Saia-Burgess Controls AG

Bahnhofstrasse 18 I CH-3280 Murten I Schweiz

T +41 (0)26 672 71 11 I F +41 (0)26 670 44 43

saia-burgess

Smart solutions for comfort and safety

Preliminary Documentation
V0.3

Differences between PCD2.M487 and the existing PCD2.M177

Saia-Burgess Controls AG

Bahnhofstrasse 18 I CH-3280 Murten I Schweiz

T +41 (0)26 672 71 11 I F +41 (0)26 670 44 43

Table of Content

1 INTRODUCTION 3

2 FEATURES 3

2.1 Amount of blocks 3

2.2 Amount of global ressources 3

2.3 STEP7 memory size 4

2.4 program execution / supported OB's 4

3 INCOMPATIBILITIES TO EXISTING XX7 5

3.1 Format of the analog inputs and outputs 5

3.2 Configuration of DP master and DP slave ..... Fehler! Textmarke nicht definiert.

3.3 Max. amount of attached periphery 5

4 COMPARISON OF XX7 SPECIFIC SFB'S & SFC'S 5

5 M487 – SPECIFIC FUNCTIONALITIES/SYSTEM FUNCTIONS 6

5.1 SFC 256 : Enable/Disable input interrupts 7

Saia-Burgess Controls AG

Bahnhofstrasse 18 I CH-3280 Murten I Schweiz

T +41 (0)26 672 71 11 I F +41 (0)26 670 44 43

Differences between M487 and the existing xx7

Differences_M487_M1x7

1 Introduction

This document lists the differences between the PCD2.M487 and PCD2.M177.

2 Features

2.1 Amount of blocks

Block	M487	M177
OB [3]	42 / see "supported OB's"	41 / see "supported OB's"
FB [3]	2048 (0 to 2047)	512 (o to 511)
FC [3]	2048 (0 to 2047)	1024 (0 to 1023)
DB [3]	2047 (1 to 2047)	1023 (1 to 1023)
SFC	512 (0 to 511) [1]	256 (0 to 255)
SFB	512 (0 to 511) [1]	256 (0 to 255)
SDB	[2]	2176 (0..2175)

Notes :

[1]) : The SFC's and SFB's are integrated in the system and the user cannot download them. The range is increased compared to the existing xx7 systems from 256 to 512 in order to have more reserves for additional functionalities and also for the OS-extensions.

[2]) : The logic behind the numbering of the SDB's is not known. The Siemens hardware- or DP-configurator creates SDB's mit numbers variing from 0 (CPU configuration) up to 2xxx (DP or FMS configuration and connection descriptions). On the M487 there will be no separate SDB-List and as the SDB's will be stored as a file in the file system we'll be able to download any number of SDB's as long as we have SRAM to store it.

[3]) The maximum length of a block is 64Kbyte on the M487 compared to 32K on M177.

2.2 Amount of global ressources

Area	M487	M177
Process image I/O	256/256 (0 to 255)	256/256 (0 to 255)
Max. I/O bytes	4096/4096	---
Flags	32678 (0.0 to 4095.7)	16384(0.0 to 2047.7)
Counter	512 (0 to 511)	256 (0 to 255)
Timer	512 (0 to 511)	256 (0 to 255)
Local Data	30720 byte	20480 byte

2.3 STEP7 memory size

The PCD2.M487 has 2MB SRAM which -except for a small amount that will be reserved for the other modules- , belongs to the PLC. The SRAM contains the global memory areas, the STEP7 program area , the diagnostic buffer of the PLC and evtl. some other PLC variables that must be remanent. As the available flash card for program backup have a capacity of 1 MB with 64K sectors the available memory size for the STEP7 programm will be 1MB As the Step7 blocks will be stored as files in a file system, the netto amount of RAM available for the user depends a lot on how many blocks are really loaded because of the ram needed for the file descriptors. The file system needs now for each file 68 byte overhead (descriptors + some pointers).

Some example calculations :

worst case →about 440 blocks (downloaded and as links in case of 256 SFB's and 256 SFC's) → file system needs 4400*68 = 292 kB → netto RAM for STEP7 programm = 731 kB. Of course this is a very unlikely case, because even if we wouldn't use a file system, the average size of a block in case of 896 kB user memory space would be only 240 byte inkl. STEP7 block header and tail !

