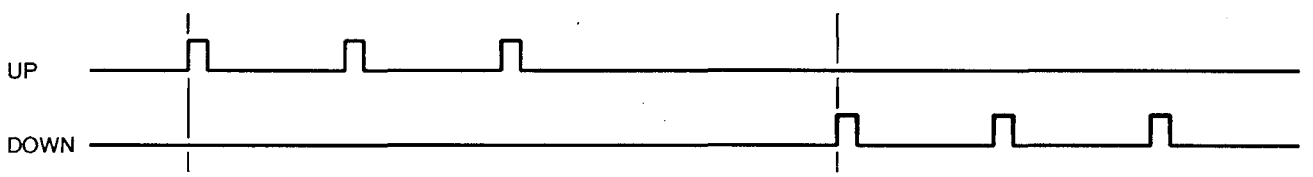
MODE x4



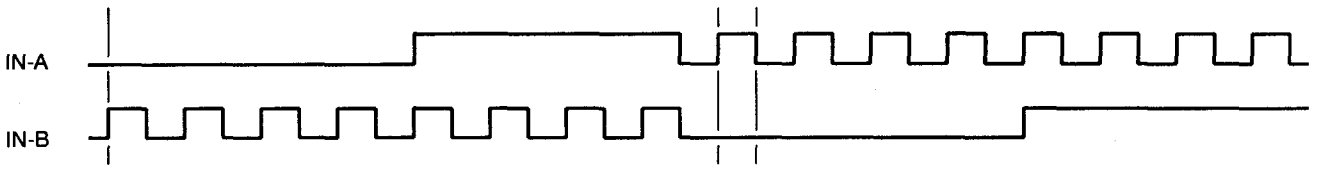
MODE x2



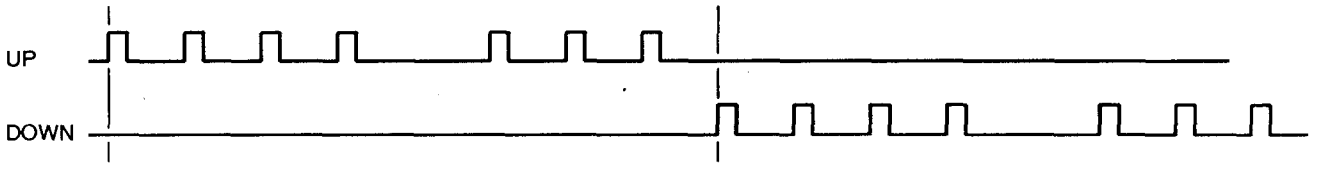
MODE x1



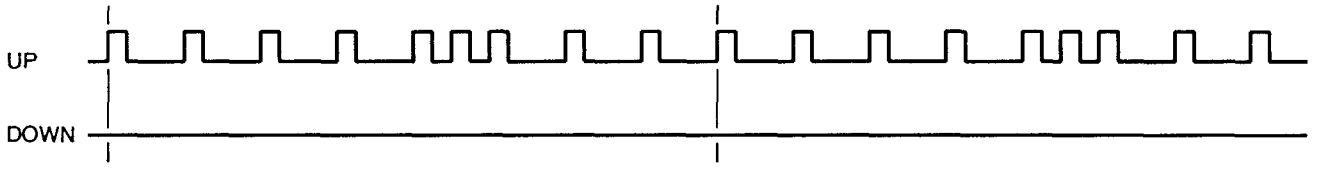
INPUT SIGNALS



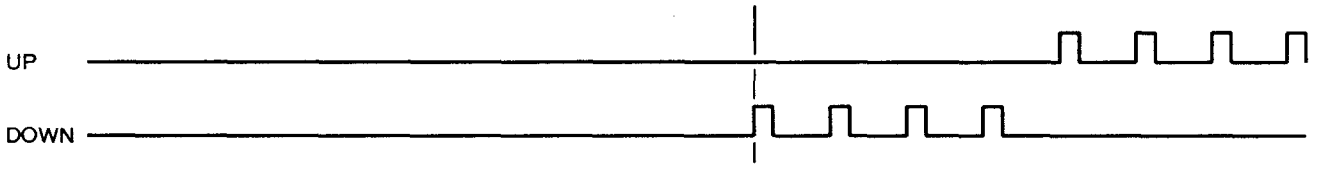
DUAL COUNT (+/-)



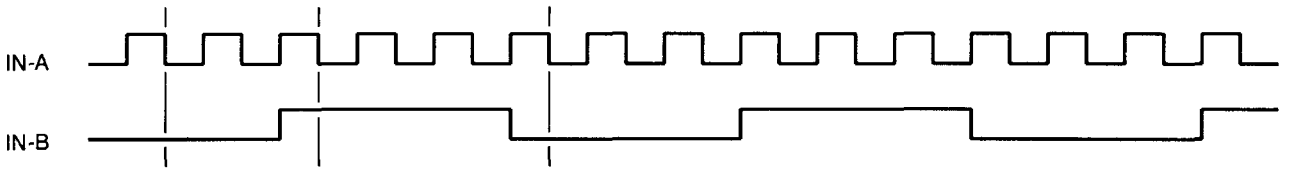
DUAL COUNT (+)



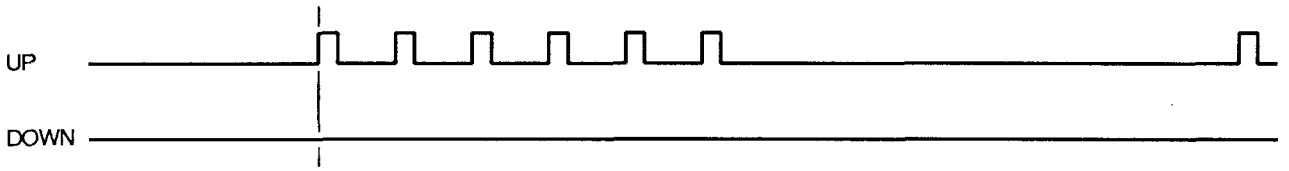
SINGLE COUNT



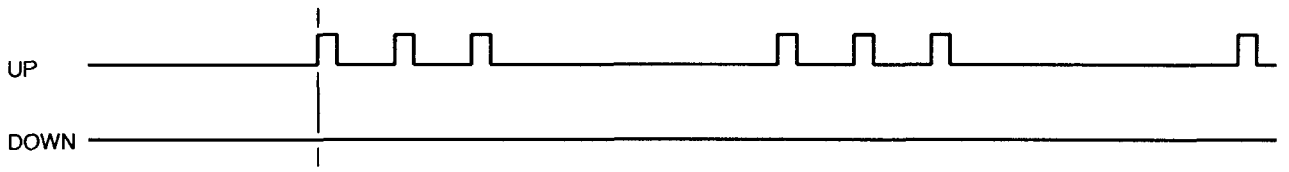
INPUT SIGNALS



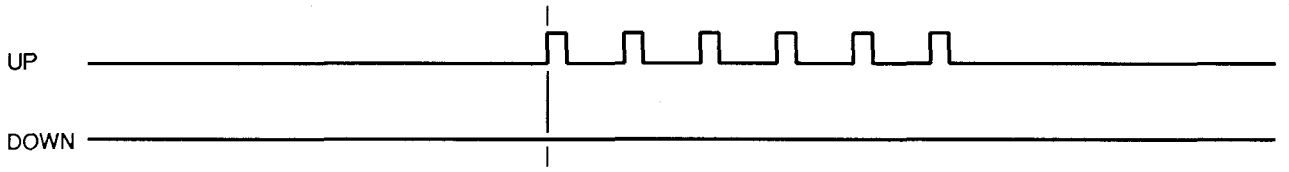
ENABLED COUNT IN-B BETWEEN POSITIVE EDGES



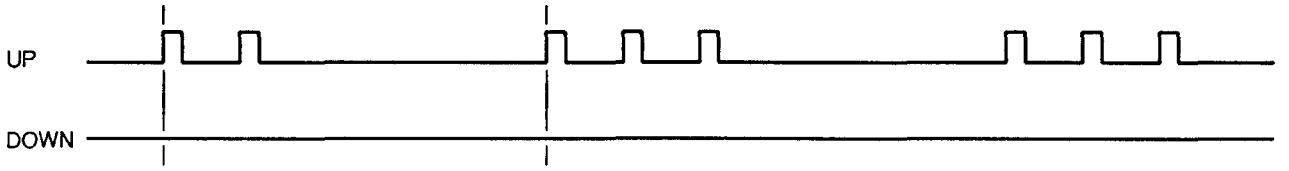
ENABLED COUNT IN-B HIGH



ENABLED COUNT IN-B BETWEEN NEGATIVE EDGES



ENABLED COUNT IN-B LOW



Summary :

for 2-phase shaft encoder signals:

Mode x4 : One count on every IN-A and IN-B edge
Mode x2 : One count on every IN-A edge
Mode x1: One count on every negative direction IN-A edge

for general signals:

Dual Count (+/-)	:	IN-A counts down, IN-B counts up	(pos. edges)
Dual Count (+)	:	IN-A and IN-B count up	(pos. edges)
Single Count	:	IN-A counts up if IN-B = H and down if IN-B = L	(neg. edges)

for enabled signals:

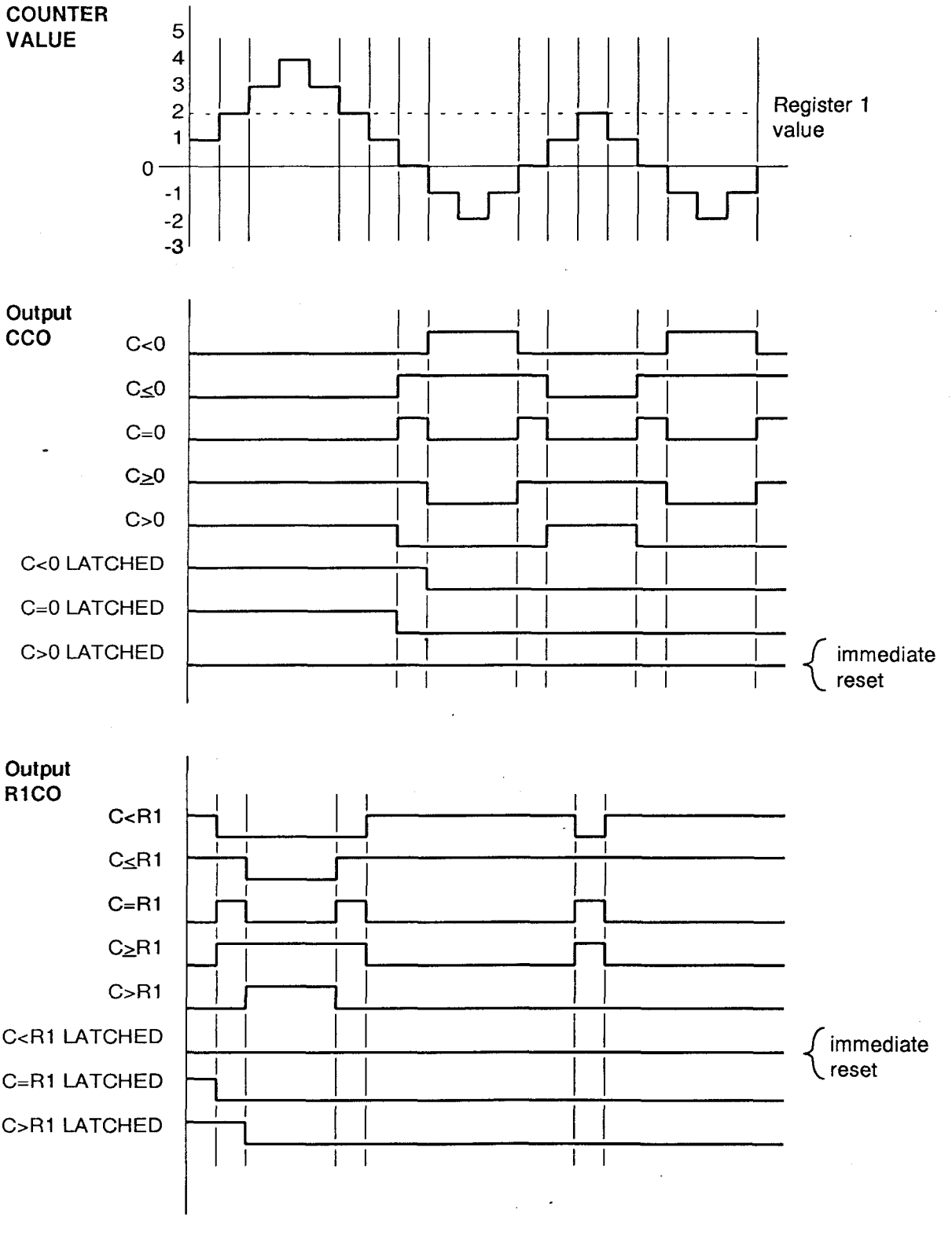
Enable Count: IN-A counts ...

- when output CCO = H
- when output R1CO = H
- when output R2CO = H
- between positive edges on IN-B
- when IN-B = H
- between negative edges on IN-B
- when IN-B = L

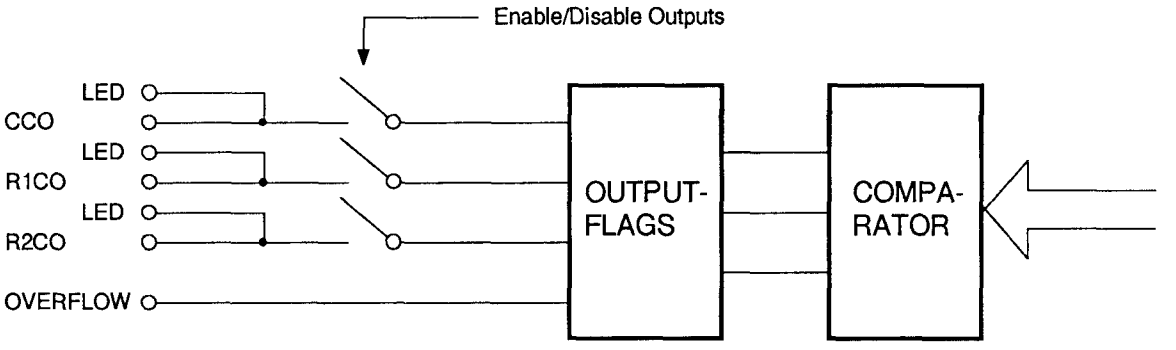
7.3.3 Write commands for outputs

Three H120 module outputs have a user-definable function. These are outputs "CCO", "R1CO" and "R2CO".

The following diagrams show typical operation of these outputs.



After power up, outputs "CCO", "R1CO" and "R2CO" are inactive, i.e. all these outputs are low. After defining individual output functions, output flags are set accordingly. The outputs remain low. Only when the "Enable Output" command is executed are the output flags switched to the outputs, as the diagram shows, using the three (virtual) switches. The "Disable Outputs" command opens the switches again. The logical state of output flags is unchanged.



Outputs defined as unlatched remain valid until there is a (hardware) reset or until they are redefined. For example, if the definition is "Output R1CO: C > R1", the output flag will be high whenever the counter value exceeds the value in register 1.

This is not the same for latched outputs: if a function is defined, e.g. "Output R2CO: C = R2 latched", output flag R2CO immediately becomes high (unless C is equal to R2). If the condition C = R2 is met, output flag R2CO is low and remains low. To repeat the function, it is necessary to execute definition "Output R2CO: C = R2 latched" once again.

N.B.: If definition "Output R2CO: C = R2 latched" is executed and, for whatever reason, the condition C = R2 is not reached, output flag R2CO remains high and, following an "Enable Output" command, this output flag is switched through to the output.

There is no unlatched command for resetting output flags. If a latched output flag has to be reset, the condition for switching off must be forced, for this example load R2 equal to C (easiest if C = 0), or a new definition should be made which is certain not to be met.

The Overflow Flag (OF_F) is set whenever the counter value is greater than +999'999 or less than -999'999. There are 3 ways to reset OF_F: using the command "Reset Overflow Flag"; loading a new value directly into the counter; or executing the command "Preset if Overflow".

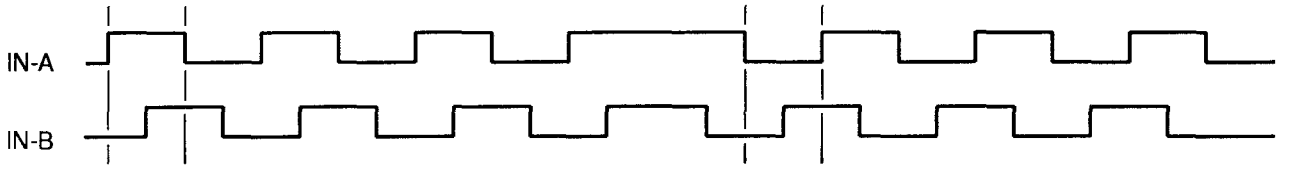
7.3.4 Commands for counting direction, "S.C.D." commands

(S.C.D. = Standard Counting Direction)

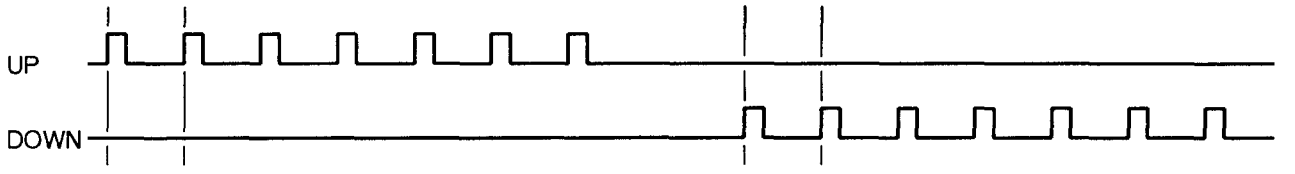
In the following example, the "Count Mode" command "x2" is intended show how an "S.C.D." command can modify the counting mode (Dual Count +/-, Enabled Count etc.).

These commands also set the counting direction when using the internal clock generator.

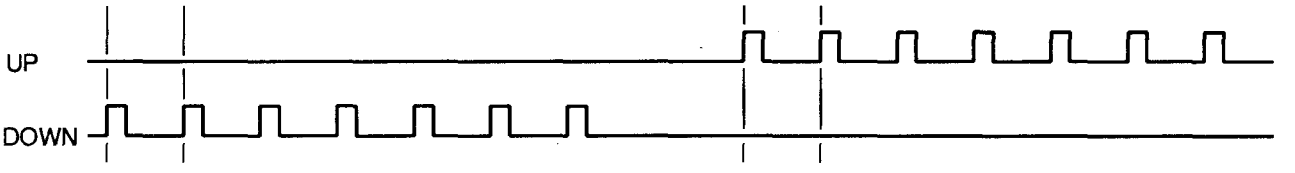
INPUT SIGNALS



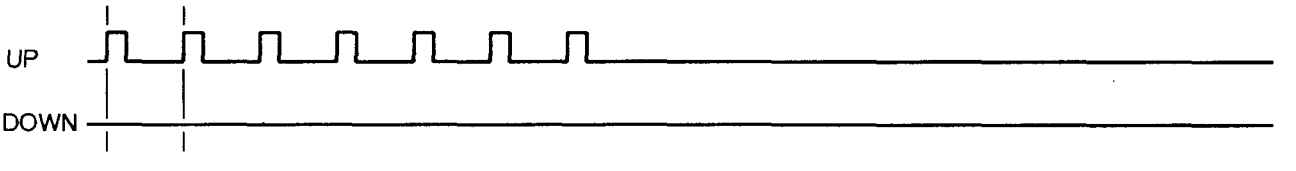
MODE x2 ; S.C.D. is Up ; Up / Down Counting



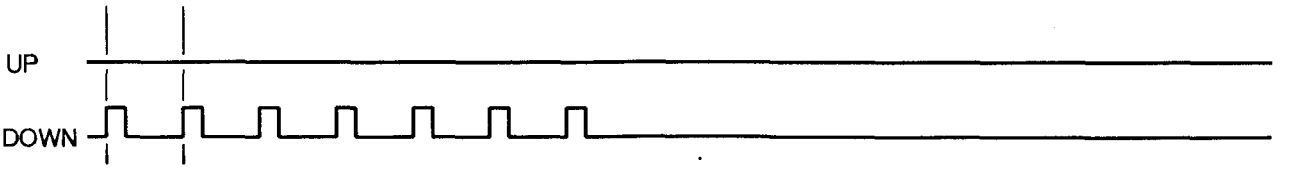
MODE x2 ; S.C.D. is Down ; Up / Down Counting



MODE x2 ; S.C.D. is Up ; Up Counting only



MODE x2 ; S.C.D. is Down ; Down Counting only



7.3.5 Commands for the divider

Three different types of command are used in connection with the divider:

1. Read or Write Divider value

   - Write To Divider
   - Write To Divider Remainder
   - Read Divider Remainder

2. Divider control

   - Bypass Divider
   - Divide IN-A
   - Divide IN-B
   - Divide IN-A and IN-B

3. Divider input control

   - Divider IN-A; Single Shot; 10 MHz
   - Divider IN-B; Single Shot; 10 kHz
   - Divider IN-A; Continuous; 100 Hz

Re 1. On power-up, the divider value is 0. This means that count pulses pass straight through the divider. If a value of 1 is loaded into the divider using the "Write To Divider" command, the first pulse received is stored (divider remainder becomes 1) before a pulse is output from the divider. This means that for 2 input pulses, only 1 is output, corresponding to a ÷2 function. Whatever divider value is entered, the divider uses it as "divider value + 1". For example, if the value 1249 is loaded, the 1250th pulse will be output, i.e. division by 1250. The highest value which can be entered is 9999, corresponding to ÷10'000.

