

PCD2.M157 Configure and diagnostic

Description

The PCD2.M157 is a fast CPU in the PCD2.Mxx7 range. His higher speed is due to the two factors, one is Processor clock and the other one is compiled S7 code. This compiled code is the S7 software translated in the processor language, this part is totally transparent for user.

Memory

The PCD2.M157 also has an higher memory amount for S7 software. This memory is configurable by step of 64 KB until a maximum of 512 KB. But this S7 memory is taken from a shared memory of an total amount of 768 KB, this total amount of memory is shared between the S7 user program and the compiled code. If your S7 software is quite huge, it might be not possible to compile all of it, because of the compiler memory wouldn't be large enough. So you may have to make some choice and configure it. The possibilities are:

- 1) Select the S7 software memory as small as you need, like this you will leave as much memory as possible for the compiled code. Eg. If your S7 software use 175 KB of memory, you should then configure the S7 memory to $3 * 64 = 192$ Kb.
- 2) If your S7 software is very large, the compiled memory might be not large enough to contains all the compiled code, then you will have to fix the priority for the S7 block to be compile.
- 3) If the speed is not an priority, but you need a lot of S7 software memory, because of a large quantity of Data Block, then you will configure the S7 memory to a large amount and few memory will be left for the compiler.

We will se in the next chapter how to configure the S7 memory and how to fix the compiling priority.

1. Configuration

The configuration of the compiling block priority and also the amount of the S7 software is done through a S7 Data Block. This data block will be considered every time the PCD2.M157 is switch from STOP to RUN mode.

The structure of the data block is composed of character string with a size of maximum 64 char. The data block can contain a maximum of 1000 strings. Then it will be possible to describe the configuration through string word. This method of configuration has been choose to make the configuration easier, it will also be possible to configure the hardware and serial port in the future, by only writing in string the needed configuration.

For convenience this DB will be then referenced as “Configuration Data Block”.

The CDB can be one of those; DB1, DB511 or DB 1023 .

In order for the system to recognize the right DB, a signature has been made on it. The first string in the CDB is a predefine one. The string will have the value: “**SAIA xx7 CDB**”

1.1. Configuration of the memory

The default value (factory setting) is 256 KB, but the S7 memory can be configure by step of 64 KB, between 64 KB till 512 KB. To do so a string identifier is predefine in the CDB.

This identifier is MEM7, you just need to make a string with the identifier and the amount of desired memory, separated by space and colon.

Eg.: “**MEM7 : 384**”

1.2. Compiling block and priority

As said before, there is the possibility that having a huge S7 software, all of it will not be compile, then you will have to define the compiling priority, for your most important function. To do so you will have to know which are the time critical function in your application. Here are some tips:

- If a function is use quite often in one cycle (call many times the same FC)
- Cyclic interrupt OB, which are call often (like OB35 every 15 20ms)

When you have defined the most critical function in your application, then you need to configure the CDB. Here is the syntax:

Syntax

The space , upper case or lower case characters are not relevant.

The identification is: **“COMP:”**

For Function : **“FC”**

For function Block: **“FB”**

For Organization Block **“OB”**

To define a range of function **“-“**

Wild card for all of one kind **“*”**

Separator between two specification **“,”**

For example:

If we want to compile all the OB, then the FC11,FC12,FC13,FC14 and only FB20. The string will look like :

‘COMP: OB*,FC11-14,FB20’

If your declaration takes more than 64 characters or for better legibility, you can make the declaration in several lines.

‘COMP: OB*’

‘COMP: FC11-14’

‘COMP: FB20,FB30-FB40’

You can declare a range, event if some function in this range doesn't exist, like if FB35 was not in the project.

Adresse	Nom	Type	Valeur initiale	Commentaire
0.0		STRUCT		
+0.0	Identificator	STRING[64]	'SAIA xx7 CDB'	Signature
+66.0	Compil_prioity_1	STRING[64]	'COMP: OB*'	All the OB's
+132.0	Compil_prioity_2	STRING[64]	'COMP: FC11-14'	FC11 to FC14
+198.0	Compil_prioity_3	STRING[64]	'COMP:fb30-fb40,FC99'	FB30 to FB40 and FC99
+264.0	Memory	STRING[64]	'MEM7:128'	S7 software memory 128 KB
=330.0		END_STRUCT		

2. Diagnostic

2.1 Compiling diagnostic “Comp_diag”

The “Comp_Diag” can be use to find out if a Function block could be compiled or not.

This function doesn't have any calling parameter, but a value as to be given to it through the ACCU1.

There are also no parameter for return value, the result of the operation will be return trough the Status Word, bits CC0 and CC1 (condition code 0 and 1) .

2.2 ACCU1 (calling parameter)

It's a double word

Description	Irrelevant	Block Type	Block number	
Byte	3	2	1	0
Value	xx	0x08 for the OB 0x0C for the FC 0x0E for the FB	Block number which the status as to be check	

Example

To check if the OB 35 is compiled:

```
L dw#16#00080023
UC SFC230
```

2.3 Operation result

As said before the result of the operation is return in the STATUS WORD, in the bits CC0 and CC1.The meaning of these 2 bits after the operation are :

CC1	CC0	Description
0	0	The block is not loaded in the PLC
0	1	The block is not compiled
1	0	The block is correctly compiled
1	1	The parameter in ACCU1 when calling the SFC30 is not valid