A small survey of some big STEP7 customer programs that are known to the development show following distribution : Intellihome : 39 blocks → 180 kBytes, ESS : 98 blocks → 153 kBytes, IBL : 173 blocks → 90 kBytes. If we make based on these figures a more realistic assumption of 500 blocks (incl. SFC's and SFB's) we get following calcualtion : 500 * 68 = 34 kByte → available RAM for STEP7 program = 990 kB.

The M487 will give to the user the possibility –as on M177- to adjust this size in 64 K Blocks between 128Kbyte and 1 MB. On the xx7 this mechanism was introduced in order the user can achive a balance between the RAM used by the compiler and the RAM for the original blocks. As on M487 the compiled blocks will be in the DRAM, this adjustment has no implications about the size for the compiled blocks, but with this mechanism the user can decide if he wants to use part of the backup flash for saing data blocks.

2.4 program execution / supported OB's

Program/OB type	**M487**	**M177**
Cyclic	OB 1	OB 1
Time interrupts	OB 30 .. 38	OB 30 .. 38
Time of Day interrupts	OB 10 .. 17	OB 10 .. 17
Delay interrupts	OB 20 .. 23	OB 20 .. 23
Process alarms	OB 40 .. 47	OB 40 .. 47
Restart	OB 100	OB 100
Background	OB 90	n.a.
Asynchronous errors	OB 80 .. 87	OB 80 .. 87
Synchronous errors	OB 121 .. 122	OB 121 .. 122

3 Incompatibilities to existing xx7

Following incompatibilities exist :

3.1 Format of the analog inputs and outputs

The format of the analog inputs and outputs shall be right aligned instead of the left aligned format used in the existing xx7. This doesn't affect the other configurable formats like Siemens format or temperature values.

3.2 Programming and monitoring via F750 (DP Master)

On the M48, DP-Master is reinitialized in stop to startup transition only if the configuration is changed. → Communication doesn't abort –as on the older xx7-, if there are running FDL services during state transition of PLC e.g programming / monitoring with PG.

3.3 Max. amount of attached periphery

The max. amount of attached periphery (local or remote via fieldbus) shall not exceed 4096 output bytes and 4096 input bytes. Allthough this is a theoretical value, the xx7 today don't have this limitation.

4 Comparison of xx7 specific SFB's & SFC's

The following table lists all the 1x7 specific system functions and system function blocks that are implemented in the 1x7. The column "Impl." presents the current implementation status.

A question mark in the column M487 indicates that the writer is not quite sure if they should be implemented or not. This must be decided by the project team / market people.

SFB/SFC	Name	Description	M487	M177	Comment
SFB240	FLASH	Copy/restore data block to and from the user flash	YES	YES	
SFB254	WRITECS	Writes data to a memory area	n.a. ?	NO	Smart7
SFB255	READCS	Read data from a memory area	n.a. ?	NO	Smart7
SFC200	CONTROL	Enable/Disable MPI thru serial port (Port 1)	YES	YES	Eventually needs enhancement in order to enable MPI thru more than one port ???
SFC220	LON_INIT	Initialise LON	n.a.?	YES	Visible only if F8xx present
SFC221	NV_SEND	Send a NV (LON)	n.a.?	YES	Visible only if F8xx present
SFC223	MSG_SEND	Send an explicit message (LON)	n.a.?	YES	Visible only if F8xx present
SFC227	PC104_RD	Read data from	n.a.	NO	PC104

		DPR			
SFC228	PC104_WR	Write data to DPR	n.a.	NO	PC104
SFC229	PC104_ST	Read status of DPR	n.a.	NO	PC104
SFC230	RD_COMP	Get compiled block info	YES	YES	
SFC233	FAST_AI	Read and compute the average	n.a.	YES	Was originally written for Niehoff.
SFC239	WDOG	Trigger the watchdog	YES	YES	Perhaps an additional SFC to set the time ?
SFC240	COM_RCV	Read serial port data	n.a. ?	YES	This function can be replaced by the SFB's 12-14 using the transparent mode
SFC241	COM_SEND	Send serial port data	n.a. ?	YES	s.a.
SFC242	COMSTAT	Read port status	n.a. ?	YES	s.a.
SFC243	COM_INIT	Initialise serial port	n.a. ?	YES	s.a.
SFC244	COM_SIG	Read/Write modem signals	YES	YES	Evtl. new SFC in order to support also the modem signals of an F522 !
SFC245	B_INIT	Initialise serial port according to protocol	YES	YES	
SFC248	INTDIR	Bidirectional counter	n.a. ?	YES	If a counter functionality will be implemented, it will be a M487 specific SFC
SFC250	INP_INT	Enable/Disable onboard input interrupts	n.a.	YES	On M487 → SFC 256
SFC251	INITCNTR	Enable/Disable onboard counter	n.a.?	YES	If a counter functionality will be implemented, it will be a M487 specific SFC
SFC252	READCNTR	Read counter	n.a. ?	YES	If a counter functionality will be implemented, it will be a M487 specific SFC
SFC253	READ_SSI	Read onboard SSI	n.a.	NO	
SFC254	GRAY2BIN	Convert fray to binary code	n.a.	NO	
SFC255	TEST	Test routines for the production	???	YES	Probably something similar
FB2..FB6		FMS specific read/write	n.a.?	YES	Visible only if F700 present
SFC300	XControl	Enable/Disable MPI thru serial port (all ports)	YES	NO	port as parameter → see SFC 200
SFC344	XCOM_SIG	Read/Write modem signals (all ports)	YES	NO	port as parameter → see SFC 344