If the divider has a value of 49 (÷50) and 120 pulses have been entered, 2 pulses will have been output to the counter and the remaining 20 pulses are stored in the divider remainder as a value of 20. This value can be read using the "Read Divider Remainder" function, but only when counting has been stopped.

It is also possible to write a "start-up" value to the divider. For example, to divide input pulses by 100 (divider value = 99) but

with the first output pulse going to the counter after the 70th input pulse, this can be achieved by using a "Write To Divider Remainder" command with a value of 30.

Further information regarding type 1 commands:

The "Write to Divider" command works two different ways. If count mode is inactive, or if it has not yet been selected, or following "Disable Count", the "Write to Divider" command is executed immediately.

If a count mode is active, "Write to Divider" is only executed when the divider remainder is 0. It is important to be aware of this when the internal clock is run to the divider to generate a pulse sequence and when, during pulse output, frequency has to be changed to produce ramps, for example. Only the divider value is changed in the user program for this. The fact that the new value is only adopted once the divider remainder is 0 ensures that the transition from old to new output frequency cannot take place until the end of a complete period.

During the time when a new divider value is required and is being loaded, the internal busy flag is high. This busy flag is taken into account by the standard "Write Divider" routine. (The user does not need to concern himself with it).

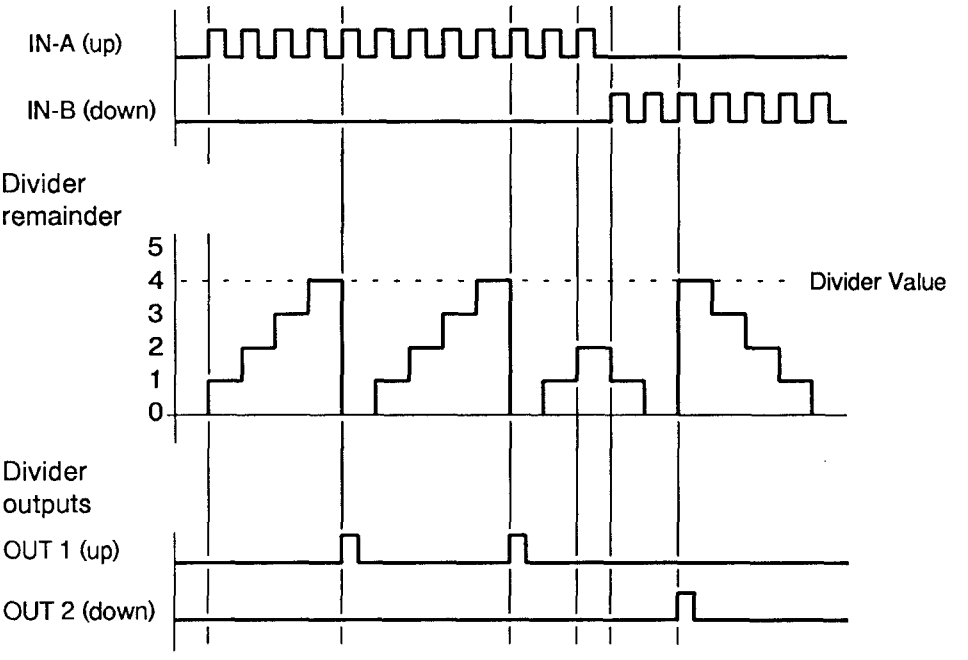
Re 2. To reduce the number of commands needed to control the counter ASIC, the divider control commands have been combined with the output commands. This means that if outputs are to be enabled without activating the divider, the "Bypass Divider" option must be used. The ASIC is always powered up in "Bypass Divider" mode.

The "Divide IN-A" option directs all UP count pulses through the divider while letting all DOWN count pulses bypass the divider. This option is used in timing applications.

The "Divide IN-B" option directs all DOWN count pulses through the divider while letting all UP count pulses bypass the divider. This option is used in frequency measurement applications.

The "Divide IN-A AND IN-B" option can be used for scaling the counting operation or increasing the effective counting range of the system. The effect of this command is to feed both UP and DOWN count pulses through the divider. If this option is used with a divider value of of 9999, the counting range is -9'999'990'999 to +9'999'999'999.

The following diagram shows divider function. In this case, the divider value is 4, i.e. every 5th input pulse is fed to the counter. The divider can be used in all count modes.



Re 3. A selected frequency from the internal clock generator can be fed into the divider, providing one of the "Divide IN-A", "Divide IN-B" or "Divide IN-A and IN-B" commands has already been executed.

For example, if "Divide IN-A" is entered and followed by "Divider IN-A; Single Shot; 10 MHz", a 10 MHz signal will be connected to the divider's UP input. The "Single Shot" part of the command activates the single shot part of the enable circuitry, so that a single time measurement can be made, initiated by this command.

Frequency measurement can be achieved by using the command "Divider IN-B; Single Shot; 10 kHz". The appropriate "Divide" command must be issued first.

An output pulse or frequency can be generated with a command like "Divider IN-A; Continuous; 100 Hz". In this case, 100 Hz would be the primary frequency, and would then be divided to produce the desired output frequency.

See also section 8.3: Generator

7.3.6 Enable count commands

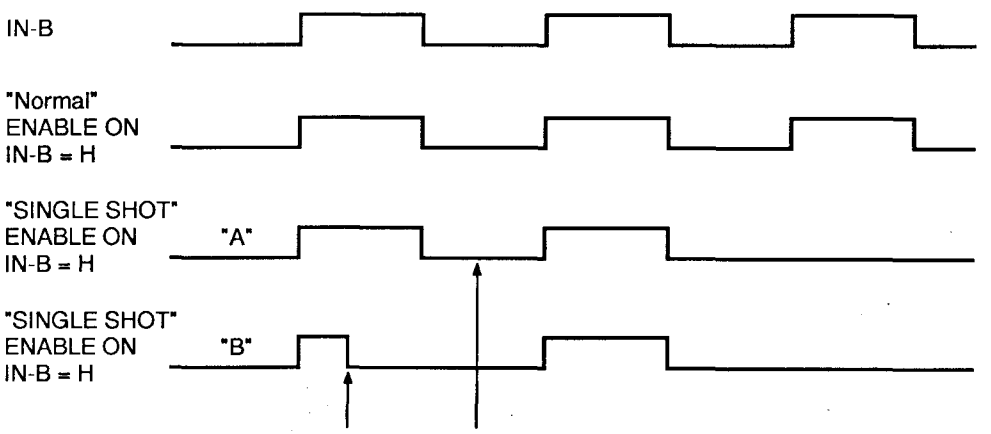
The "enable count" commands are a subset of the signal selection and count mode commands. A graphical representation of their operation is shown in section 7.3.2. The "enable count" commands are listed below:

Enabled count - Signals at input "IN-A" count up...
(negative edges)

- when output CCO = H
- when output R1CO = H
- when output R2CO = H
- between positive edges at IN-B
- when IN-B = H
- between negative edges at IN-B
- when IN-B = L

For the first 3 "Enable Count" modes, counting runs precisely parallel with the logical state of outputs "CCO", "R1CO" and "R2CO".

The reaction of the 4 other "Enable Count" modes is shown in the following diagram, using the example "Enable Count if IN-B = H".



"Enable" activates counting, i.e. the enable circuit serves as a gating circuit for incoming signals to be counted. If the enable is high, signals are counted, if the enable is low, signals are not counted. (See also logic diagram in Section 4.2). The logical state of the enable circuit can be polled in the program, in order for example to allow the program to run on only after a completed count or measurement. (See Section 7.4).

With a "Single Shot" command, e.g. "Divider IN-A; Single Shot; 100Hz", it is possible to influence enable circuit function. After a command of this kind only 1 enable cycle is executed. The next cycle will only be executed after a new single shot command. (See diagram, cases "A" and "B").

Important:

Since the enable condition has to be written before any single shot command, (see Section 7.6), enable may already be high when the single shot command is given. In this case the enable immediately becomes low and only reverts to high while the enable condition is met, which can lead to polling errors, e.g. to recognize the conclusion of a measurement. (Case "B" in diagram). Care should therefore be taken to ensure that, when executing the single shot command, the enable is low. (Case "A" in diagram).

If this is not possible within the context of a particular task, dynamic polling of the edges of the IN-B signal should be used to ensure that the correct enable signal is evaluated and that the program continues to run correctly.

In retrospect, it is evident that the single shot FUNCTION works correctly, but that the enable enable flag switches through as if the single shot command had not been given at all. This makes polling of the enable flag practically impossible in fast applications.

For the measurement of frequency and period length, combinations of divider and single shot commands are used. It would be excessive to describe all these applications in detail: the examples of typical applications in section 8 should be consulted instead.

7.3.7 Preset register commands

Similar to registers 1 and 2 and the counter, the preset register can be loaded with max. ± 999'999. (Write to Preset pos/neg)

Under certain conditions, the contents of the preset register can be loaded directly into the counter.

These conditions may be the following:

- Preset if Overflow
- Preset if C = R1
- Preset if C = 0
- Preset if Preset-IN = H
- Preset on Preset-IN, Positive Edge (*)
- No Preset, Preset disabled

Preset functions are combined with counting direction data (codes 192 - 223).

If a +24V signal is present at the preset input and the command "Preset if Preset-IN = H" is entered, the preset is executed continuously, thus preventing counting.

*) This dynamic preset command is only available from revision C onwards (revised ASIC circuit).

7.3.8 Reset

Although "Reset" is not actually a command, it is appropriate to mention it here.

When the PCD is powered up, both of the H120 module's systems are reset. This clears the counter, register and divider values, and resets the counting mode, status flags and outputs.

If jumper "RES/PRE" is in the default position "Q" (source operation) a +24V signal must always be connected to the reset input to allow the H120 module to operate. This means that the reset LED on the front panel will always be on.

If this +24V signal is removed and reapplied, everything is reset as for power-up of the unit, and the module must be re-configured.

NOTE: If the program is restarted with a "Restart Cold", the H120 is NOT reset.
An output should be used to reset the H120, which can be switched by code in the start-up XOB 16.

e.g.:

```
XOB   16

RES  O  47      ; Reset the H120
NOP             ; Short delay
NOP
NOP
NOP
NOP
SET  O  47      ; Remove the reset
.
.
.

ECOB
```

In this example, output 47 is connected to the reset input of the H120 module, the +24V signal can then be used to reset it from the program.

A reset can be issued by the application program at any time.

7.4 Read commands (RCF and RDR)

The "write" commands define the operating mode, counter and register values etc. Once these are defined, the H120 module then operates independantly, with immediate switching of outputs according to the mode, register and counter values. The outputs are available on the terminal connectors, and their states are indicated by the LEDs on the front panel. Without the "read" commands, the program could be controlled only by wiring these outputs to PCD inputs, where they could be polled in the usual way.

However, use of the H120 module is simplified by using "read" commands in the user program. These commands copy counter values into PCD registers, and copy the operating status (e.g. "R1 < C", "C < 0" etc.) into PCD flags, which can then be interrogated directly by the user program.

The "write" commands are normally only processed once per program cycle, but the "read" commands must be executed continuously to ensure the PCD flags and registers always contain the latest values. This requires management by the user program. Sections 6 and 8 show program structures which allow the "read" commands to be used correctly.

Internally, the module handles output flags and status flags in different ways. The former are updated in real time, the latter must be latched before reading. Values are read into registers by reading individual bytes from the module. Since these elements are all handled in different ways, a general-purpose "RCF" (Read Counter and Flags) function was developed. When the user program calls this function, the current counter value is copied to a PCD register (CT_VAL), and the status flags are copied to 16 PCD flags. The flags can then be polled by the user program, and the counter value can be accessed.

After processing the "RCF" function, the following information is available in register "CT_VAL", in addition to the counter value:

PCD flag	24	25	26	27	28	29	30	31
H1 flag	R1F	R1<C	R2F	R2<C	CF	C<0	OF_F	ENABLE
Symbol	R1CO	---	R2CO	---	CCO	---	OVFLF	ENAB

PCD flag	32	33	34	35	36	37	38	39
H1 flag	IN-B	IN-A	R2F	R2<C	CF	C<0	OF_F	PRESET
Symbol	INB	INA	---	---	---	---	---	PRES

The flag in the "Symbol" row of the above table can be examined by the user program.

e.g. STH R2CO or ANL INB

The definition file "H1DEF_12" must contain the definition of the base address "BF_12" of an array of 100 flags. In the above table, flags 24..39 in the "PCD flag" row are offsets into this array of flags. For example, if the base address was defined as 500, the "C < 0" status flags would be located at address 529. (The same status is also in flag 537, some values are present at two addresses due to the ASIC circuit's configuration.)

When testing a program, it is useful to write flags 24..31 to an output module after each "RCF" command, using the BITO instruction, so that the status can be viewed. The same data can also be viewed using the "Debugger" or the PCD8.P100 service unit. However, the status is only updated when the "RCF" function is executed.

Counting, and its control over the 3 outputs "CCO", "RCO" and "R2CO", is done by the counting/measuring module independantly, without being influenced by the user program. These 3 outputs control the user program. The cycle time of the user program influences when these outputs or flags can be processed. This doesn't effect the counting task, which is hardware controlled.

An additional read function is "RDR" (Read Divider Remainder). This transfers the 4-digit value into the PCD register "DIV_VAL".

Register 1, register 2 and the divider value CANNOT be read by the user program. These values are not changed by the module, and the loaded value can also be stored in a PCD register for later reference, if required.

7.5 External display using the PCA2.D14 display module

The PCD4.H120 module has 4 additional outputs for connection to two PCA2.D14 display modules. Each display module can show two 6-digit decimal values. The user can define which data is displayed. Displaying the current counter value and the preset register value is recommended.

Function blocks to control the display module are contained in file "H1FB_12". The instruction "CFB D14_xy" is used by the user program to conditionally or unconditionally call the display function, supplying three variables as parameters. Parameters 1 and 2 are the PCD registers to be shown on the upper and lower displays. The 3rd parameter selects display module 1 or 2 (ENABLE_1 or ENABLE_2). Different formats can be displayed, such as with sign bits or spaces, refer to the PCD2.D14 description. This requires some routines to be programmed outside the "H1DEF" file.

Display modules are connected in accordance with the diagram in section 5.1.

If not using a display module, and the outputs are used as general purpose controls, it should be note that the outputs "ENABLE_1" and "ENABLE_2" are logically inverted ("SET" switches the output off and "RES" switches it on). These outputs can carry only up to 100mA.

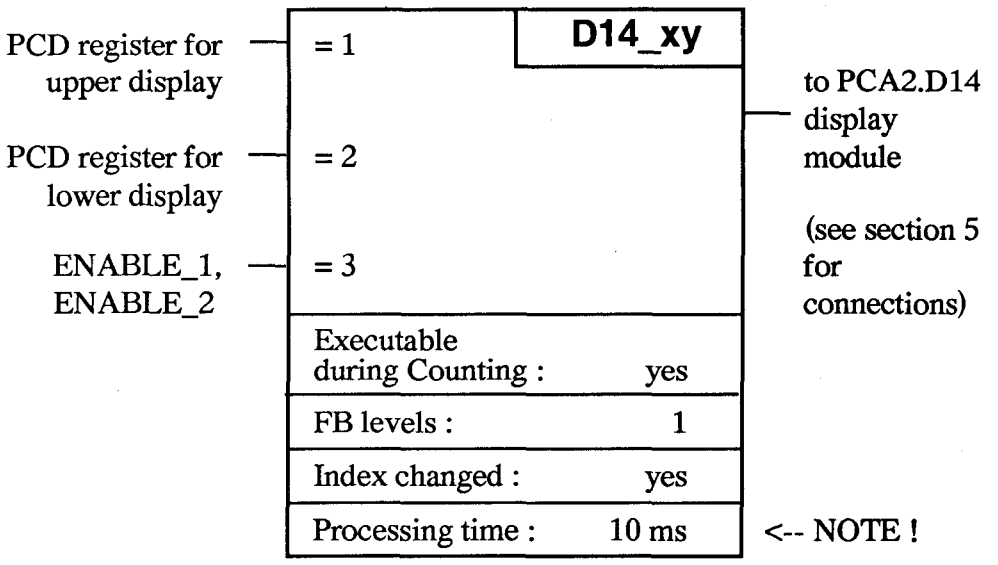
Note on FB D14_xy

Executing the D14 routine once takes about 10ms, which is rather a long time. The user program should therefore be designed to call this routine only as often as is really necessary. It is not necessary to refresh the display constantly, because the eye can't follow the display, and the user program is slowed down considerably. It is better to update the display every 0.5 second, every second, or at even longer intervals, or even not cyclically at all.

While the display routine is being processed, the user program can't do anything else. Of course, this doesn't affect the H120 module, as it runs independently. However, other parts of the program may be adversely affected. A display routine which runs in small stages, each handling a separate section of the display function is possible, but it would complicate the user program.

D14 Function : - Display on PCA2.D14 D14

Software package : PCD9.H1E1 File : H1FB_xy.SRC



This FB can be used, for example, to write the current counter value obtained with function "RCF" (Read Counter and Flags) to the PCA2.D14 display module.

Since this routine can delay the user program, it should not be called constantly, limit it to every 0.5 second or only call it when required (e.g. when a display value has changed).

```
e.g.: STL   T   0
      LDL   T   0
            5             ; 0,5 sec

      CFB   H   D14_12      ; Call every 0.5 sec
                CT_VAL_1    ; upper display
                LD_VAL_1    ; lower display
                ENABLE_1    ; System 1
```

7.6 Command entry order

The following rules concerning the order of commands must be followed when a sequence of commands is written to the H120 module, otherwise the system may not operate as desired.

1. A "Write to Counter" command must stand before any command which concerns the outputs and output flags CCO, R1CO and R2CO.

2. "Write To Preset" must precede any "Preset if ..." command.

3. "Write To Register 1" must precede "Preset if C=R1".

4. "Write To Divider" must precede "Set Count Mode", unless a new divider value will be written while counting.

5. "Write To Divider Remainder" must be done before "Write To Divider".

6. "Enabled Count" must precede any "Single Shot" command.

> The best way to create a successful user program for the counting and measuring module is to follow the programming examples in Chapter 8.

8. User program

8.1 Fundamental principles

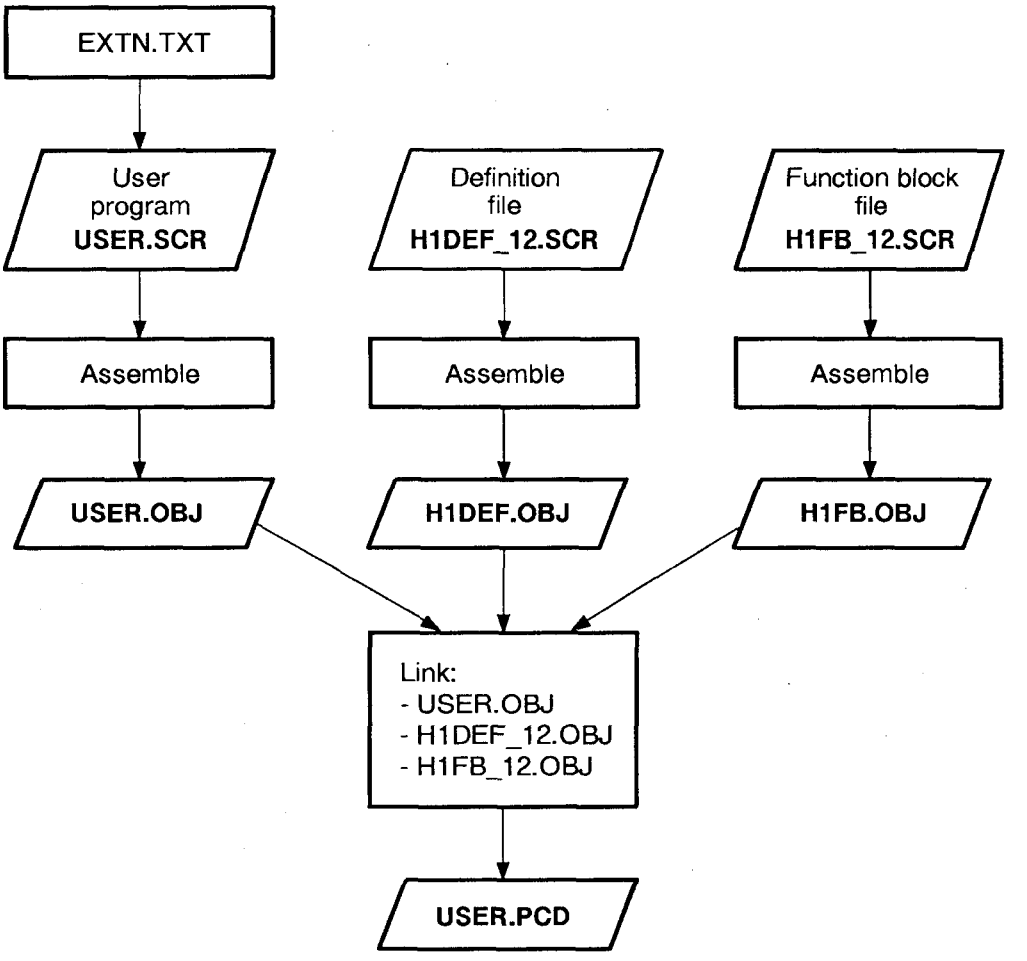
The preceding chapters outline the versatility of the PCD4.H120 counting and measuring module. The aim of this chapter is to demonstrate how some typical applications can be programmed using its wide range of features.

