

6. Programming

6.1 Hardware configuration

The Module has to be declare in the hardware configuration of the PCD.xx7, this is done on the DB1, DB511 or DB1023.

The module identification is : **82h**

Number of input byte needed are : **8**

Number of output byte needed are : **8**

This module has to be out of the process image area, so the Input and Output address have to be over the address 256 (e.g. CPU 414).

In this example the module PCD2.H210 is on the first slot and its address are 2000 for input and output.

Address	Name	Type	Initial Value	Comment
0.0		STRUCT		
+0.0	Kennbyte1	CHAR	'M'	This identify this DB
+1.0	Kennbyte2	CHAR	'x'	as the Hardware configuration,
+2.0	Kennbyte3	CHAR	'x'	and the modules setting
+3.0	Kennbyte4	CHAR	'7'	
+4.0	Modul1	STRUCT		
+0.0	kenn	WORD	W#16#82	H210
+2.0	PANr	INT	0	No PA Update
+4.0	InCnt	INT	8	
+6.0	OutCnt	INT	8	
+8.0	InBase	INT	2000	PEB 2000 - 2007
+10.0	OutBase	INT	2000	PAB 2000 - 2007
+12.0	mask	BYTE	B#16#0	
+13.0	dummy_b	BYTE	B#16#0	
+14.0	dummy_w	WORD	W#16#0	
=16.0		END_STRUCT		
=20.0		END_STRUCT		

6.2 Access to the module function

The access to the module function is made through direct periphery access.

If you are setting value or giving command to the module, you will write to the periphery. On the other hand if you are reading value or checking status you will read the periphery.

Here is a list of the possible module accesses, the first column “Adr”, specify the offset address from the module base address.

Adr	T PAB	T PAW	T PAD	L PEB	L PEW	L PED
0	DIR	Config	Dest_24	Emerg	Signat	Actuell
1	Vmin	OutA2		OnDest		
2	Vmax	OutA3		DIR		
3	Acc	Start ***		Latch		
4	Load ***	Stop ***		E0		
5				E1		
6				E2		
7				E3		

6.2.1 DIR (PAB)

Command: select the motion direction setting the output A1 “DIR”, possible value are (0 or 1).

6.2.2 Vmin

Transfer the Vmin of the motion in the loading register, possible value are (1..255, see chapter 7)

6.2.3 Vmax

Transfer the Vmax of the motion in the loading register, possible value are (1..255, see chapter 7)

6.2.4 Acc

Transfer the Acceleration of the motion in the loading register, possible value are (1..255, see chapter 7)

6.2.5 Load

Command to transfer the loading register (Destination, V_min, V_max, Acc) to the working register. (*** no value is required)

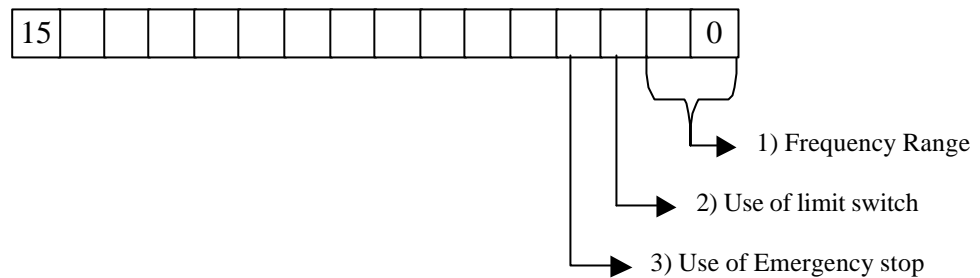
6.2.6 Config

This word is used to configure:

- 1) Frequency range
Specify in which frequency range the motor will work. Possible value (0..3, see chapter 7)
- 2) Use of module integrated function for limit switch
If the inputs E1 and E2 are used as limit switches, then the bit has to be set at the value TRUE.
- 3) Use of module integrated function for emergency stop.
If the input E0 is used for emergency stop, then the bit has to be set at the

value TRUE.

word config:



6.2.7 OutA2..A3

Command to write the value on the output “A2” or “A3”.

6.2.8 Start

Command to start the motion with the value actually in the working register. (*** no value is required)

6.2.9 Stop

Command to stop the motion. (*** no value is required)

6.2.10 Dest_24

Transfer the path for destination in the loading register (24 bits).

6.2.11 Emerg

Latched status of the input Emergency stop “E0”, only if the input “E0” is configured as Emergency stop input.

6.2.12 OnDest

If the path has been completed and the destination is reached. The flag is set to TRUE.

6.2.13 DIR (PEB)

Status of the Output Direction “DIR“ “A1”.

6.2.14 Latch

Latched status of the two Limit switch input “E1” and “E3”. This flag is a OR operation between the 2 inputs.

6.2.15 E0..E3

Read the actual status of the input E0..E3.

6.2.16 Signat

This signature word is use to identify the FPGA software.