5 M487 – specific functionalities/system functions

5.1 SFC 256 : Enable/Disable input interrupts

Parameter	Type	Size	Range	Description
IRQ_NO (IN0)	INPUT	INT	0..3	Interrupt line, corresponds to IX0 to IX3
REQ_TYPE (IN1)	INPUT	BOOL	F / T	TRUE enables the interrupt, FALSE disables them
OB_NR (IN2)	INPUT	INT	40..47	The number of the OB which should be called when an interrupt occurs
OB_INFO (IN3)	INPUT	WORD	XXXX	Value that will be placed in the local data of the interrupt OB as start info
RET_VAL (OUT)	OUTPUT	WORD	YYYY	Error information

Possible SFC 256 specific error codes in RET_VAL :

Error Code	Description
0x0000	No errros
0x8080	IRQ_NO out of range
0x8081	OB_NR out of range

Programming example :

Calling the SFC 256 :

```
CALL   SFC      256
       IN0      = 1              // INT 0
       IN1      = TRUE           // Enable
       IN2      = 41             // OB 41 will be called
       IN3      = W#16#CAFE      // Start information
       RET_VAL  = MW 10          // MW 10 will contain the error info
```

Programming the Interrupt OB :

```
L      #OB41_MDL_ADDR           // The information is stored as the base address of
L      W#16#CAFE                // the module which initiates the interrupt
<>I                             // same ?
```

```
JC      CONT
...                                     // Do something
```

Read / Write onboard IO's

The onboard IO's can be accessed directly via byte access on peripheryaddress 0xFFFF (65535). This address is reserved and can't be used for normal peripherie modules!

Read Inputs

```
L       PEB 65535       // Ix0..3 are loaded into akku1 Bit0..3
```

Write Outputs

```
T       PAB 65535       // Bits 4..5 of akku1 are written to onboard outputs Ox4..5
```

5.2 SFC 300 : extended Enable/Disable MPI thru serial port

(→ enhanced SFC 200)

Parameter	Type	Size	Range	Description
Port (IN0)	Input	INT	ValisPort	Portnumber
DoStart (IN1)	Input	BOOL	F / T	new state of driver
RetVal (OUT0)	Output	INT	xxxx	return value

Possible SFC 300 specific error codes in RET_VAL :

Error Code	Description
0	OK (done)
1	working
-1	illegal port
-2	unknown error
-3	driver error

5.3 SFC 344 : extended Read/Write modem signals

(→ enhanced SFC 244)

Parameter	Type	Size	Range	Description
Port (IN0)	Input	INT	ValisPort	Portnumber
DTR (IN1)	Input	BOOL	F / T	DTR will be set to this state
DCD (OUT0)	Output	BOOL	F / T	current state of DCD
DSR (OUT1)	Output	BOOL	F / T	current state of DSR
RetVal (OUT2)	Output	INT	xxxx	return value

Possible SFC 344 specific error codes in RET_VAL :

Error Code	Description
0	OK
-1	illegal port
-2	specified port doesn't support modemsignals

5.4 Serial ports

- Port 1 and port 6: speed up to 115k
- New modes for RS232 and RS422: without hardware flowcontrol (RTS / CTS)

At the moment multi-drop-mode (MDM) isn't implemented on M487.

5.5 Programmingport (MPI on PGU)

- Baudrates 500k and 1.5M will be supported

5.6 Programming via serial port

- There will be gateway functionality soon. That means you can connect via serial line to M487 and access nodes on the MPI bus.

5.7 LON / FMS

This fieldbuses are currently not supportet/planed on M48x.