The function blocks, described in section 7, will be used. These are available on diskette as software package "PCD9.H1E1".

For counting module no. 1, with systems 1 and 2, the following files are used:

- EXTN.TXT
- H1DEF_12.SRC
- H1FB_12.SRC

The diagram below shows the standard structure of user programs for the PCD4.H120 counting module.



User programs for the PCD4.H120 counting and measuring module CANNOT be programmed with SAIA's "SEDIT" editor. An ordinary ASCII text editor must be used, e.g. IBM's PE or PE2.

- **EXTN.TXT** contains the list of externally defined symbols. This list should be copied or $included at the start of every user program.

```
; For systems 1 and 2:
;-------------------------------------------------------------------
  extn    wf_1,wd_1,wcr_1,rdr_1,rcf_1
  extn    ld_val_1,ct_val_1,div_val_1,f_reg_1,s_reg_1
  extn    r1co_1,r2co_1,cco_1,ovflf_1,enab_1,inb_1,ina_1,pres_1

  extn    wf_2,wd_2,wcr_2,rdr_2,rcf_2
  extn    ld_val_2,ct_val_2,div_val_2,f_reg_2,s_reg_2
  extn    r1co_2,r2co_2,cco_2,ovflf_2,enab_2,inb_2,ina_2,pres_2

  extn    d14_12,enable_1,enable_2
;-------------------------------------------------------------------
```

- **H1DEF_12.SRC** contains definitions of all symbols used and the base addresses of the H120 module, of a few registers, flags and the FBs. These four items of data MUST be defined, unless the default values are to be used.

  If the use of a "PCA2.D14" display module is planned, this must also be defined.

```
*******************************************************************

ba_12     equ   o   48   ;; Base address of the H120 module

bflag_12  equ   f   500  ;; Base address of 100 flags (F500-599)

br_12     equ   r   500  ;; Base register of 20 regist.(R500-519)

bfb_12    equ       800  ;; Base function block of 20 FBs (FB800-819)

d14_12    equ       999  ;; FB for display module PCA2.D14 ¦
f_d14_12  equ   f   400  ;; Base address of 48 flags       ¦ optional
c_d14_12  equ   c   999  ;; 1 scratch counter              ¦ for D14
r_d14_12  equ   r   999  ;; 1 scratch register             ¦

*******************************************************************
```

The definitions in the above table result in the following assignments:

- Base address of module: 48 (addresses 48 - 63, 16 addresses)
- 100 flags, addresses 500 - 599
- 20 registers, addresses 500 - 519
- 20 function blocks (FB), addresses 800 - 819

One or two PCA2.D14 display modules use FB 999, flags F 400 - 447, PCD counter C 999 and one PCD register R 999.

The programmer is responsible for avoiding clashes in the allocation of addresses between his program and SAIA's function blocks.

- **H1FB_12.SRC** contains all the FBs for execution of commands in the counting and measuring module.

This file must not be changed !

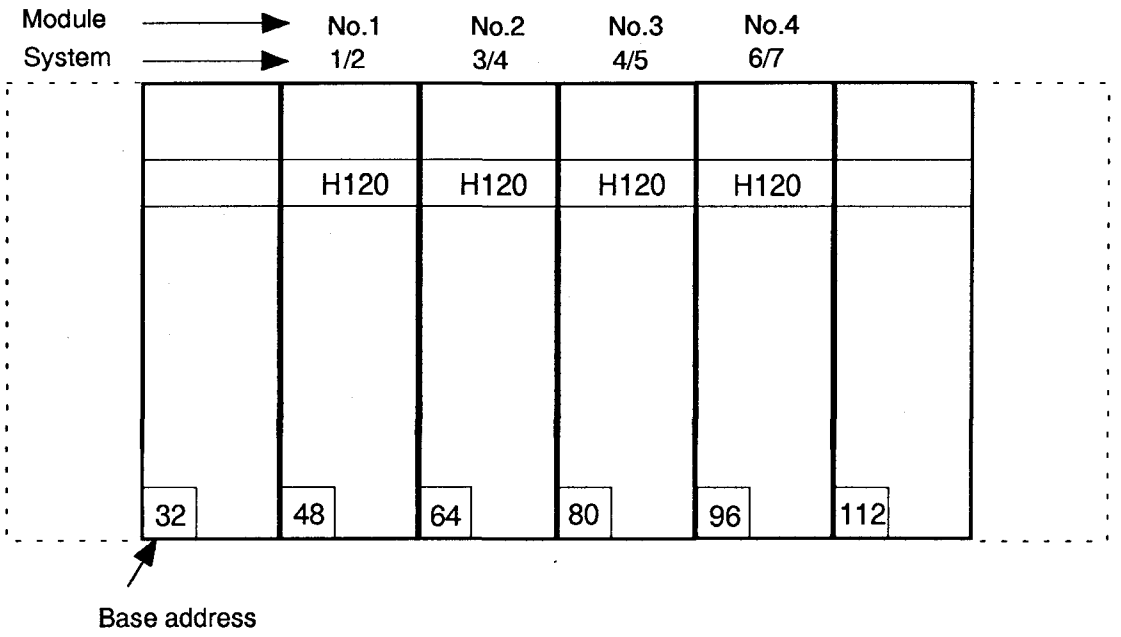
All user program commands with format "abc_1" e.g. CFB WF_1, refer to system 1, whereas all "abc_2" commands e.g. STH R1CO_2, refer to system 2 in the PCD4.H120 module.

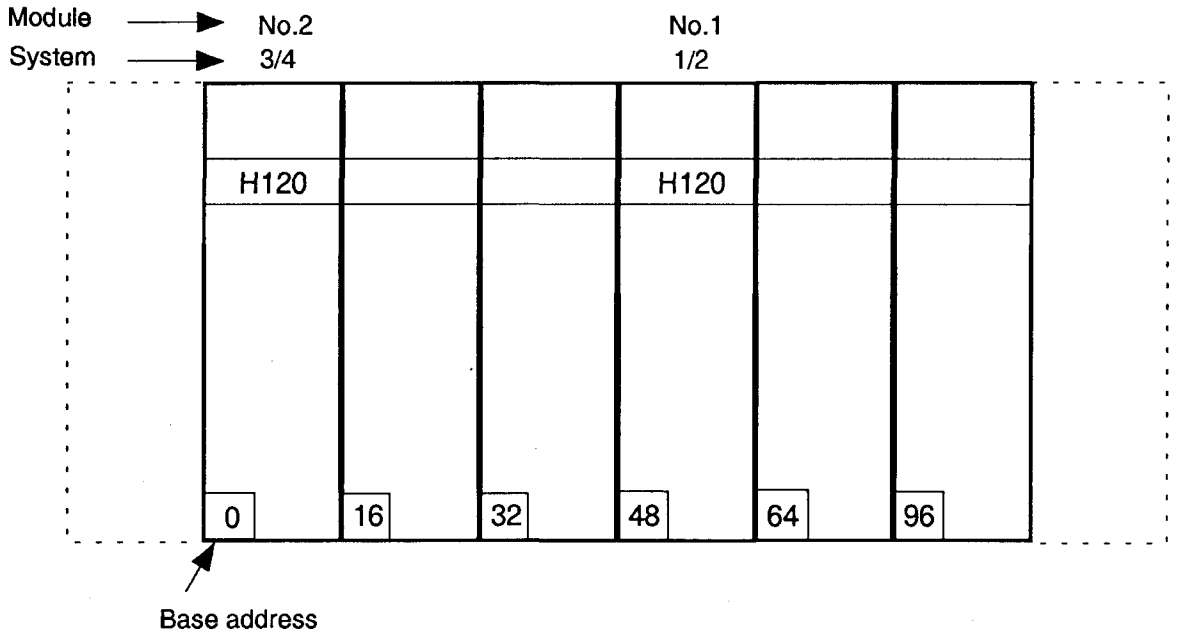
If more than one H120 module is used in the same application, the first module with systems 1 and 2 uses files "xxx_12", the second module uses files "xxx_34", the third module uses files "xxx_56", etc. The commands in the user program will then be, for example, CFB WF_4 for system 4 in module no. 2 or CFB WF_5 for system 5 in module no. 3.

These files for modules 1 to 4 are contained in software package PCD9.H1E1.

Each one of the H120 modules can have any base addresses, both for the hardware and for the PCD flags, registers and FBs. Every module is handled independently, i.e. for each individual module the "H1DEF_xy" files should be provided with unique base addresses, and the corresponding "H1FB_xy" files should be assembled and linked.

2 examples for arranging H120 modules in a PCD4 system





The following summary shows the significance of individual registers and flags relative to defined base addresses.

The symbol names assigned to registers and flags are all pre-defined and can be used in user programs. (See examples)

The present version of the debugger does not allow the use of symbolic names for elements. The table below provides an overall view of the absolute addresses of individual elements.

For example, if the divider value of system 2 is to be viewed with the debugger and the register base address (BR_12) is 500, this produces "DIV_VAL_2" = BR_12+12 = R 512.

	Symbol	Function	Element addresses System 1	System 2
*	CT_VAL_x	Register for counter value	BR_12+1	BR_12+11
*	DIV_VAL_x	Register for divider value	BR_12+2	BR_12+12
*	F_REG_x	Register for output flags	BR_12+3	BR_12+13
*	S_REG_x	Register für status flags	BR_12+4	BR_12+14
	LD_VAL_x	"Transport" register for values to be loaded	BR_12+5	BR_12+15
*	O_FLAG_x	Output Flags to 8 PCD flags	BFLAG_12+24	BFLAG_12+74
*	S_FLAG_x	Status Flags to 8 PCD flags	BFLAG_12+32	BFLAG_12+82
**	R1CO_x	Register 1 output	BFLAG_12+24	BFLAG_12+74
**	R2CO_x	Register 2 output	BFLAG_12+26	BFLAG_12+76
**	CCO_x	Counter output	BFLAG_12+28	BFLAG_12+78
**	OVFLF_x	Overflow flag	BFLAG_12+30	BFLAG_12+80
**	ENAB_x	Enable flag	BFLAG_12+31	BFLAG_12+81
**	INB_x	Input IN-B	BFLAG_12+32	BFLAG_12+82
**	INA_x	Input IN-A	BFLAG_12+33	BFLAG_12+83
**	PRES_x	Preset flag	BFLAG_12+39	BFLAG_12+89
	WF_x	FB Write Function	BFB_12	BFB_12+10
	WD_x	FB Write Divider	BFB_12+1	BFB_12+11
	WCR_x	FB Write Counter/Register	BFB_12+2	BFB_12+12
	RDR_x	FB Read Divider Remainder	BFB_12+3	BFB_12+13
	RCF_x	FB Read Counter and Flags	BFB_12+4	BFB_12+14

_x: signifies System_1 or System_2 (also System_3, _4 etc.)

* These registers and flags can be viewed using the debugger. It is recommended that the CT_VAL is output to a PCA2.D14 display module and the O_FLAGS to outputs during testing.

** The "RCF" (Read Counter and Flags) function block has been structured so that elements designated with ** can be polled directly by the user program using this symbol, e.g. STH CCO_2.

The following sections deal with practical examples from the various fields of application for the PCD4.H120 counting and measuring module. In each case, the structure of the application (counter, generator, interval measurement, frequency measurement) is explained. Various examples are then briefly described and discussed. A proven solution for each example is included as source program ".SRC" on the "PCD9.H1E1" software package diskette.

All programming examples are based on the PCD4 workshop models, V-PCX20, equipped with:

- Supply N...
- Processor M...
- Input module E100, addresses 0 - 15
- Input module E100, addresses 15 - 31
- Output module A400, addresses 32 - 47
- Counting module H120, addresses 48 - 63

In addition, one or two PCA2.D14 display modules and a basic oscilloscope are required, with perhaps a laboratory pulse generator.

Some details on the "PCD9.H1E1" software package

Each "H1FB..." module uses fewer than 400 program lines in the PCD.

The file "H1DEF..." and list "EXTN.TXT" do not take up any memory space in the PCD.

Only a single FB level is used.

Indexing is used only in the routine for the PCA2.D14.

8.2 User programs for counting tasks (*)

A typical user program for a counting task can be separated into three fundamental parts.

- One-time definitions fixed for the whole duration of the application (initialization).
  - Input filter
  - Counting mode
  - Constant output definitions (non-latched)
  - if appropriate, define divider value
  - if appropriate, define preset value
- Definitions for the immediate counting task.
  - Reset/load counter
  - Load registers
  - Define outputs (latched)
  - Activate outputs

From here counting runs automatically on the H120 module, i.e. the module is ready to receive external signals and to process them in the defined manner.

- Evaluation of the counting task within the user program.
  - Read counters and flags
  - Poll flags and outputs (inputs)
  - Display counter value
  - *if appropriate, stop the counter*

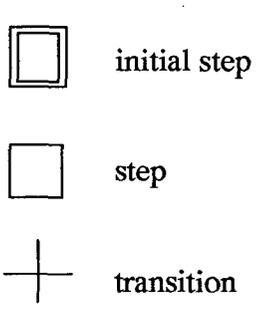
As the previous diagram of typical program structure shows, counting tasks almost exclusively entail a sequential process.

Apart from the one-time definitions executed during controller power-up, the procedure is to define the counting task, then run the counting task to count the externally derived signals over which the user program has no direct influence. After the completion of the counting task, the user program continues. The end of count procedure must always be programmed, even if it only prepares for a new count. The sequence is the same in every case.

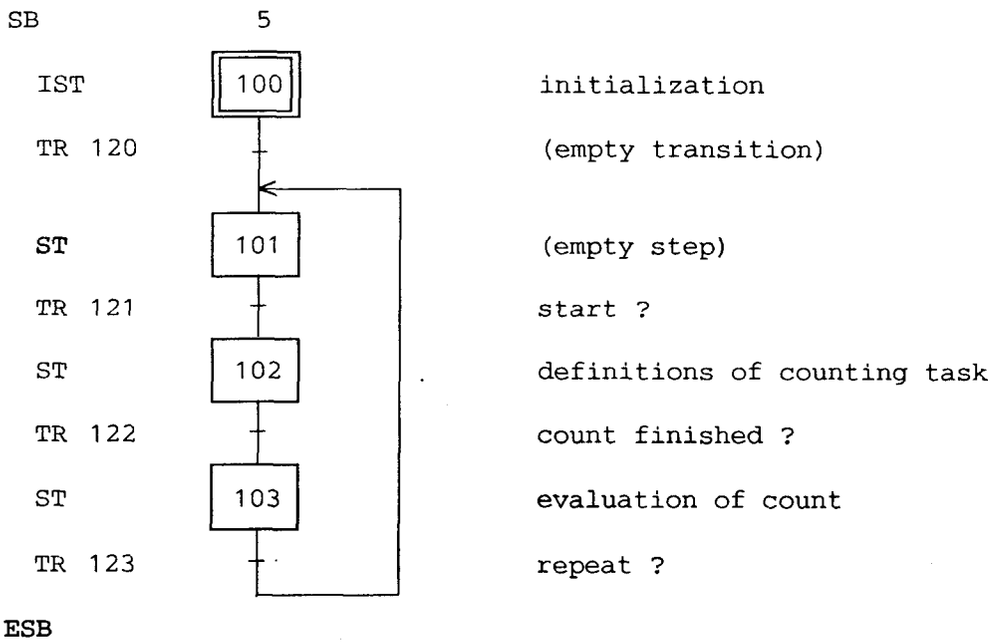
*) All the following examples are capable of running and their titles refer to the relevant chapter number. "EX_825.SRC" indicates that this example (EX = example) is dealt with in section 8.2.5. The source code for all these examples is included on the **"PCD9.H1E1"** software package diskettes.

GRAFTEC produces the best structure for this type of task:

```
COB     0
        0

CSB     5

ECOB
```



For details about GRAFTEC, please consult Chapter 6 of the PCD user manual.



After every false transition, the program returns to the calling block (COB 0) and continues from there. All additional COBs are executed (if present). After returning to COB 0, CSB 5 (call sequential block) causes the program to re-execute all the code in the transition which was false during the previous cycle.

8.2.1 Basic example for very simple "Up/Down" counters (EX_821.SRC)

The purpose is to count the pulses at input IN-A. Maximum counting frequency is 16 kHz. If +24V is present at IN-B, the counter should count upwards. If 0V is present, it should count downwards. Counting system 1 is used.

The counter status and output flags should be viewed using the debugger.

Base addresses in the "H1DEF" file are as follows:

BA_12 = 48, BFLAG_12 = 500, BR_12 = 500, BFB_12 = 800

Counter status can be read from register "CT_VAL_1" (R 501) and output flags read from flags 524 - 531.

Filename for this example: **EX_821.SRC**

```
;---------------------------------
xob             16

cfb             wf_1
                163             ;input filters: 16kHz

cfb             wf_1
                234             ;count mode: single count

exob
;---------------------------------

cob             0
                0

cfb             rcf_1
                0               ;read counter and flags

ecob
;---------------------------------
```

Initialization must at the very least include definitions for the input filter and count mode. If either of these definitions is omitted, the counter will not work.

In this most simple of cases, the user program consists only of reading the counter and flags. The GRAFTEC structure is not used because the program would only contain a single transition.

8.2.2 Simple "Up /Down" counter with display (EX_822.SRC)

To modify the task shown on the preceding page to display the counter status continuously every 0.5 sec. on a PCA2.D14 display, and to transfer the output flags to PCD outputs 32 - 39, incorporate in the COB a call to FB "D14_12" for the display, and a BITO instruction to transfer the output flags from register "F_REG_1" to the outputs.

Filename for this example: **EX_822.SRC**

```
;--------------------------------
xob             16

cfb             wf_1
                163             ;input filters: 16kHz

cfb             wf_1
                234             ;count mode: single count

exob
;--------------------------------

cob             0
                0

cfb             rcf_1
                0               ;read counter and output flags

stl     t       0
ldl     t       0
                5               ;0,5sec

cfb     h       d14_12
                ct_val_1        ;upper display
                ld_val_1        ;lower display
                enable_1        ;system 1

bito            8
                f_reg_1
        o       32              ;outputs 32-39

ecob
;--------------------------------
```

If the display routine is to be called continuously, FB "D14_12" should be called unconditionally. Timer T 0 can then be dispensed with.

8.2.3 Counting with 1 preset (EX_823.SRC)

Same example as "QUICK2" in section 6.

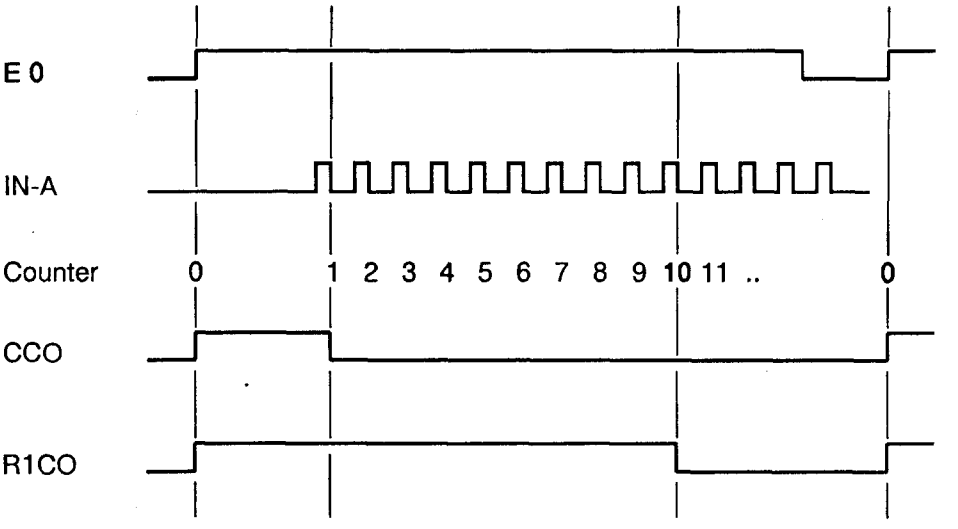
After switching on digital PCD input 0, counting of pulses reaching input IN-A of the H120 module will start, and output "R1CO" will be set = H. Once a preset number of pulses has been counted, output "R1CO" will be reset = L.

The number of pulses to be counted is set by the two BCD switches, which are wired to inputs 16 - 23. (The diagram below shows a setting of 10).

For extra control, the "CCO" (counter output) should be defined such that it is = H whenever the counter is ≤ 0 (less than or equal to zero). The maximum counting frequency should be 100 Hz.