6.2.17 Actuell

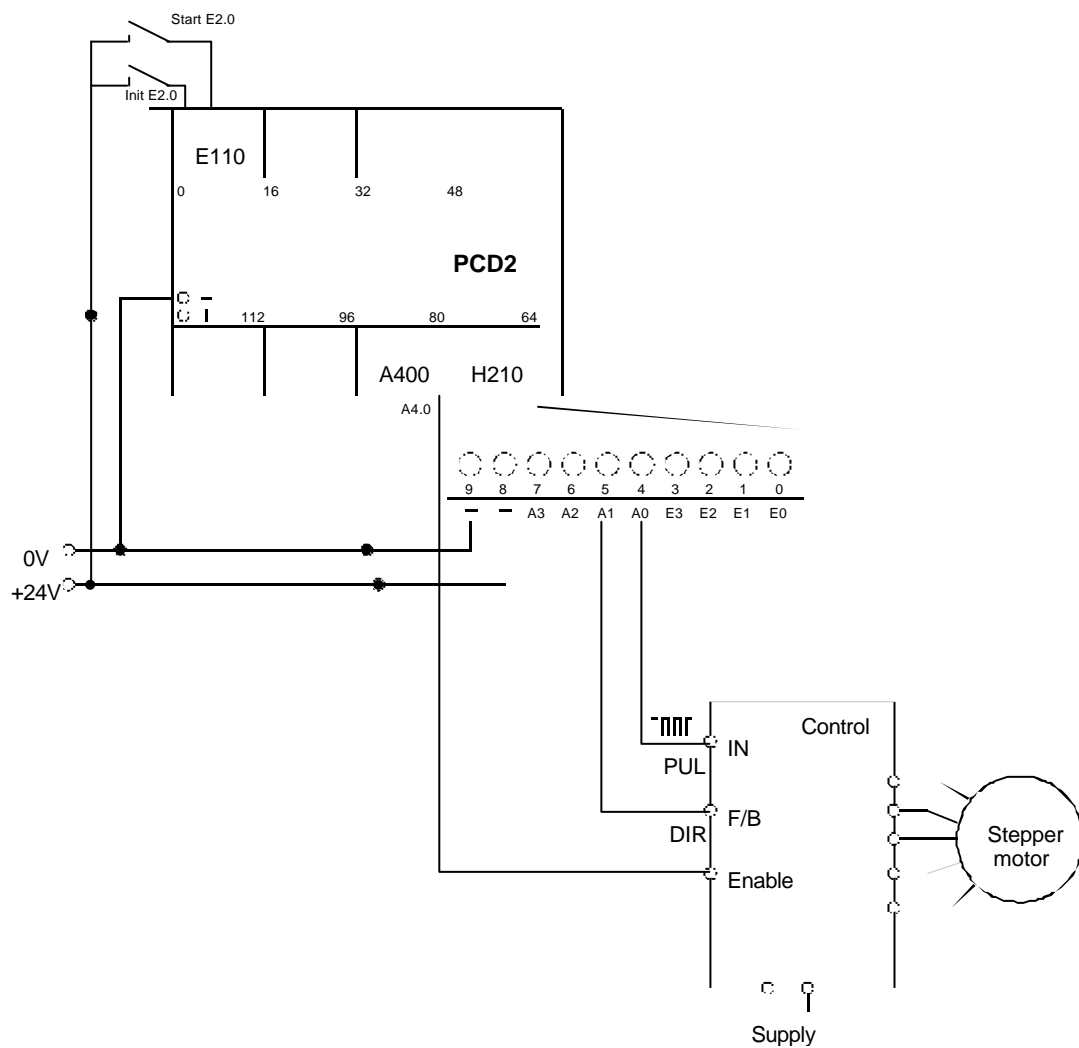
Read how many steps still have to be done in the motion

6.3 Programming example in AWL

To operate the PCD.H210 to control a stepper motor, few elements are needed, here is what we use in the example :

The individual elements are:

- PCD1 or PCD2 equipped with at least
 - 1 PCD2.H210
 - 1 PCD2.E110
 - 1 PCD2.A400
- Suitable drive electronics
- Stepper motor, possibly with sliding carriage
- Supply device



6.3.1 Given information

The following minimal program is suggested to commission a stepper motor in the easiest way.

Assume the stepper motor has the following characteristics:

Start-stop frequency:	38 Hz
Maximum frequency:	572 Hz
Acceleration:	2185 Hz/sec
Steps/revolution:	48

Example task: After power-up of PCD, switching ON input 'E2.0' should initialize the H210 module. Switching ON the input 'E2.1' cause the stepper motor to move 2100 steps. When the end position is reached, the motor will move automatically 1600 steps backward to reach the absolute position of 500 steps.

(The example can be found on the diskette).

6.3.2 Initialization of the H210

As we saw on the DB1, the base address will be 2000, for input and output.

```

L    W#16#0           // Frequency range 0,
T    PAW 2000          // Transfert the config to the module

L    4                // Vmin
T    PAB 2001          // Vmin in LOADING register

L    60               // Vmax
T    PAB 2002          // Vmax in LOADING register

L    70               // acceleration
T    PAB 2003          // Acceleration in LOADING register

L    0                // Reset the target position (safety)
T    PAD 2000

T    PAB 2004          // Transfert all the LOADING
                      registers to the WORKING registers

```

6.3.3 Start of first motion

```

L   2100           // PATH 2100 steps
T   PAD 2000       // Transfert the PATH in LOADING register
T   PAB 2004       // Transfert all the LOADING registers to the
                    WORKING registers
T   PAW 2003       // START motion

```

6.3.4 Detect END of first motion

```

L   PEB 2001       // LOAD the OnDestination flag
L   0
==I               // TEST if the position is reached
SPB st31          // Jump if position not reached

```

6.3.5 Invert direction and start second move

```

L   PEB 2002       // LOAD the actual direction
L   1
XOW               // Invert direction
T   PAB 2000       // Transfer new direction to module

L   1600           // LOAD the new PATH in the H210 module
T   PAD 2000       // Transfer PATH in LOADING register

T   PAB 2004       // Transfer all the LOADING registers to the
                    WORKING registers
T   PAW 2003       // START motion

```

The full example can be found on the diskette