The upper screen of the D14 display module should show counter value (CT_VAL_1) while the lower screen should give the last BCD value loaded (LD_VAL_1). The display should be refreshed every 0.5 sec.

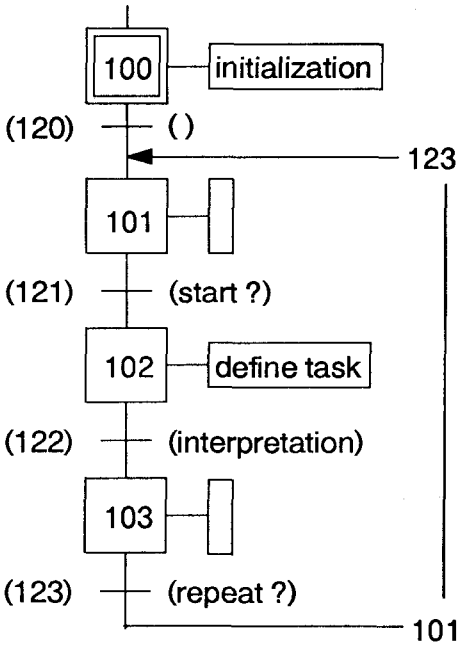
Timing diagram:



Filename for this example : **EX_823.SRC**
(same as QUICK2.SRC)

A detailed description of this program can be found in section 6.2.

Structure and notes on example "EX_823"



EXTN- and EQU list:

The "EXTN" list is taken from the file "EXTN.TXT" (include it in the source module).

SB 5 : Sequential Block
The SB forms GRAFTEC's "shell". Every individual GRAFTEC structure is enclosed in an SB.

IST 100 : Initializes the H120 module. As XOB 16, the IST is only executed the first time the SB is called. It is logical to initialize the H120 module in the IST, so that all the code is contained in the same SB. It is recommended that only those initializations which are global to all PCD tasks should be executed in XOB 16.

TR 120 and ST 101: These are empty and are used only to satisfy the GRAFTEC structure rules. (An empty transition is treated as successful).

TR 121* : This transition can only be exited to the next step if the ACCU is = H at the end of the transition, i.e. if PCD input "I 0" is switched on.

*) See footnote on following page.

ST 102 : This step comprises loading the counter (with zero) and register 1 with the preset value, including definition of the "R1CO" and its activation. A step is only processed once, after the preceding transition becomes true. At the start of each transition and each step, the ACCU is set = H. (Not significant for this example).

TR 122 : This transition is the heart of the program and requires some clarification.

Programmers who are new to GRAFTEC rightly expect a GRAFTEC loop for the continuous processing of function block "Read Counter and Flags", which would continue to run until the preset value had been reached. Full use is made here of the fact that a transition can only be exited for the next step if the ACCU is = H at the end of the transition, and if not then the ENTIRE transition is executed again on the next pass. The function block "Read Counter and Flags" is executed at every pass and the transition's state is determined by polling "R1CO".

ST 103 : This step is empty and is for GRAFTEC structuring only.

TR 123* : Checks that input "I 0" is = L. Step 103 and transition 123 could be left out by using "AND" to link the "I 0" = L condition in TR 122 with the "R1CO" poll, but are included for the sake of clarity.

COB 0 : Every PCD user program must have at least one COB. In the above example, SB 5 is called unconditionally from COB 0. The SB returns after EVERY false transition and EVERY step. Therefore, in our example the program returns to COB 0 after every transition, executes the rest of the COB, then calls the SB again.

Apart from calling the SB, the COB also calls FB "D14_12", which has 3 parameters, to activate the display. If this display routine were called unconditionally, the display would be refreshed on every pass, which would be undesirable for two reasons. First, changes of this speed are too fast for the eye and, secondly, the display routine take about 10ms to execute, which would slow down the user program considerably.

*) To fulfil the task, function block "Read Counter and Flags" only has to be called from TR 122. However, to enable the continuous viewing of counter and flag values, this routine is called wherever the program might linger at a TR.

This example shows how the program can be designed to call the display routine only every 0.5 sec.

If the program "EX_823.PCD" is loaded and running in the PCD, the course of the program can be viewed online from the GRAFTEC editor.

8.2.4 Counting with 2 presets (EX_824.SRC)

For this task, turning on digital PCD input 0 should start the counting of pulses reaching input IN-A of the H120 module.

When the number of pulses at input IN-A reaches that defined in preset 1 (in the diagram = 5), "R1CO" should be set = H. If the second preset value is reached (in the diagram = 9), "R2CO" should be set = H.

If both register outputs (R1CO and R2CO) are = H, output "OK" should be set. As long as input "Start" is = H, counting and display should remain active.

If input "Start" is set = L, counting should stop, and outputs "R1CO" and "R2CO" should be reset. If input "Start" is set = H again, the whole sequence starts again from the beginning.

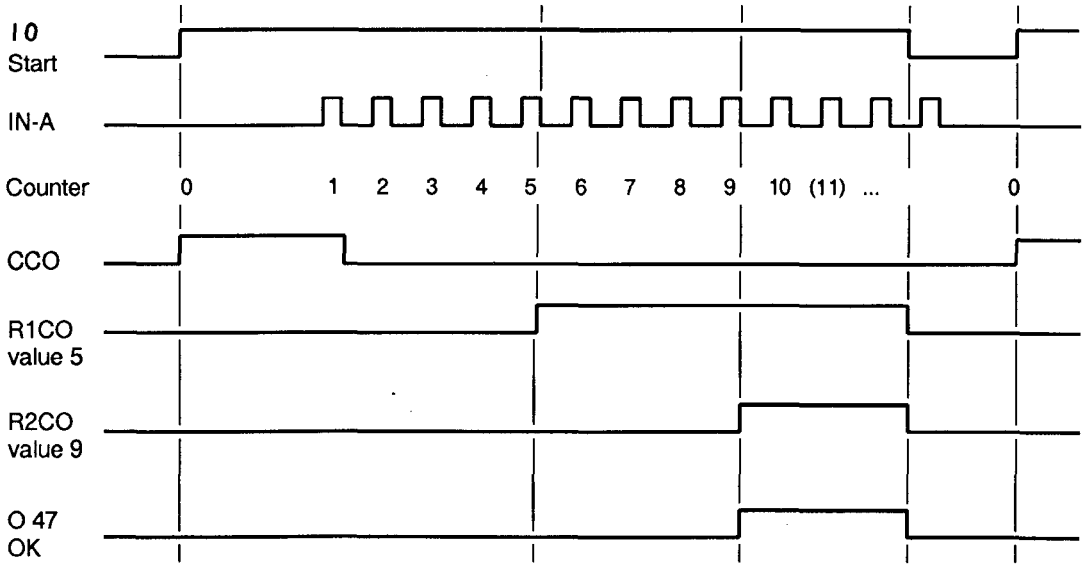
The preset value for register 1 is the BCD value at PCD inputs 16 -23, the preset value for register 2 is the BCD value at PCD inputs 24 - 31.

For checking purposes, the "CCO" (counter output) should be defined such that it is = H when the counter is ≤ 0 (less than or equal to zero).

Maximum counting frequency should be 100 Hz.

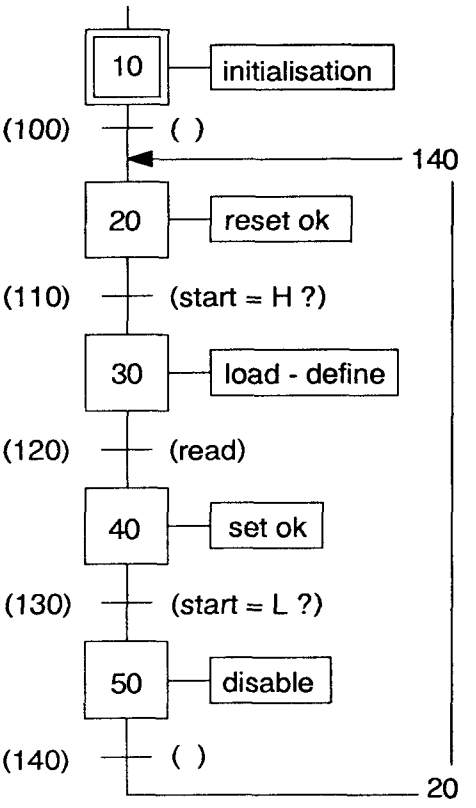
Current counter value should be shown in the upper display, with the lower display showing the last preset value loaded for register 2.

System No. 1 is being used for this task.



IN-B and /Reset : +24V for upwards counting

Structure and notes on example "EX_824"



The SB is called from COB 0 by the user program. Each time the program returns to COB 0 the display and output flags are refreshed every 0.5 sec.

SB 0 : Main Sequential Block

IST 10 : Defines values which are valid for the entire duration of the program: input filter value, behaviour of outputs "CCO", "R1CO" and "R2CO". This only defines the counting behaviour, the enabling is done in ST 30 (Enable Outputs).

Count mode is not defined here in this example, as it will change during the course of the program. Count mode must be defined at a point where it will be taken into account at each pass (ST 30).

TR 100 : Empty, for structural purposes only.

ST 20 : Reset output "OK"

TR 110 : Poll start condition

ST 30 : Defines the counting mode, resets the counter to zero, reads the preset values from the BCD switches and loads them into the registers 1 and 2. Outputs and flags are also enabled. The module is now ready for counting and works independently of the user program, i.e. outputs "CCO", "R1CO" and "R2CO" are set or reset by the counter module itself.

TR 120 : For further operation of the SB, polling clarifies whether both outputs "R1CO" and R2CO" are = H, or whether the start condition is = L. Prior to this linkage, counter status and all flags are polled. Since all false TRs are re-executed in full when the program returns to the SB, up-to-date counter and flag status is always ensured. On exit of a false TR, the program leaves the SB and returns to the point from which the SB was called, i.e. COB 0 in the present example, execting from there to call the display or other routines.

ST 40 : Output "OK" is set.

TR 130 : The start condition is polled. As in TR 120, the counter status and all flags are polled and sent to the display and/ or to the outputs.

ST 50 : Counting is halted and output function disabled.

TR 140 : Empty, for structural purposes only. Returns to ST 20.

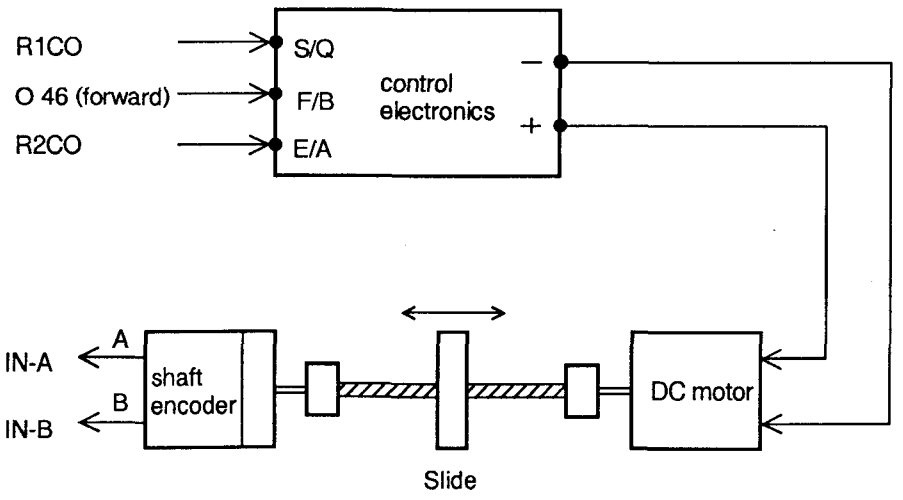
8.2.5 Position control with incremental shaft encoder, 1 axis, positive values (EX_825.SRC)

The task is to move the slider of the motion demo unit (DC motor, spindle, slide, incremental shaft encoder and the appropriate control electronics) from a starting position to another position, then to return to the starting position after a pause. Full acceleration should be used to reach a preset speed, then the velocity slowly reduced until the target position is reached.

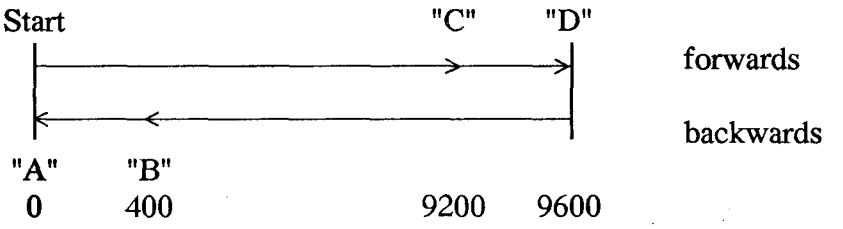
Some technical details of the motion demo unit (V-PCX 10) :

DC motor with gearing	approx. 1200 rpm at 24V=
Incremental shaft encoder	48 pulses/rev., two-phase,
Spindle gradient	1.0 mm
Electronic inputs (V-PCX 11)	forwards/backwards (F/B): H = forwards; L = backwards
	slow/quick (S/Q): L = slow; H = quick
	on/off (E/A) L = motor off, short out H = motor on for forwards/backwards, slow/quick
Electronic outputs	motor (note polarity)
Supply for electronics	24V= smoothed

Wiring for this example :



Motion :



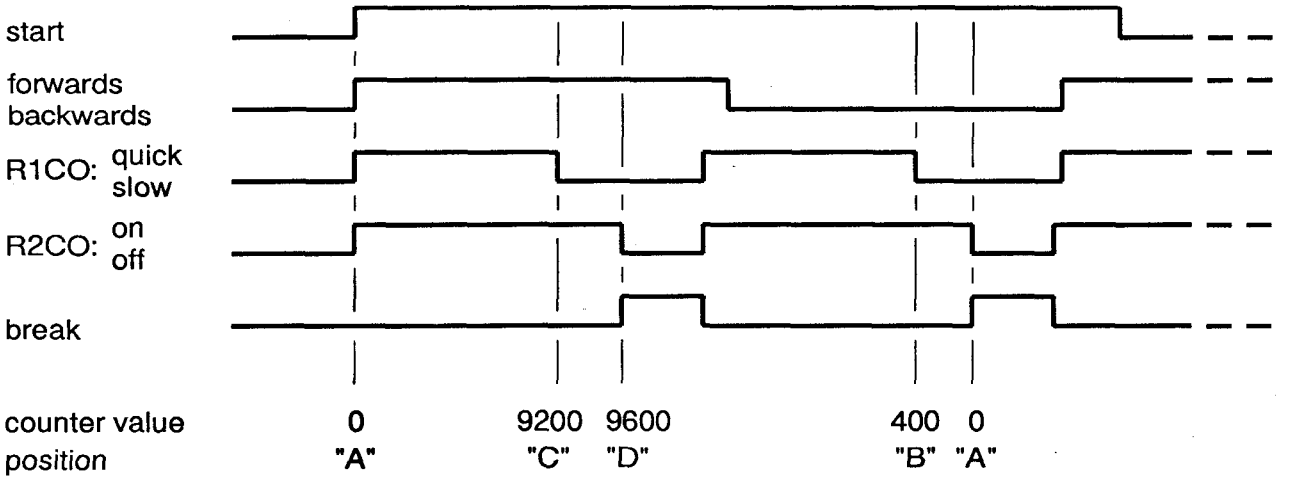
Explanation of structure:

Register 1 is loaded with the changeover value and register 2 with the end stop value. Both register outputs R1CO and R2CO are set as latched H. R1CO switches "quick/slow", R2CO switches the motor "on/off".

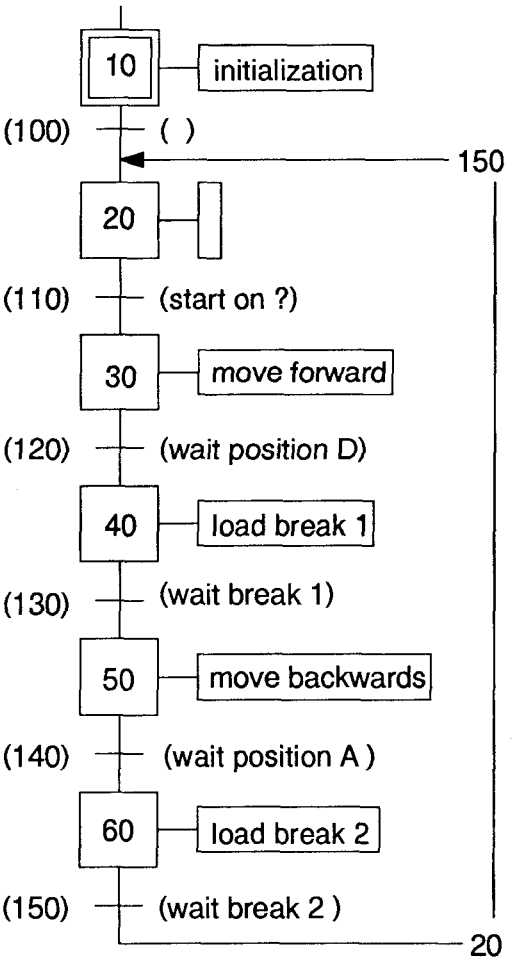
If count = R1, R1CO switches off internally within the module, resulting in "slow" mode. If count = R2, R2CO switches off, stopping the motor. The return journey follows the same pattern.

The counter is set to zero only when the controller is powered up. The counter is therefore a precise representation of slide position, even if there are slight inaccuracies (exceeding the limit position) or if the spindle is turned by hand.

Diagram to show states while running:



Structure and notes on example "EX_825" :



The SB is called in the usual way from COB 0. Each time the program returns to COB 0, the display and output flags are refreshed every 0.5 sec.

SB 0 : Sequential block

IST 10 : Initialization of system 1. Input filters are defined for the pulse sequence of the incremental shaft encoder. The count mode is defined, for our example "x 2" (can be changed to "x 4"). The counter is set to zero only here. In a practical application, it would also be possible to load the previous position from before the installation was last switched off, which had been stored in a non-volatile PCD register.

TR 100 : Empty, for GRAFTEC structural purposes only.

ST 20 : Empty, for GRAFTEC structural purposes only.

TR 110 : Awaits start condition. Each time this TR is false, the counter status and all flags are read, the TR is exits for COB 0 and the display routine is called to display the counter status and output flags.

ST 30 : Both registers are loaded with position values. Register 1 holds the changeover value and register 2 the end stop value.
Since there will always be some overrun of the end position in this configuration, the value can be corrected by the number of pulses, i.e. 9595 instead of 9600. In addition, the function of both register outputs is defined, the control electronics switched to "forwards" and the outputs then enabled. The motor runs and the user program goes to TR 120.

TR 120 : The motor turns, pulses from the shaft encoder clock the the counter. If the changeover value in register 1 is reached, R1CO switches off and the motor is switched to "slow" by the hardware. This is NOT done by the user program. Switching off the motor once the value in register 2 is reached is also done by the hardware.

This TR polls R2CO to determine the transition state for control of the SB. Prior to this, the counter and output flags are read in the usual way.

ST 40 : Break 1 is loaded here.

TR 130 : Here the program awaits stop position 1, while reading the counter and output flags, so that any overrun of the destination position can be seen from the display.

ST 50 : As ST 30, with other register values, backwards.

TR 140 : As TR 120.

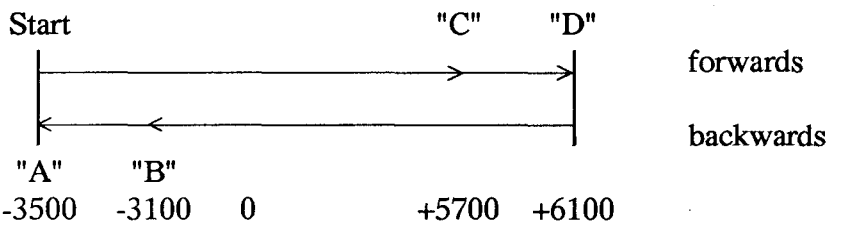
ST 60 : As ST 40, but loads break 2.

TR 150 : As TR 130. Then return to ST 20 for another pass.

8.2.6 Position control with incremental shaft encoder, 1 axis, positive and negative values (EX_826.SRC)

Task and program structure are the same as for the preceding example 8.2.5, except the axis should have its zero point approximately in the middle, so that slider motion occurs both in the negative and positive counting ranges.

Motion :



If the slide is at position "A" during power-up, the counter should be loaded with -3500 when it is switched on.

Negative values are written to the counter and registers with the command "Write to Counter, Negative" or "Write to Register 1 or 2, Negative". Only UNSIGNED (positive) values are loaded.

The display of the counter contents via the debugger or on the D14 display module always shows absolute positive values, i.e. without a minus sign. Output flag "C < 0" or the correspondingly defined output "CCO" shows whether the value displayed is positive or negative.

To display a SIGNED count value with the debugger, the user program can negate the count (2's complement).

User program "EX_826.SRC" provides for +/- display when viewing the counter with the debugger. Counter flag "C < 0" (see section 7.4) is polled from COB 0, after the program has returned from SB 0. If the counter value is negative, PB 10 is called, if it is positive, PB 11 is called. In PB 11 the value is converted for negative display. The displayable value is in PCD register R 1000.

A special routine is required for +/- display on the D14, which will not be described here.

The source program for this example can be found under filename "EX_826.SRC" in software package "PCD9.H1E1".

8.2.7 Position control with multiple axes (multiple systems) (EX_827.SRC)

Based on the preceding examples in sections 8.2.5 and 8.2.6, the task is to create a user program for 4 similar axes with two PCD4.H120 modules. Each individual axis should be driven independently of the others. Counter values and the last loaded register values are to be output to 4 display modules. Output flags will not be shown.

System configuration :

Input module	PCD4.E100 addresses 0 - 15
Comb. I/O module	PCD4.B900 addresses 16 - 31
Counter module 2	PCD4.H120 addresses 32 - 47 (systems 3 and 4)
Counter module 1	PCD4.H120 addresses 48 - 63 (systems 1 and 2)

Various program structures are possible:

a) Derived from the preceding example "EX_825", an SB is created to correspond to each axis and each of the 4 SBs is called from its own COB.

b) Four SBs are created in the same way, but they are all called from the same COB.

c) All 4 axes are programmed in a single SB. This method should normally only be used when the axes are dependent on each other. Apart from the ease of synchronization, it also has the advantage that the state of all the axes can be viewed together on the same screen, e.g. using the Graftec editor on-line.

For our example "EX_827" we will use (b).

The EXTN declarations for modules 1 and 2, i.e. for systems 1, 2, 3 and 4 must be copied in from the "EXTN.TXT" file.

All commands for module 1 (systems 1 and 2) are of the type "_12", while those for module 2 (systems 3 and 4) are of the type "_34".

All commands which refer to the individual systems are of the type "_1", "_2", "_3" or "_4".

Base addresses for the different modules must be correctly defined in files "H1DEF_12" and "H1DEF_34". Files "H1FB_12" and H1FB_34" should be left unchanged.

After assembling the user program, the four "H1..." files and any others, they are linked together.

e.g.: EX_827.OBJ + H1DEF_12.OBJ + H1DEF_34.OBJ
+ H1FB_12.OBJ + H1FB_34.OBJ

The loadable file EX_827.PCD is produced.

The source program for this example is in file "EX_827.SRC" in software package "PCD9.H1E1".

The header from the "MAP" file shows the linked modules and their sizes:

```
*** SAIA PCD LINKER V1.5 ***    FILE: EX_827.PCD
<<< PRE-RELEASE VERSIONS FOR SAIA INTERNAL USE ONLY >>>

                                    CODE START  CODE SIZE   TEXT SIZE
SOURCE FILE    ASSEMBLY  DATE       LINE        IN LINES    IN BYTES
-------------------------------------------------------------------

EX_827.SRC     14.06.91  14.10      000000      000492      000000
H1DEF_12.SRC   11.12.90  11.16      000492      000000      000000
H1DEF_34.SRC   14.06.91  14.11      000492      000000      000000
H1FB_12.SRC     5.02.91  16.45      000492      000352      000000
H1FB_34.SRC    14.06.91  14.12      000844      000352      000000

-------------------------------------------------------------------


          TOTAL CODE SIZE:        001196 LINES

          TOTAL TEXT SIZE:        000000 BYTES

          GLOBAL SYMBOLS:         112
```

8.2.8 Count scaling (use of divider) (EX_828.SRC)

Using as a starting point the example in section 8.2.5 "Position control with incremental shaft encoder, 1 axis, positive values", this example demonstrates the scaling features.

The motion demo unit (V-PCX10) has an incremental shaft encoder which yields 48 pulses per revolution. Operation is in "x 2" mode, so that 96 signals per revolution are output. The spindle gradient is 1 mm, resulting in 96 pulses for 1 mm.

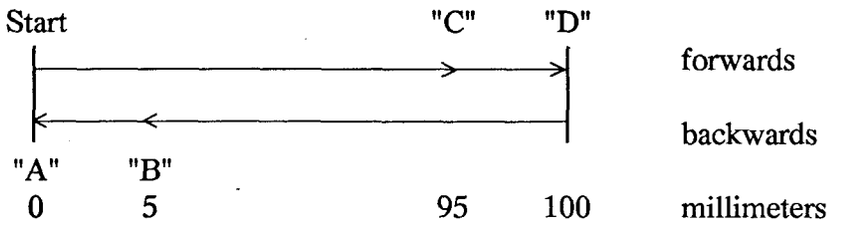
The aim of this examples (EX_828) is to divide by 96 the signals arriving at the counter module, in order to carry out position control or counting directly in millimetres.

The divider as described in section 7.3.5 is used here. During the initialization, the divider is loaded with a value of 95, and using the command "Divide IN-A and IN-B" it is instructed to direct signals from both count module inputs via the divider.

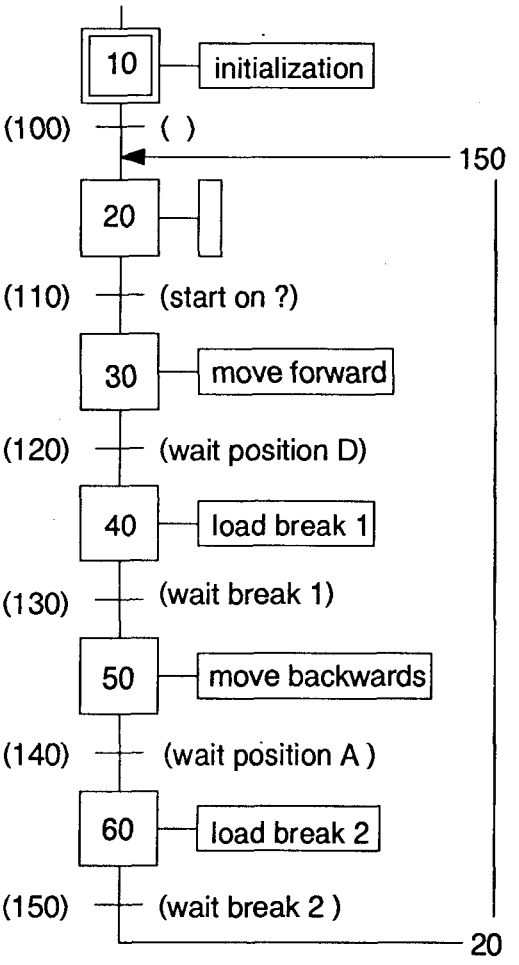
During upward counting, the first 95 pulses go into the divider remainder register. The 96th pulse then goes to the counter and the remainder becomes 0. The next 95 pulses again go into the remainder register, and so on. In this way only every 96th signal reaches the counter. Remainder values are stored in the divider remainder register, where they can also be read (Read Divider Remainder).

This scaling is realised in example "EX_828.SRC". All values are given in millimetres. The D14 display module shows the counter value in the upper display, and the divider remainder in the lower. By slowly turning the spindle manually, it is possible to view counting with the divider. Using the debugger, the counter value can be seen in base register + 1 and remainder value in base register + 2.

Motion :



Structure and notes on example "EX_828" :



The SB is called in the usual way from COB 0. Each time the program returns to COB 0 the display is refreshed.

SB 0 : Sequential block

IST 10 : Initialization of system 1. Defines the input filters, count mode and CCO output, and loads a divider of 95. The counter is also reset to 0. For effective counting and CCO output reaction even before motion starts, the divider and CCO outputs are enabled with "Divide IN-A and IN-B; Enable Outputs".

TR 100 : Empty, for GRAFTEC structural purposes only.

ST 20 : Empty, for GRAFTEC structural purposes only.

TR 110 : Await start condition. Each time this TR is false, the counter status, all flags and the divider remainder are read. The TR is then exited for COB 0 and the display routine is called to display the counter status, divider remainder and output flags.

ST 30 : Both registers are loaded with position values in mm. Register 1 holds the changeover position and register 2 the end stop position. In addition, the functions of both register outputs are defined and the control electronics switched to "forwards", after which the outputs and divider are enabled with "Divide IN-A and IN-B; Enable Outputs". The motor runs and execution moves to TR 120.

TR 120 : The motor turns, pulses from the shaft encoder clock the counter. If the changeover value in register 1 is reached, R1CO switches off and the motor is switched to "slow" by the hardware. This is NOT done by the user program. Switching off the motor once the value in register 2 is reached is also done by the hardware.

This TR polls R2CO to determine the transition state for control of the SB. Prior to this, the counter, divider remainder and output flags are read in the usual way.

ST 40 : Break 1 is loaded here.

TR 130 : Stop position 1 is awaited here, while the counter, divider remainder and output flags are read.

ST 50 : As ST 30, with other register values, backwards.

TR 140 : As TR 120

ST 60 : As ST 40, but break 2.

TR 150 : As TR 130, then returns to ST 20 for a new pass.

8.2.9 The use of counting for measurement (EX_829.SRC)

Summary of task:

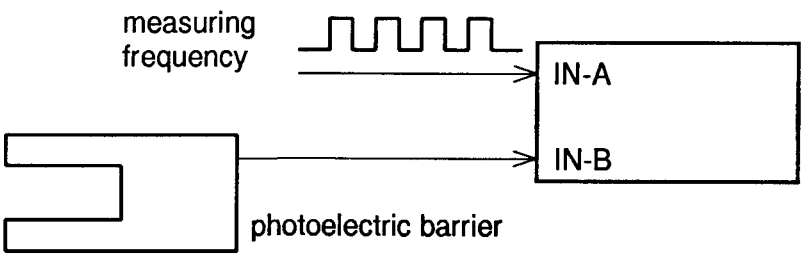
Whenever a photoelectric beam is broken, pulses with a frequency of 10 kHz (for example) are counted as they reach the counter. When the photoelectric beam is restored, counting should stop, the counter value transferred to a PCD register and the counter reset to 0. If the photoelectric beam is again broken, the 10 kHz pulses are counted again in the same way, when the beam is restored, the count value is stored in the next consecutive PCD register, and so on.

This kind of device is often used in practice, for example, to measure items passing on a conveyer belt or lengths of paper or material running through a photoelectric barrier. It entails the generation of count pulse frequency by the device itself, in proportion to belt velocity and feed rate of the lengths being measured.

It is therefore a question of counting measurement signals for the duration of a particular situation, e.g. while the photoelectric cell is covered, during which counting is controlled directly by the photoelectric barrier, i.e. by an input on the counting module, and NOT via a digital PCD input.

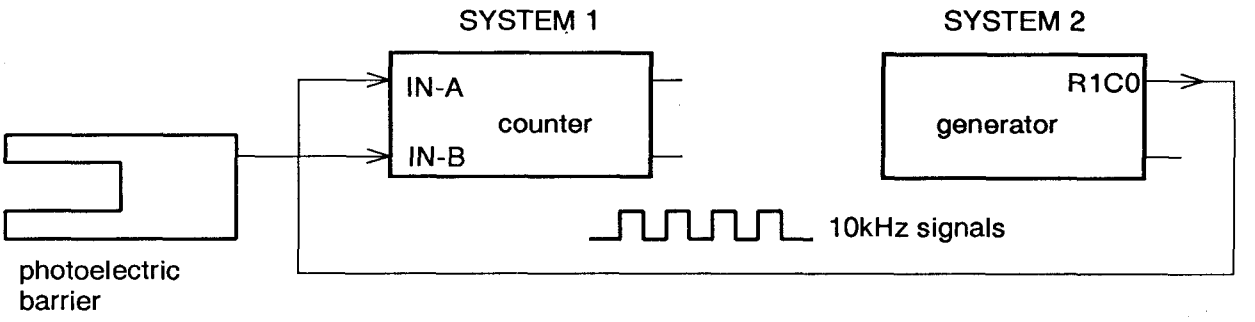
Count mode command no. 254 "Count enabled if IN-B = L" provides an elegant solution to this task, as counting is only enabled if IN-B = L.

If our photoelectric barrier is connected to input "IN-B" and the 10 kHz signals to input "IN-A", the task is already solved as far as counting is concerned.

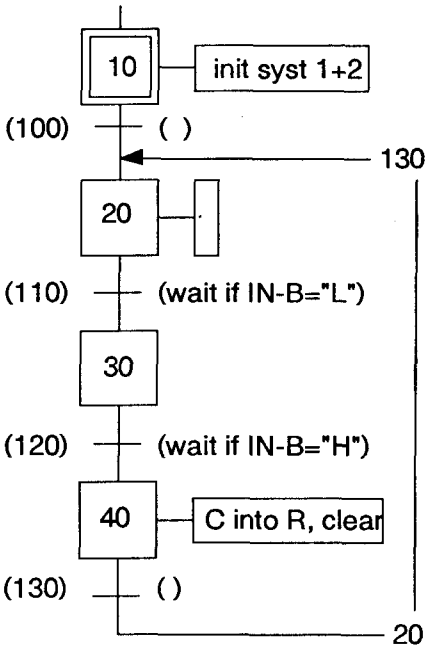


This kind of arrangement forms the basis of example "EX_829.SRC", in which system 2 of the counting module is set up as a 10 kHz generator, so that the example can be demonstrated without needing additional devices. (The generator is discussed later in section 8.3)

Instead of a photoelectric barrier, a debounced switch will be used.



Structure and notes on example "EX_829" :



The SB is called from COB 0. If PCD input I 0 is switched on, the current counter value and the last count are displayed on the D14 display module. PCD input I 1 deletes registers 1000 - 1100, and the results are stored again from register 1000 onward.

SB 0 : Main Sequential block

IST 10 : Initializes the application:

Generator for 10 kHz on system 2. (More details on this can be found in section 8.3). The generator is defined and started up. The generator runs independently on the counting module, it will continue to run even if the CPU is stopped.

Input filters, output CCO, count mode: "Enabled Count if IN-B = L" and CCO output enabling are defined and executed for the counting task (system 1).

TR 100 : Empty, for structural purposes only.

ST 20 : Empty, for structural purposes only.

TR 110 : The counter contents and all counter and status flags are polled and refreshed. Input IN-B of system 1 (photoelectric barrier) is then checked for "L". This polling only serves to keep the rest of the user program running.

Counting itself is controlled (started) within the module itself, according to the definition "Enabled Count if IN-B = L" and is not influenced at all by the user program.

ST 30 : Empty, for structural purposes only.

TR 120 : The counter contents and all counter and status flags are polled and refreshed. Input IN-B of system 1 (photo-electric barrier) is then checked for "H". This polling only serves to keep the rest of the user program running.

Counting itself is controlled (stopped) within the module itself, according to the definition "Enabled Count if IN-B = L" and is not influenced at all by the user program.

ST 40 : After counting has definitely ended, the counter contents are read and stored in the next PCD register, starting from R 1000. Afterwards the counter is reset to 0. After the empty transition TR 130 and empty step ST 20 have been processed, the program goes back to TR 130, which awaits the start of the next count.

TR 130 : Empty, for structural purposes only.

8.2.10 Example for general use of divider and divider remainder

This example is primarily intended to demonstrate principles for handling the divider remainder, particularly the command sequence for writing and resetting the divider remainder.

Section 7.6 Command sequence, point 5., states that:

> "Write To Divider Remainder" must be followed by "Write To Divider".

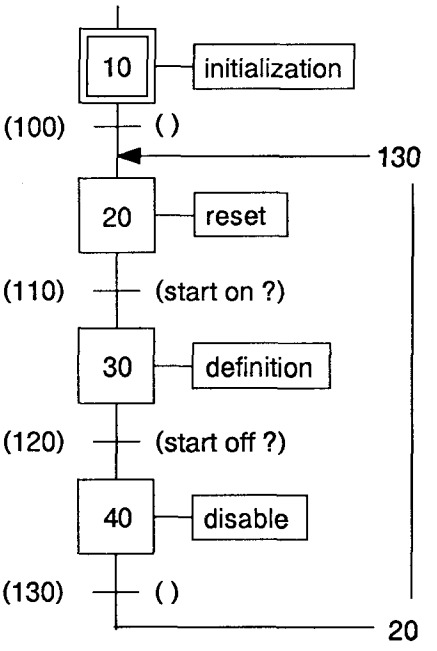
If any application requires the divider remainder to be set to a new value (usually zero), both the divider remainder and the divider must be reloaded.

The following example "EX_8210" shows the command sequence in a working program.

Task:	After turning on the PCD "Start" input, the counter and the divider remainder should be reset. Counting is enabled and incoming signals are counted. After counting, if the PCD "Start" input is switched off, counting should be disabled and both the counter and divider remainder should be reset to zero.
Count mode: *)	Count Mode: Dual Count (+/-) Signals at input "A" count "down", Signals at input "B" count "up".
Divider:	Every tenth signal should be transmitted to the counter. —> Divider value = 9.
Display:	Using a PCA2.D14 display module, the upper display should show the counter status and the lower display should show the divider remainder.
Neg. values:	If the counter contains a negative value, output "CCO" should be enabled.

*) Any count mode desired can be selected.

Example "EX_8210" program structure



The SB containing the actual user program for the H120 module is called from COB 0 in the usual way. The display is also continuously refreshed each time the program returns to COB 0.

SB 0 : Sequential Block

IST 10 : Initialization of system 1. Input filters and CCO output function are defined.

TR 100 : Empty, for GRAFTEC structural purposes only.

ST 20 : Reset counter,
Reset divider remainder,
Load divider.

Important: "Write To Divider Remainder" must be followed by "Write To Divider".

TR 110 : Await start condition.
Before the actual start condition "STH Start", both the counter with output flags and the divider remainder are read (refreshed).

ST 30 : Count mode is defined and outputs enabled.

TR 120 : As TR 110. Termination of counting is awaited.

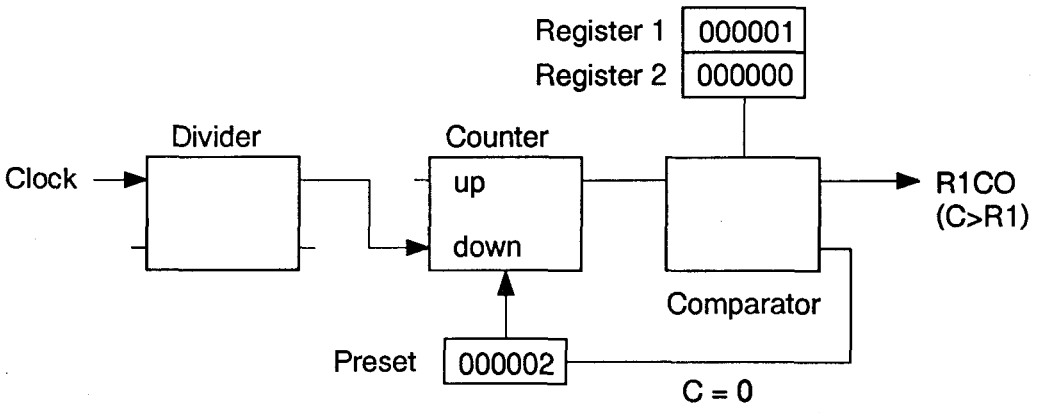
ST 40 : Counting is disabled.

TR 130 : Empty, for GRAFTEC structural purposes only.

8.3 User programs for pulse outputs

There are no commands to directly select an output frequency and start a pulse generator. However, by skilful combination of various commands and counting module elements, it is possible to produce several types of pulse generator.

It is ultimately a question of generating a chain of pulses at the module output, e.g. at R1CO, which is accessible to the user, and whose frequency is defined by the user program.



The description of this circuit is given later.

8.3.1 Generating pulse chains, pulse generators

Example standard generator:

Create a generator to produce a symmetrical square wave signal with a frequency of 1000Hz using output R1CO.

The following commands should be written in sequence:

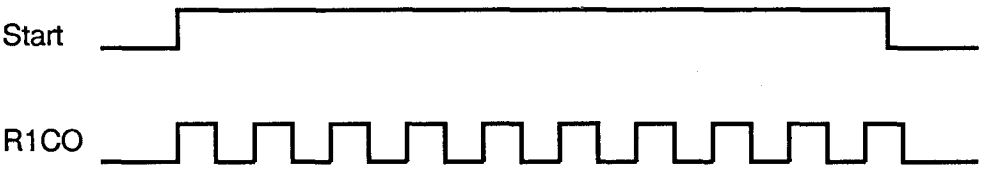
```
Command No. 140  : Output R1CO: C > R1
Command No.  40  : Write to Register 1, pos. value 1
Command No.  36  : Write to Preset, pos. value 2
Command No. 219  : S.C.D. is Down; Preset if C = 0

Command No.  64  : Write to Divider, value 499
Command No. 188  : Divider IN-A: Continuous 1 MHz
Command No. 106  : Divide IN-A; Enable Outputs
```

Once this command sequence has been processed, symmetrical square wave signals with a frequency of 1000Hz can be measured at output R1CO. The counting module is now working as a generator in an absolutely autonomous fashion. Even if the CPU is stopped or completely removed, as long as supply is present at the module, square-wave signals with a frequency of 1000 Hz will be produced at output R1CO.

An explanation is first given of how to handle this generator, together with an example. Generator function is described at the end of the section.

Example "EX_831":

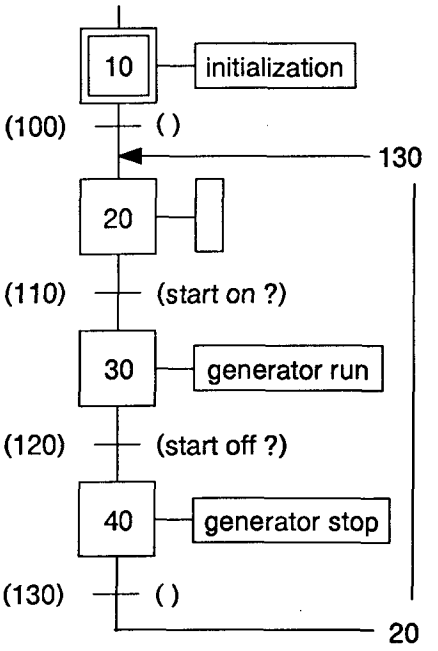
For as long as PCD4 input "Start" is switched on, symmetrical pulses with a frequency of 1000 Hz should be produced at output R1CO of counting module system 2.



The command sequence shown at the beginning of this section is placed in the user program. Once this command sequence is executed, the counting module works independantly as a pulse generator, with the values given. This was already done in example "EX_829".

For this example, we will divide the command sequence into an initialization section and a parameter section.

Structure of example "EX_831":



The SB containing the actual user program for the H120 module is called from COB 0 in the usual way.

SB 0 : Sequential Block

IST 10 : Initialization of system 2 as a generator. The precise function of this part of the program is described later.

TR 100 : Empty, for structural purposes only.

ST 20 : Empty, for structural purposes only.

TR 110 : Await start condition.

ST 30 : After generator function has been defined in IST 10, divider value is entered here which, together with base frequency, determines the output frequency. Base frequency is then defined and the generator is finally started with "Enable Outputs".
From this point onward, the generator runs independently, i.e. without receiving any further commands from the user program.

TR 120 : Await stop condition.

ST 40 : Generator stop. "Disable Outputs" on its own will also stop the generator, however a subsequent restart cannot be easily done.

TR 130 : Empty, for structural purposes only.

For testing, the divider value at address 32 and/or the base frequency at address 37 can be changed using the debugger. The new values take effect after switching PCD input I 0 off and on again, and the output frequency adjusted correspondingly.

It is useful to connect an oscilloscope at output R1CO to examine the output signals. Since the module's output is "Open Collector", a load resistor for 0V must be incorporated. For test purposes, connection of the output to a PCD input (2,2k Ohms) is sufficient. For higher frequencies (>20kHz) an increased load should be selected to achieve steeper edges. (max. 500mA —> 48 Ohms at 24V).

Any generator output should always be run with +24V.
(Supply terminal "+")

Function of the standard generator
(c.f. circuit at beginning of this section)

Signals from the internal clock generator are sent via the divider to the counter, which is then decremented. Register 1 has been loaded with the value 1 and the preset register with the value 2. The preset is activated when C is equal to 0. Output R1CO is defined to be high when C is greater than R1. A value 0...9999 is written to the divider.

The command sequence is shown again here:

Command No. 140 : Output R1CO: C > R1
Command No. 40 : Write to Register 1, pos. value 1
Command No. 36 : Write to Preset, pos. value 2
Command No. 219 : S.C.D. is Down; Preset if C = 0

Command No. 64 : Write to Divider, value 499
Command No. 188 : Divider IN-A: Continuous 1 MHz
Command No. 106 : Divide IN-A; Enable Outputs

When the controller is powered up, the counter is at zero and the preset value of 2 is thus loaded into the counter. R1CO will be high, since C is greater than R1 (C = 2, R = 1).

Signals are continually going from the internal clock generator to the divider and then to the counter, which is decremented. (S.C.D. is Down). After the first decrementing signal, the counter = 1 and R1CO goes low, since the condition C > R1 is no longer fulfilled. After the next decrementing signal, the counter = 0 and thus meets the preset condition again; the counter is once more loaded with 2 and output R1CO thus becomes high again, and so forth.

This circuit is a free-running system which, once started, runs independently and supplies symmetrical output signals without further intervention and without any influence from the CPU or the user program.

The generator is stopped essentially by disabling output R1CO with "Disable Outputs" and/or by stopping the counter with "Disable Count".

Frequency is changed by adjusting the internal clock generator frequency for wider or narrower steps by adjusting the divider value. These changes can be made while the generator is running. The user program should be designed accordingly, as illustrated by the following examples.

Important: The maximum count frequency of the counter is 166kHz. This limit should be taken into account when selecting the base frequency and divider value, otherwise the calculated and actual frequencies will not agree.

8.3.2 Output of a selectable number of pulses with selectable frequency

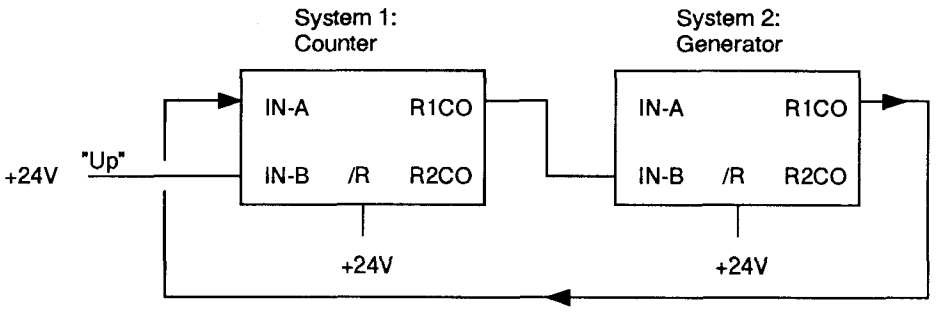
Produce a generator whose frequency can be varied using two BCD switches (on inputs 16 - 23), between 50 Hz (BCD input = 99) and 5000 Hz (BCD input = 00). The number of pulses output can be set by two BCD switches (on inputs 24 - 31), multiplied by a factor of 100.

Pulse output is started by the PCD "Start" input.

The upper display of a PCA2.D14 display module should continuously show the current counter status of the number of output pulses while the lower display shows the preset value for the number of pulses to be output.

Pulses should be generated by system 2 and counted by system 1.

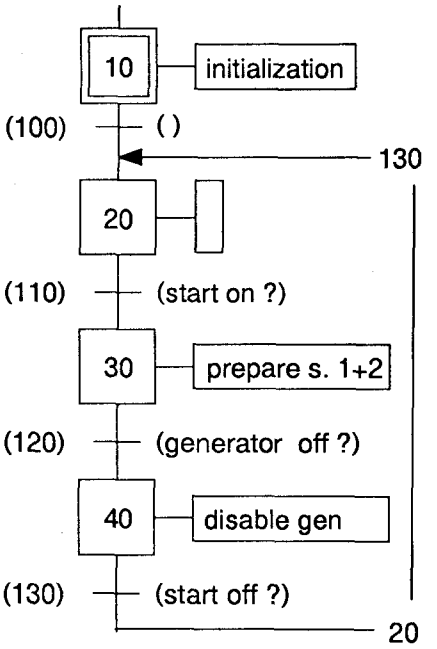
Structure:



Command 250 "Count Mode: Enabled Count if IN-B = H" ensures that the generator only outputs pulses if IN-B on the generator is high. Since IN-B_2 is connected with R1CO_1, care must be taken that R1CO_1 remains high until the number of pulses indicated has been reached.

The generator uses the principle of the preceding example "EX_831": Generating pulse chains, while the counter uses that of the example "EX_823": Counting with 1 preset.

Structure of example "EX_832":



The SB containing the actual user program for the H120 module is called from COB 0 in the usual way. In addition, each time the program returns to COB 0 the display and flags are refreshed.

SB 0 : (Sequential Block)

IST 10 : Initialization of both systems according to the examples given.

TR 100 : Empty, for structural purposes only.

ST 20 : Set counter (system 1) to zero.

TR 110 : Await start condition. Whenever the TR is not fulfilled, counter status and flags are read and, after leaving the TR, the display routine is called in COB 0 and counter status and output flags are displayed.

ST 30 : When the start condition is fulfilled, both counter (system 1) and generator (system 2) are loaded with the appropriate values and conditions in this ST and then enabled. If this ST is processed, pulses are output at R1CO_2 and sent to the counter via the connection with IN-A_1. When the given number of pulses has been reached, the hardware connection R1CO_1 - IN-B_2 blocks the generator. After processing, this ST is left for the next TR and the counting module then functions completely autonomously.

TR 120 : For continued running of the SB, the program waits here until all pulses are output. The switching condition is R1CO_1 = L. Whenever the TR is not fulfilled, counter status and flags are read and, after leaving the TR, the display routine is called in COB 0 and counter status and output flags are displayed.

ST 40 : In ST 30 the generator was only blocked, not disabled. To guarantee a correct restart in the next pass, the generator is switched off correctly in this ST.

TR 130 : Wait until start condition is removed. Then return to ST 20 and zero the counter.

8.3.3 Generation of ramps

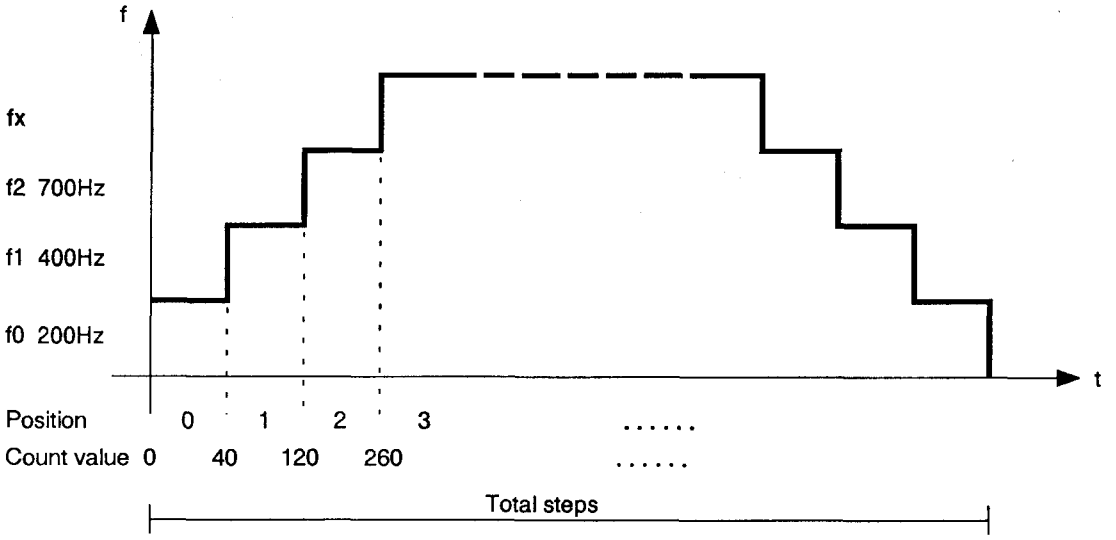
Changing frequency during pulse output is permissible (*). If a symmetrical square pulse generator is present, according to the two preceding sections, frequency can be changed by simply altering the divider value. The frequency is only changed after the current period has completed. This is the basis for the generation of proper graduated ramps, e.g. for accelerating and braking stepper motors.

However, for the control of stepper motors with continuous linear ramps, the PCD4.H200 module offers the best solution.

Manufacturers of stepper motors advise that a graduated ramp consisting of at least 10 positions gives almost the same result as a continuously rising or falling ramp.

Principle of a graduated frequency ramp:

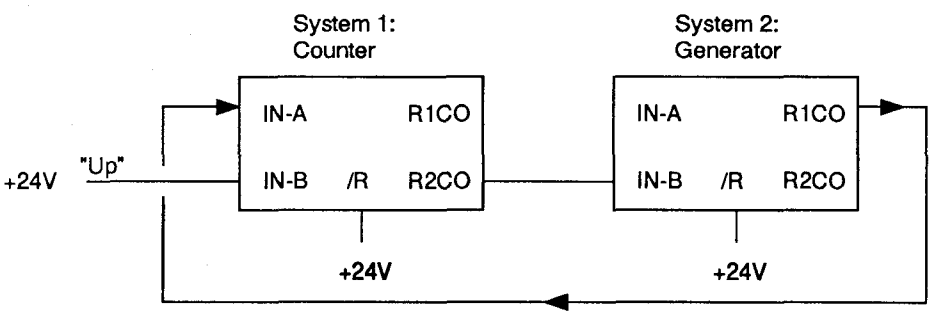
Simple example of a time linear ramp



*) from index c), for Index a) and b) with max. frequency of 5kHz

PCD4.H120 arrangement for realization of a ramp generator.

Both systems of a module are used, with one system working as the generator and the other as the counter.



A frequency change takes place at fixed intervals of, for example, 200ms. The ramp begins at position 0. Count value is 0. Once the generator has been started, pulses with a frequency of 200Hz are output for a period of 200ms. This results in 40 pulses with a frequency of f0 for position 0.

After the first 40 pulses have been output and counted, frequency is immediately raised to f1 = 400Hz. For the next 200ms, pulses with a frequency of f1 are output, i.e. 80 pulses. The counter counts up to 40 + 80 = 120.

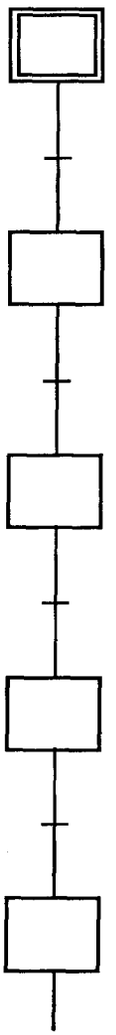
Frequency is then raised to f2, e.g. 700Hz. Over the next 200ms 140 pulses are output. the counter counts up to 120 + 140 = 260.

And so forth...

Each position has its own frequency and count value. The entire pulse train can be summarized in a table:
(Duration of each position = 200ms)

Start		Count value : 0
Pos. 0 :	f = 200Hz	Count value : 0 + 40 = 40
Pos. 1 :	f = 400Hz	Count value : 40 + 80 = 120
Pos. 2 :	f = 700Hz	Count value : 120 + 140 = 260
Pos. 3 :	f = 1000Hz	Count value : 260 + 200 = 460
Pos. 4 :	f = 1200Hz	

If the values for a complete pulse train are known, calculated in the PCD or arrived at empirically, a user program is created according to the following principle:



Initialize both systems

Start ?

Select frequency f0
Enter count value for pos. 0 in R
Start generator

Count value for pos. 0 reached ?

Select frequency f1
Enter count value for pos. 1 in R

Count value for pos. 1 reached ?

Select frequency f2
Enter count value for pos. 2 in R

Count value for pos. 2 reached ?

etc.

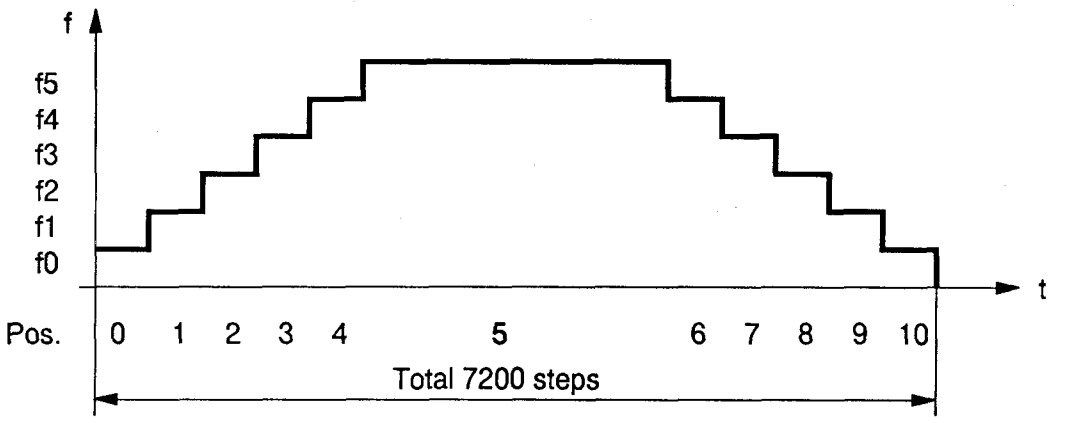
The principle of time controlled positions is also conceivable. However, this is less precise but uses only one system in the counting module. It would thus be possible to control 2 axes with one module.

Example EX_833

A stepper motor should accelerate in 5 positions and, at the end, decelerate in 5 positions. Acceleration and deceleration ramps should be symmetrical. 7200 steps should be taken forwards and, after a pause of 2 sec, 7200 steps should be taken backwards again.

The example can be realized using the V-PCX 10/11 workshop model.

To avoid overloading the user program, values for the individual positions are given and not calculated by the user program.



So that the ramp can be easily viewed using an oscilloscope or the model, duration of the individual positions has been set at 500 ms. In practice, this value would be substantially shorter.

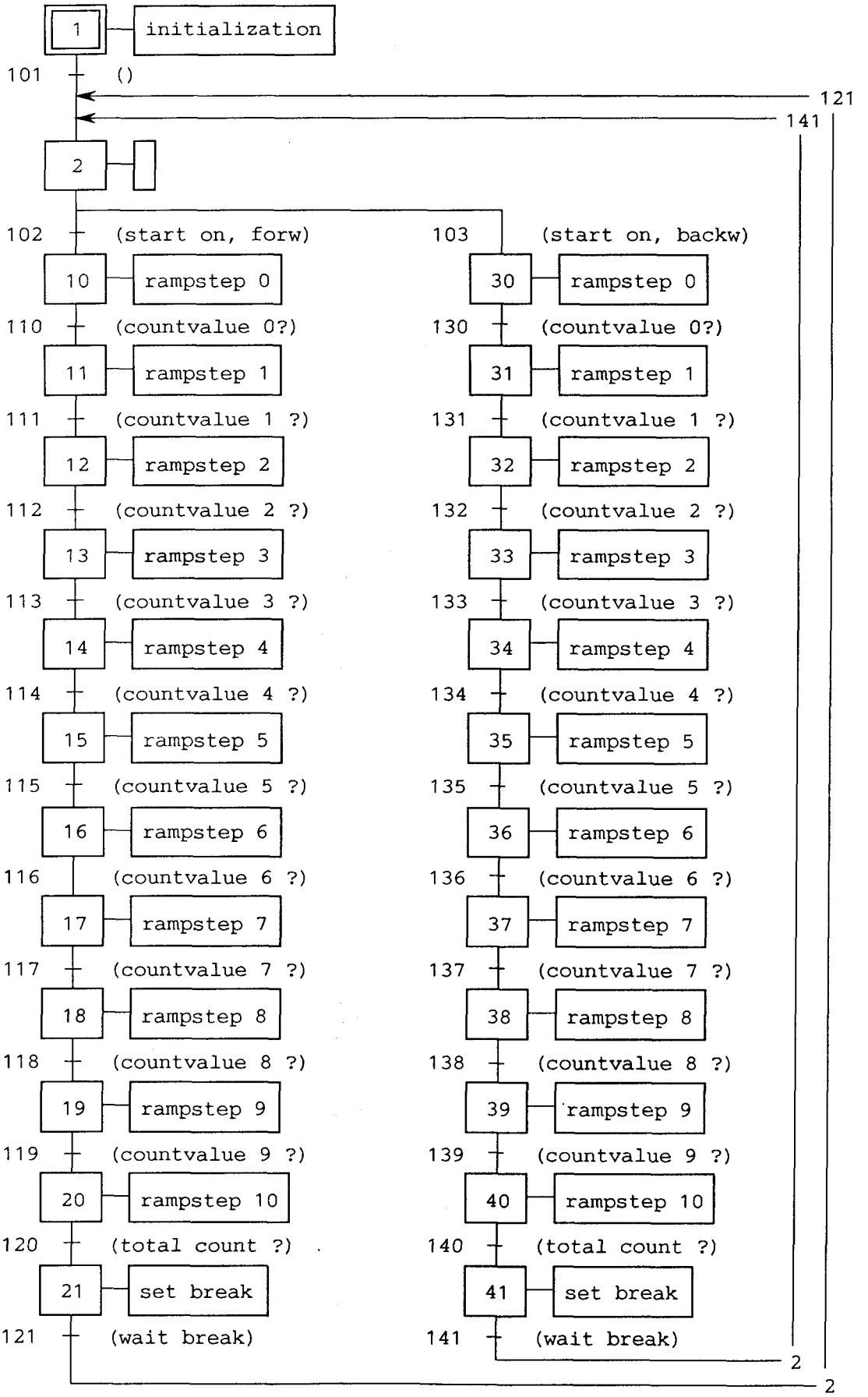
Pos.	Frequency f [Hz]	Divider Value DV	Count Value CV
Start			0
0	100	4999	0 + 50 = 50
1	200	2499	50 + 100 = 150
2	400	1249	150 + 200 = 350
3	800	624	350 + 400 = 750
4	1200	416	750 + 600 = 1350
5	1500	332	7200 - 1350 = 5850
6	1200	416	7200 - 750 = 6450
7	800	624	7200 - 350 = 6850
8	400	1249	7200 - 150 = 7050
9	200	2499	7200 - 50 = 7150
10	100	4999	7200

An internal base frequency of 1MHz is selected for the generator. Divider values are calculated according to the formula in section 8.3.1.

Motion control using a stepper motor is primarily concerned with the destination, which must be reached with absolute precision. This destination is therefore dealt with separately in the user program. Destination is loaded into one register (R 2) and then continuously revised intermediate positions are loaded into another register (R 1) in the counting module.

Intermediate positions are only noted by the software, which means that, as a result of program waits, small displacements may result which are of no consequence. Destination is handled directly, with the generator being stopped when the end value in register 2 is reached. The generator only runs if IN-B_2 is high, i.e. if R2CO_1 (latched) has not yet been switched back.

The GRAFTEC structure shown on the following page has been designed for optimum clarity. For this reason, two practically identical step sequences are represented, the left sequence for forwards and the right for reverse. When the program is running, its progress can be viewed on line using the Graftec editor.


1
initialization
101
()
121
141
2
102
(start on, forw)
103
(start on, backw)
10
rampstep 0
30
rampstep 0
110
(countvalue 0?)
130
(countvalue 0?)
11
rampstep 1
31
rampstep 1
111
(countvalue 1 ?)
131
(countvalue 1 ?)
12
rampstep 2
32
rampstep 2
112
(countvalue 2 ?)
132
(countvalue 2 ?)
13
rampstep 3
33
rampstep 3
113
(countvalue 3 ?)
133
(countvalue 3 ?)
14
rampstep 4
34
rampstep 4
114
(countvalue 4 ?)
134
(countvalue 4 ?)
15
rampstep 5
35
rampstep 5
115
(countvalue 5 ?)
135
(countvalue 5 ?)
16
rampstep 6
36
rampstep 6
116
(countvalue 6 ?)
136
(countvalue 6 ?)
17
rampstep 7
37
rampstep 7
117
(countvalue 7 ?)
137
(countvalue 7 ?)
18
rampstep 8
38
rampstep 8
118
(countvalue 8 ?)
138
(countvalue 8 ?)
19
rampstep 9
39
rampstep 9
119
(countvalue 9 ?)
139
(countvalue 9 ?)
20
rampstep 10
40
rampstep 10
120
(total count ?)
140
(total count ?)
21
set break
41
set break
121
(wait break)
141
(wait break)
2
2

Comments on example EX_833

In addition to the "EXTN list" and definitions (EQU) of individual resources, counter and divider values (CV and DV) for this example are defined in two EQU tables.

The SB containing the actual user program for the H120 module is called from COB 0 in the usual way. In addition, each time the program returns to COB 0 the display is refreshed.

SB 0 : Sequential Block

IST 1 : Initialization for System 1: Counter
Definition of input filters
Count Mode: Single Count

Initialization for System 2: Generator as in preceding examples

TR 101 : Empty, for GRAFTEC structural purposes only.

ST 2 : Empty, for GRAFTEC structural purposes only.

TR 102 : Start condition, forwards.

TR 103 : Start condition, backwards

ST 10/30 : System 1:
- reset counter
- load reg. 2 with max. no. of steps (CV_10)
- load reg. 1 with no. steps to 1st position (CV_0)
- define R2CO: C = R2 latched
- define R1CO: C < R1
- enable outputs
- switch on stepper motor driver

System 2:
- load divider with value for 1st position (DV_0)
- generator runs if IN-B = H
(runs until max. no. of steps has been reached)
- define internal clock frequency at 1 MHz
- start generator

TR 110/119 : Refresh counter contents and output status.
130/139 Test whether no. of steps reached for position concerned:
STL R1CO_1

ST 11/19 : System 1: Load next no. of steps in reg. 1
31/39 System 2: Enter next divider value for new frequency

ST 20/40 : Only enter divider value for last frequency position
(the next no. of steps is end value in reg. 2)

TR 120/140 : Refresh counter contents and output status.
Test whether total no. of steps reached:
STL R2CO_1

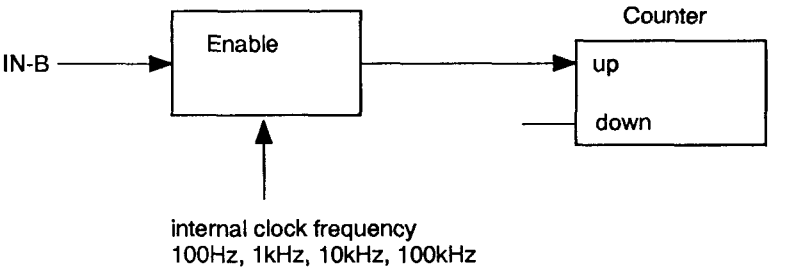
ST 21 : Load delay, switch on to run backwards, switch off stepper motor driver.

ST 41 : Load delay, switch on to run forwards (switch off backwards), switch off stepper motor driver.

TR 121/141 : Refresh counter contents and output status.
Await delay.

8.4 User programs for period measurement

The following circuit is the basis of period measurement:



If count mode is for example "Enabled if IN-B = H", for as long as IN-B is high, the previously defined internal clock frequency is routed to the counter via the enable circuit.

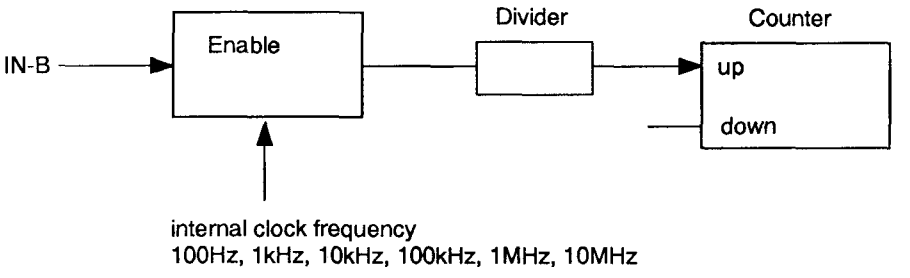
If a high signal is present at IN-B for 7.5 sec and clock frequency is 10kHz, during this period a previously reset counter will measure

7.5 sec * 10'000 pulses/sec = 750'000.

The result of this measurement is in ten-thousandths of a second and can be further processed by the user program.

Internal clock frequency and measuring time must be selected not to exceed the counter value of 999'999, otherwise an overflow wil occur.

Equally, consideration must be given to the fact that the counter's max. counting frequency is 333kHz; in this case therefore a max. clock frequency of 100kHz can be applied.



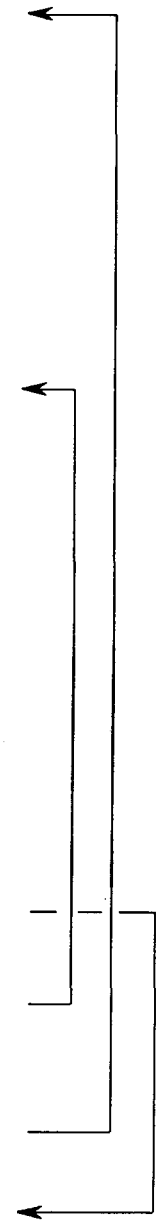
However, insertion of the divider enables clock frequencies up to 10 MHz to be used, in which case the measurement result is located partly in the counter and partly in the divider remainder.

If 10 MHz is used and divided by 1000 in the divider, the last 3 figures of the measurement result are located in the divider remainder.

8.4.1 Program structure

The structure of a user program for period measurement is as follows:

Start	Command no.
Input filter definition	160 - 165
Load divider (if used)	64
Define count mode:	
- Enabled Count between pos. edges on IN-B	248
- Enabled Count if IN-B = H	250
- Enabled Count between neg. edges on IN-B	252
- Enabled Count if IN-B = L	254
Enable divider IN-A: *)	
- Divider IN-A	104
Select clock frequency:	
- Divider IN-A: Single Shot; 100 Hz	168
- Divider IN-A: Single Shot; 1 kHz	169
- Divider IN-A: Single Shot; 10 kHz	170
- Divider IN-A: Single Shot; 100 kHz	171
- Divider IN-A: Single Shot; 1 MHz	172
- Divider IN-A: Single Shot; 10 MHz	173
Ready for measurement	
Evaluate measurement:	
- Read counter value and outputs (CFB RCF_x)	0
- Read divider remainder (CFB RDR_x)	8
- Poll "Enable" (STL ENAB_x)	
- Poll "IN-B" (STL INB_x)	
For new measurement:	
- Zero counter (load 0)	
If divider used:	
- Disable count	236
- Zero divider remainder (load 0)	65
End of measurement	



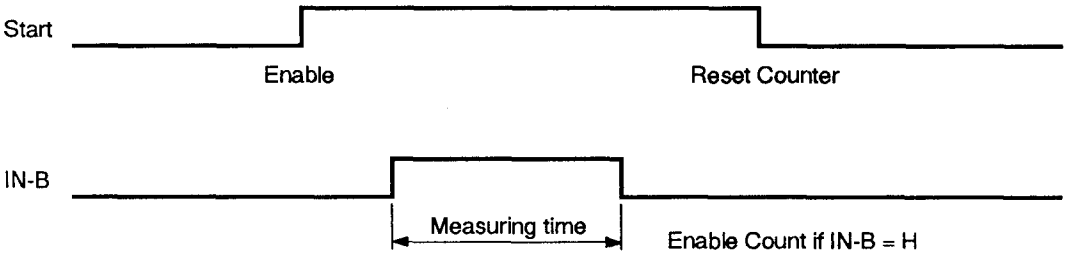
*) Divider "A" must always be enabled. If the divider is not used, it is bypassed with this command.

8.4.2 Period measurement without divider

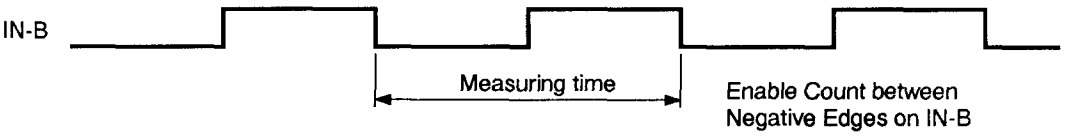
Task:

When the "Start" input is powered up, the counting and measuring module should be ready to measure a pulse at IN-B of system 1. Measurement should be to a resolution of 10 uS, i.e. a measuring frequency of 100 kHz should be used. When the "Start" input is powered off, the counter should be reset.

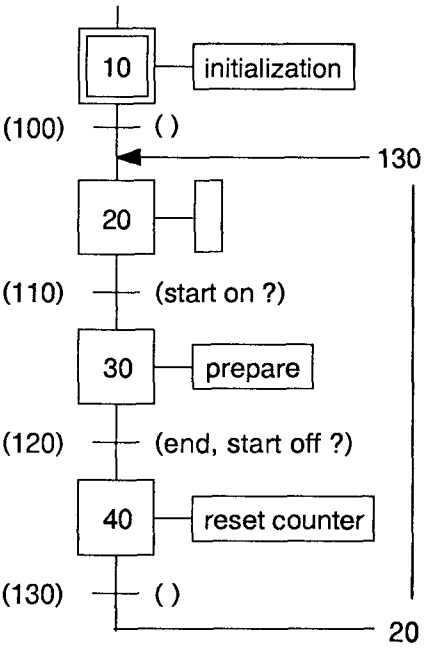
Measuring time should be shown continuously on a PCA2.D14 display.



Alternative: Measuring time between negative edges



Structure of example "EX_842"



The SB containing the actual user program for the H120 module is called from COB 0 in the usual way. In addition, each time the program returns to COB 0 the display and flags are refreshed.

SB 0 : Sequential Block.

IST 10 : Initialization consists of:
- defining input filters
- defining count modes
- enabling divider "A"

TR 100 : Empty, for GRAFTEC structural purposes only.

ST 20 : Empty, for GRAFTEC structural purposes only.

TR 110 : Refresh counter contents and output status.
Test whether input "Start" is on.

ST 30 : Determine clock frequency, enable measurement.

TR 120 : Refresh counter contents and output status.
End measurement:
- Enable = L AND
- Input "Start" = L

ST 40 : Reset counter (load zero).

TR 130 : Empty, for GRAFTEC structural purposes only.

8.4.3 Period measurement with divider

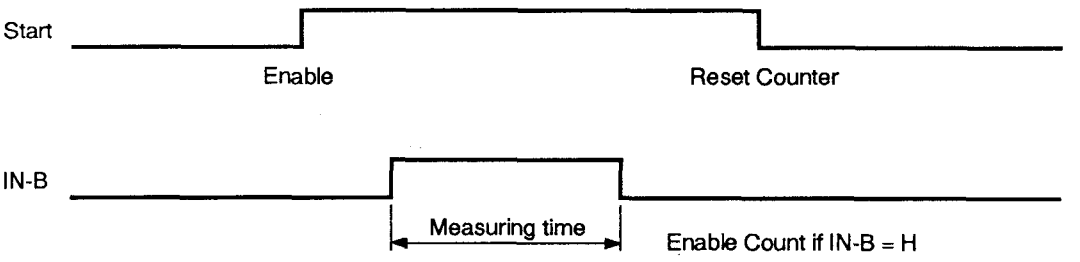
Task: (same as previous example, but with different values)

When the "Start" input is powered up, the counting and measuring module should be ready to measure a pulse at IN-B of system 1. Measurement should be to a resolution of 0.1 uS, i.e. a measuring frequency of 10 MHz should be used.

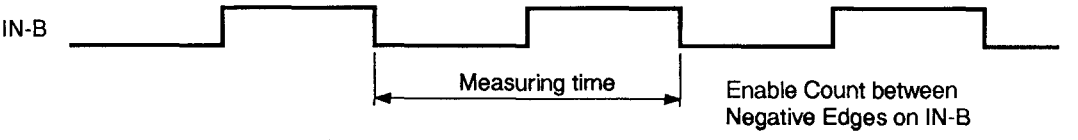
Counting takes place with an intermediate divider. The divider is loaded with 999, i.e. every 1000th signal is transmitted to the counter with the rest going to the divider remainder. The last 3 figures are therefore always located in the divider remainder.

When the "Start" input is powered off, counter and divider remainder should be reset.

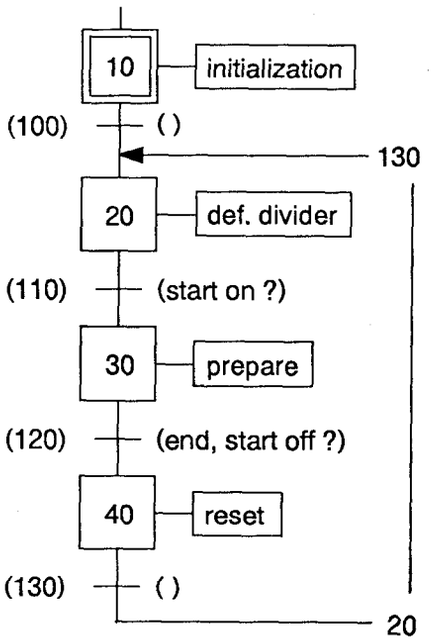
Measuring time should be shown continuously on a PCA2.D14 display, with the counter value above and divider remainder value below.



Alternative: Measuring time between negative edges



Structure of example "EX_843"



The SB containing the actual user program for the H120 module is called from COB 0 in the usual way. In addition, each time the program returns to COB 0 the display and flags are refreshed.

SB 0 : Sequential Block

IST 10 : Initialization consists of:
- defining input filters

TR 100 : Empty, for GRAFTEC structural purposes only

ST 20 : Definitions:
- load divider
- count mode
- enable divider "A"

TR 110 : Refresh counter contents and output status.
Read divider remainder.
Test whether "Start" input powered up.

ST 30 : Determine clock frequency, enable measurement.

TR 120 : Refresh counter contents and output status.
Read divider remainder.
End measurement:
- Enable = L AND
- Input "Start" = L

ST 40 : Reset counter.
Disable count.
Reset divider remainder.

TR 130 : Empty, for GRAFTEC structural purposes only.

8.4.4 Period measurement throughout more than one period

This example illustrates how two counting systems can be combined, and also shows a more complex user program, which handles both counting systems in the same execution cycle.

If a task requires period measurement throughout more than one period, the two counting systems of a PCD4.H120 module must be combined. One system carries out the actual period measurement, while the other counts the periods.

Task:

Signals with a frequency of 1 - 100 Hz should be measured with a resolution of 0,1 µS. Each measuring cycle should consist of 100 periods.

If the PCD "Start" input is switched on and off again, a single measurement is executed. If the PCD "Start" input remains switched on, the measurement is executed, the result is stored in 2 registers (R 1000 and R 1001) and there is a 5 second delay until the next measurement is started. The result is stored in the next 2 registers. If 1000 measurements are recorded, PCD output 46 becomes active.

PCD input 2 clears registers 1000 - 2999.

If PCD input 1 is high, two D14 display modules are active. One display should show counter status above and content of divider remainder below, while the other should show the number of periods above and the delay below.

For example, if the display shows the following measurement result

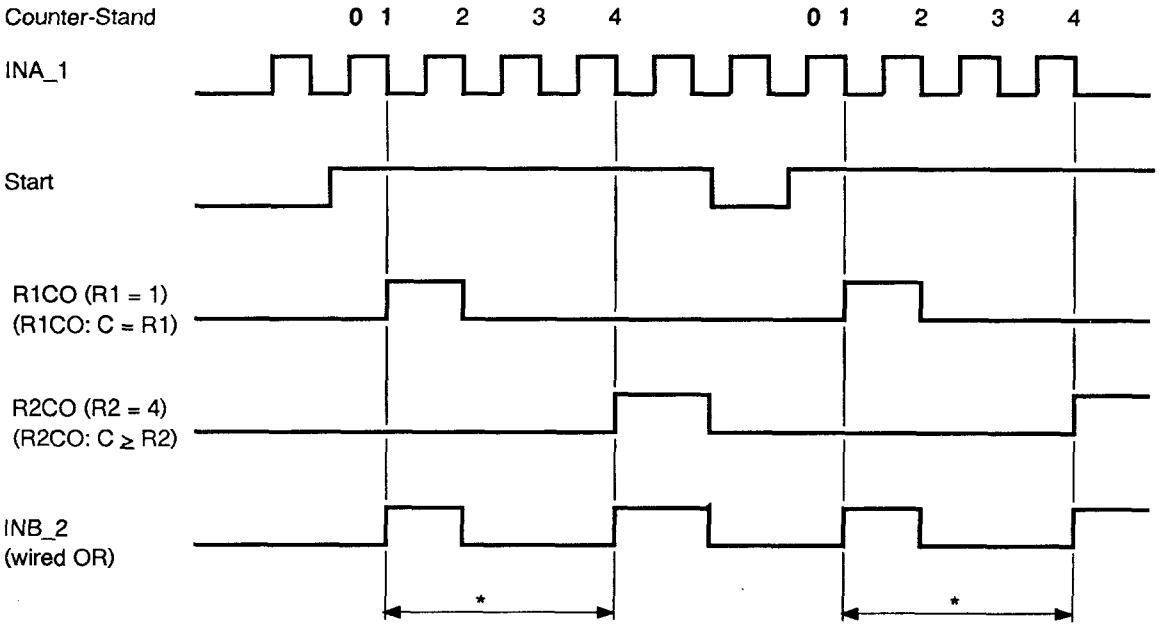
above	below
101067	000815

this should be interpreted as

10.1067815 S or 10106.7815 mS or 10106781.5 µS

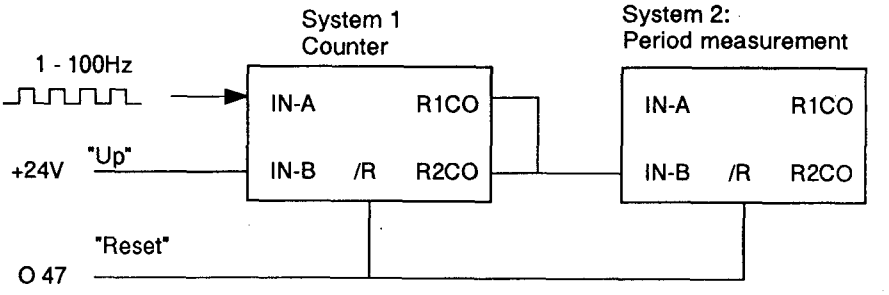
The principle of measuring throughout more than one period is demonstrated by the following diagram:

(Measurement is over 4 periods)



* Enable Count between positive Edges on INB_2

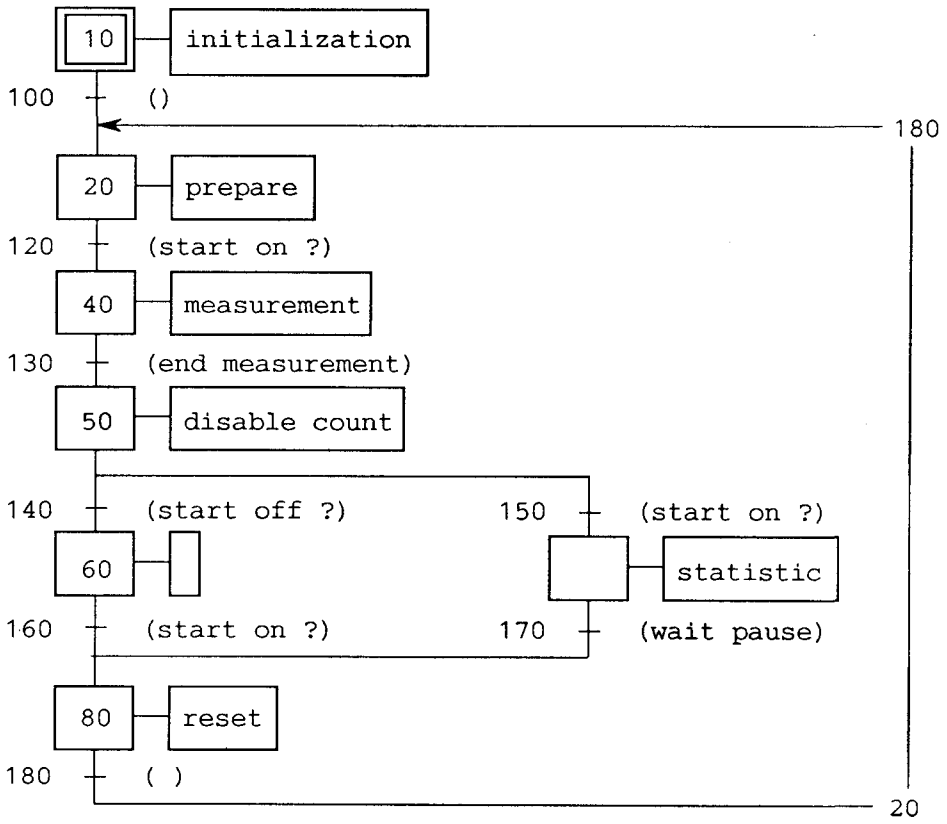
Structure:



Module outputs R1CO_1 and R2CO_1 (open collector) are wired directly to module input INB_2.

For a "Restart Cold" to execute an automatic hardware reset of the H120 module, both /RESET inputs on PCD output 47 should be wired. At the start of the program this output is kept low for a short time, to force a hardware reset.

The following GRAFTEC structure forms the basis of example "EX_844":



This program will run for demonstration purposes. However, in practice the D14 display modules ought not to be updated continuously. For a practical application, the read routines for counters and the divider remainder would also have to be processed in a single place and once only to read the measurement results and to transfer them to the result registers. For higher frequencies the "Read Counter and Flag" routine (mainly in TR 130) ought to be replaced by an improved routine, which would only poll specific output flags.

For any practical case, it would be useful to check whether a direct program structure with wait loop would not be more suitable. Practice shows that a controlled program structure with

measurement - await end of measurement - evaluate measurement - react to evaluation - enable new measurement - etc.

gives the best results, since no synchronisation can arise between the measurement and the course of the program. However, such a structure requires this part of the program to run in its own CPU.

Comments on example "EX_844"

A hardware reset is executed in XOB 16 so that the H120 module is also reset when a "Restart Cold" is done. (See also section 7.3.8 Reset).

The SB containing the actual user program for the H120 module is called from COB 0 in the usual way. In addition, each time the program returns to COB 0 the displays and flags are refreshed, if the "Display" input is high.

SB 0 : Sequential Block.

IST 10 : Both systems are initialized:
- for system 1 all outputs and the input filter
- for system 2 the input filter only, i.e. the input filter is bypassed for period measurement

TR 100 : Empty, for GRAFTEC structural purposes only.

ST 20 : Preparation for measurement:
- Load divider. This should not be done in IST, since at the end of the program the divider remainder is loaded with 0, after which a "Write to Divider" must follow.
- "Count Mode" for the measurement is defined: "Enabled Count between pos. edges on IN-B"
- The divider is enabled, so that signals from the internal clock generator are transmitted to the counter.
- The number of periods +1 is indicated, throughout which measurement will take place. This could also happen in IST. However, the value could not then be changed online.

TR 120 : Start of measurement is awaited.

ST 40 : Measurement is activated, whereby for system 2
- 1 measuring cycle is enabled (single shot)

and for system 1
- count mode is defined and
- outputs are enabled.

With the next rising edge at IN-B (2) measurement starts, stopping with the next-but-one rising edge. (See diagram).

TR 130 : The program must ascertain the end of measurement, in order to continue in its course. There are various ways of determining the end of measurement.

The enable flag is disregarded for reasons explained in section 7.3.6.

According to the diagram, measurement is certainly over when output R2CO_1 is high. This output and its flag are polled, after all flags have previously been refreshed.

The best designed program would only refresh the output flags in this TR and poll the R2CO_1 flag. For demonstration purposes, however, both counters and the divider remainder of period measurement are also refreshed and, if the "Display" input is on, all values are displayed.

However, in a concrete application the results are only of interest after measurement is concluded. These results would be obtained in the next STEP and then either evaluated directly or stored in PCD registers.

ST 50 : After measurement is completed, counting is disabled for both systems. This is not absolutely necessary, but conditions are more closely defined this way.

In practice, the measurement results should be processed here in ST 50, as described for TR 130, and evaluated or stored.

TR 150/150: Is "Start" input powered on or off?

- "On": Continuous measurements with statistics
- "Off": Individual measurements

ST 60 : Empty, no action. Continue from TR 160.

ST 70 : Start 5 sec. delay

Statistics of measurement results (optional). Note that indexing has been used in the D14 routines, ensure that no conficts arise if the index register is used again. The relevant index register value should be stored in a PCD register and reloaded on each measuring cycle.

TR 160 : For operation of individual measurements, check whether input "Start" is "On".

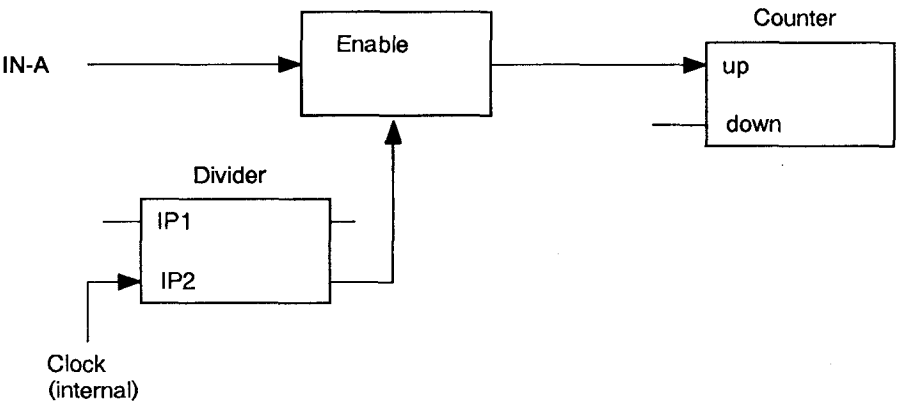
TR 170 : Await delay.

ST 80 : Reset both counters and divider remainders, load with zero.

TR 190 : Empty, for GRAFTEC structural purposes only.

8.5 User programs for frequency measurement

The following circuit is the basis of frequency measurement:



The internal clock generator, through the divider, controls the enable circuit. The enable circuit opens the time window for the duration of measurement as defined by the user program, and directs the signals arriving at input IN-A to the counter. If the time window is defined as 1 sec., after one measuring period the counter value corresponds to the frequency in Hertz.

To obtain a time window of 1 sec., the internal clock generator should be set to 1 kHz with a divider of 1000.

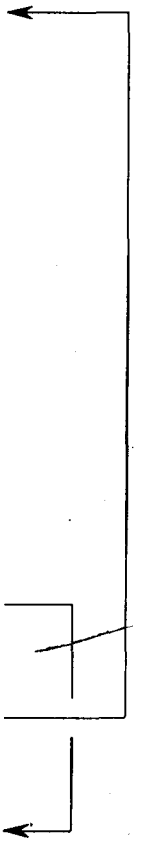
The following standard program for continuous frequency measurement opens the time window after measurement is started for the defined time, normally 1 sec. Thereafter the time window remains closed for the same amount of time and is then opened again for the defined timespan, etc.

To influence the course of the program, the "Enable" element can be polled under "ENAB" (such as "R1CO" or "CCO").

8.5.1 Program structure

The structure of a user program for continuous frequency measurement is as follows:

Start	Command no.
Input filter definition	160 - 165
Load divider	64
Enable divider IN-B: - Divider IN-B	112
Define count mode: - Enabled Count between positive edges on IN-B	254
Select clock frequency:	
- Divider IN-B: Single Shot; 100 Hz	176
- Divider IN-B: Single Shot; 1 kHz	177
- Divider IN-B: Single Shot; 10 kHz	178
- Divider IN-B: Single Shot; 100 kHz	179
- Divider IN-B: Single Shot; 1 MHz	180
- Divider IN-B: Single Shot; 10 MHz	181
Ready for measurement: - Read counter value and outputs (CFB RCF_x) - Poll whether "Enable" is high (STH ENAB_x)	
Execute measurement	
Evaluate measurement: - Read counter value and outputs (CFB RCF_x) - Poll whether "Enable" is low (STL ENAB_x)	0
For new measurement: - Reset counter (load with zero)	
End of measurement	

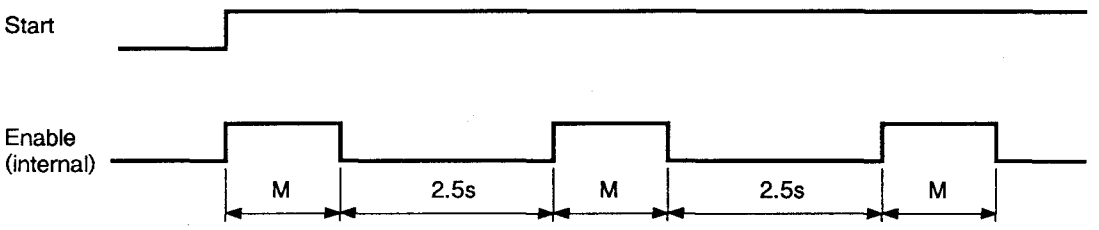


8.5.2 Continuous frequency measurement

Task:

When the PCD "Start" input is turned on, the frequency of a pulse chain present at input IN-A of the H120 module should be measured and displayed on a PCA2.D14 display module. Measuring time should be exactly 1 second, corresponding to a display of the measured frequency in Hz.

For as long as the "Start" input is switched on, measurement should be repeated every 2.5 seconds.



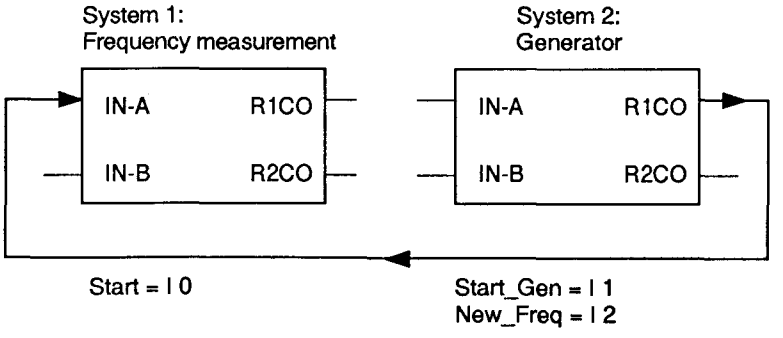
Example "EX_852" comprises frequency measurement on system 1 as described.

To make frequency measurement comprehensible without requiring a large number of devices, system 2 is set up as a variable frequency pulse generator. Frequency is preselected by 2 BCD switches on inputs 16 - 23, 00 = 50 kHz and 99 = 500 Hz. The generator is switched on or off suing PCD input 1, and PCD input 2 enters a new frequency value (divider value).

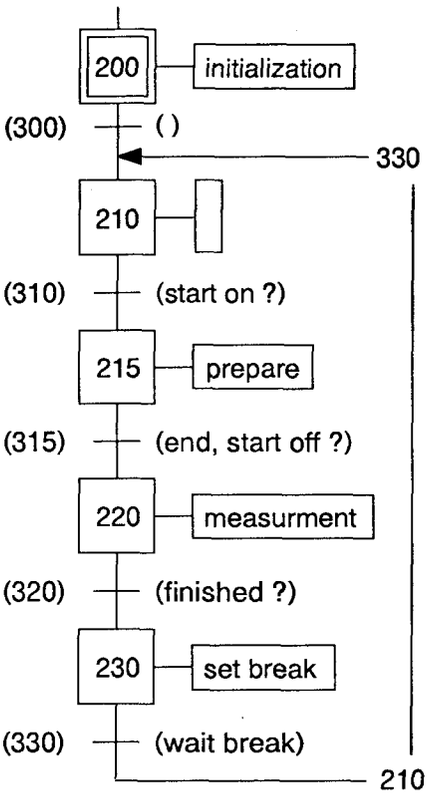
$$\text{frequency} = \frac{100'000\ \text{Hz}}{2 * (\text{BCD input value} + 1)}$$

$$\text{BCD value} = 19 \longrightarrow \frac{100'000}{2 * (19 + 1)} = 2500\ \text{Hz}$$

Structure:



Structure of example "EX_852"



The SB containing the actual user program for the H120 module is called from COB 0 in the usual way. In addition, each time the program returns to COB 0 the display and flags are refreshed. The "Enable" flag shown at output 39 will be of particular interest in this example.

SB 0 : Sequential Block.

IST 200 : The initialization consists of:
- defining the input filters
- loading the divider
- enabling divider "B"
- defining count mode

TR 300 : Empty, for GRAFTEC structural purposes only.

ST 210 : Empty, for GRAFTEC structural purposes only.

TR 310 : Refresh counter contents and output status.
Poll start condition.

ST 215 : Reset counter (load zero).
Select internal clock frequency.

TR 315 : Refresh counter contents and output status.
Poll whether "Enable" is high. If time window is open, measurement starts.

ST 220 : Empty. The module carries out measurement independently.

TR 320 : Refresh counter contents and output status.
Poll whether "Enable" is low. If time window is closed again, measurement is over.

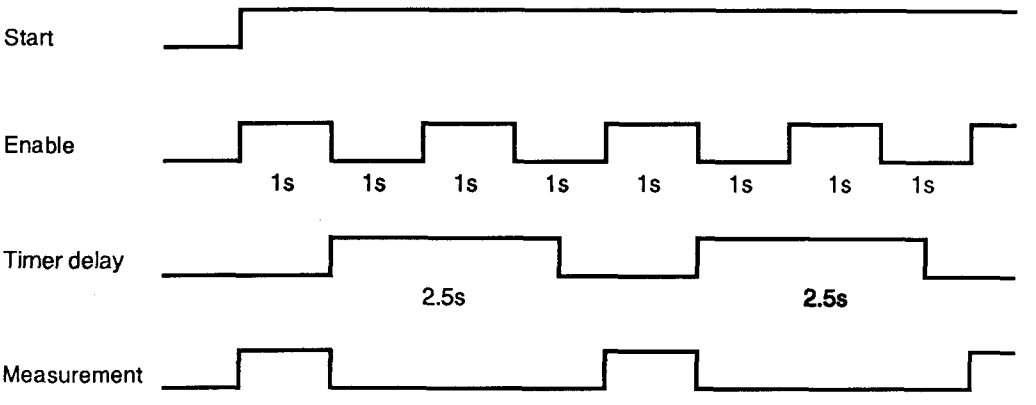
ST 230 : Measurement is over. Counter value is transferred to a PCD register for the purposes of display or further processing. (Not significant for course of program). A timer is loaded for the delay.

TR 330 : Refresh counter contents and output status.
Await delay.

Jump back to ST 210. If "Start" is still on, the counter is reset and a new measurement starts as soon as the time window next opens after the delay.

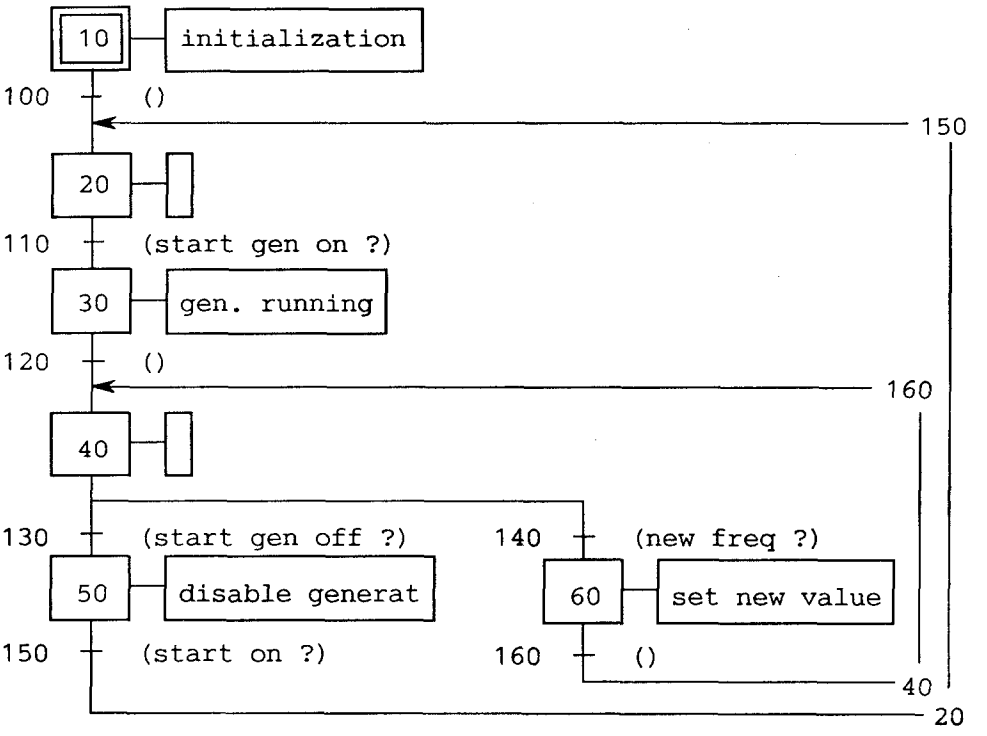
If no delay is switched, the next measurement takes place when the time window opens again after the previous measurement.

Actual course of frequency measurement with delay:



Without the delay, measurements occur whenever "Enable" is high.

To complete the picture, the GRAFTEC structure of the pulse generator is also reproduced:



From:

Company
Department
Name
Address

Tel.

Date

To:
SAIA AG
Industrial Electronics
Dept. SPS210

CH-3280 Murten
SWITZERLAND

Manual PCD4.H120

If you have any suggestions concerning the SAIA PLC, or have found any errors in this manual, brief details would be appreciated.

SAIA AG
Industrial Electronics and Components
CH-3280 Murten/Switzerland

Telephone 037 727 111
Telefax 037 714 443
Telex 942 127

Representatives

Belgique Landis & Gyr Belge SA, Dépt. Industrie
Avenue des Anciens Combattants 190, B-1140 Bruxelles
☎ 02 729 02 11, Tx 65 930, Fax 02 726 23 31

Danmark Kemp & Lauritzen A/S
Roskildevej 12, DK-2620 Albertslund
☎ 045 42 64 48 22, Fax 045 42 64 10 46

Deutschland SAIA GmbH
Daimlerstrasse 1 K, D-6072 Dreieich
☎ 06103 8906—0, Fax 06103 89 06 66

España Landis & Gyr BC SA
Batalla del Salado 25, Apartado 575, 28045 Madrid
☎ 91 467 19 00, Tx 22 976, Fax 91 239 44 79

France SAIA Sàrl.
10, Blvd. Louise Michel, F-92230 Gennevilliers
☎ 1 4086 03 45, Tx 613 189, Fax 1 4791 40 13

Great Britain Burgess-SAIA Ltd.
Dukes Way, Team Valley, Gateshead, Tyne & Wear NE 11 0UB
☎ 091 487 7171, Tx 53229, Fax 091 487 1610

Italia SAIA S.r.l.
Via Cadamosto 3, 20094 Corsico MI
☎ 02 48600600, Fax 02 48600692

Nederland Landis & Gyr BV, Div. Electrowater
Kampenringweg 45, Postbus 444, NL-2800 AK-Gouda
☎ 01820 65 685, Tx 20 657, Fax 01820 32 437

Norge Malthe Winje & Co A/S
Haukelivn. 22, Postboks 531, N-1411 Kolbotn
☎ 02 99 25 00, Fax 02 99 25 00

Österreich SAIA-Burgess Gesellschaft m.b.H.
Schallmooser Hauptstrasse 38, A-5020 Salzburg
☎ 0662 88 49 10, Fax 0662 88 49 10-11

Portugal Infocontrol Electronica e Automatismo LDA.
Rua Alvarez Botelho-Lote 154, Alfragide sul, P-2700 Amadora
☎ 1-47 10123 29, Fax 1-47 10676

Suomi
Finnland Landis & Gyr Suomi OY
SF-02430 Masala
☎ 90-29 731, Tx 121 039, Fax 90-29 755 31

Sverige Beving Compotech AB
St. Eriksgatan 113a, Box 21 104, S-10031 Stockholm
☎ 08 31 47 80, Tx 10 040, Fax 08 34 31 42

USA After sales services; Maxmar Controls Inc.
99 Castleton Street, Pleasantville, New York 10570-3403
☎ 914 747 3540, Fax 914 747 3567

Australia Landis & Gyr (Australia) Pty Ltd
411 Ferntree Gully Road, P.O. Box 202, Mount Waverley, Vic. 3149
☎ 3 544-2322, Tx 32 244, Fax 3 543 7496

Argentina Electromedidor S.A.I.y C.
Defensa 320, RA-1065 Buenos Aires
☎ 1 337125, Tx 23 377, Fax 1 3319582

Printed in Switzerland 26/735 E1 7.92 K02