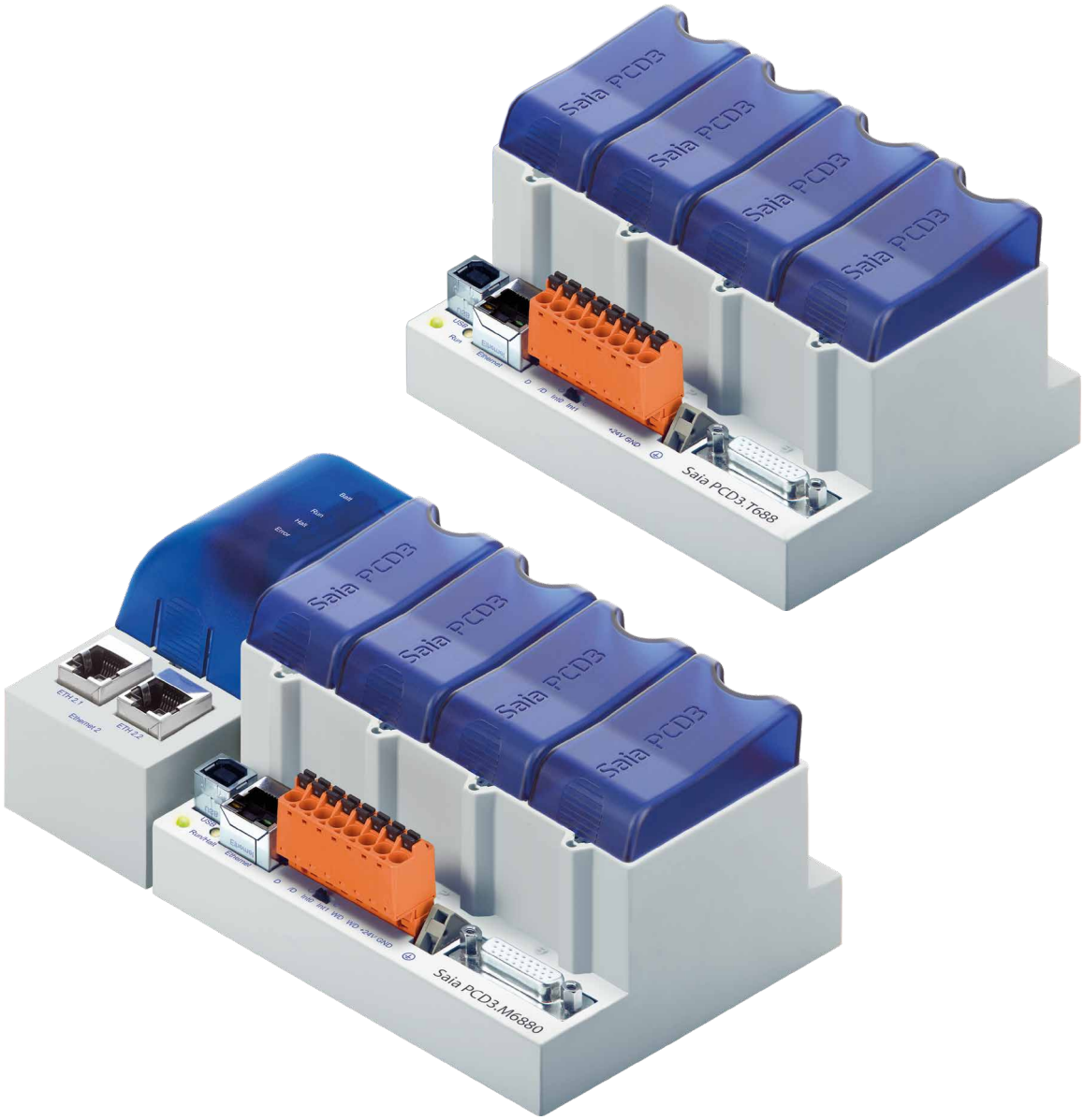


Saia PCD® Standby System

redundant automation solutions

Document 27-645 | Edition ENG 03 | 2022-02-28

0

0 Content

0

0

0.1 Document History

Version	Changes	Published	Comments
ENG01	2015-11-27 2016-03-11 2017-04-03	2015-11-27 2016-03-16 2017-04-03	First edition PCD3.Fxxx --> 26-857 new footer
ENG02	2017-04-26	2017-04-26	Ch3.1 - more possible failures listed
ENG03	2022-02-28	2022-02-28	New pictures with lasered products (no more printed and no labels anymore)

0.2 Trademarks

Saia PCD® is a registered trademark of Saia-Burgess Controls AG.

Technical changes are subject to the state of technology.

Published in Switzerland

1 Standby System Overview

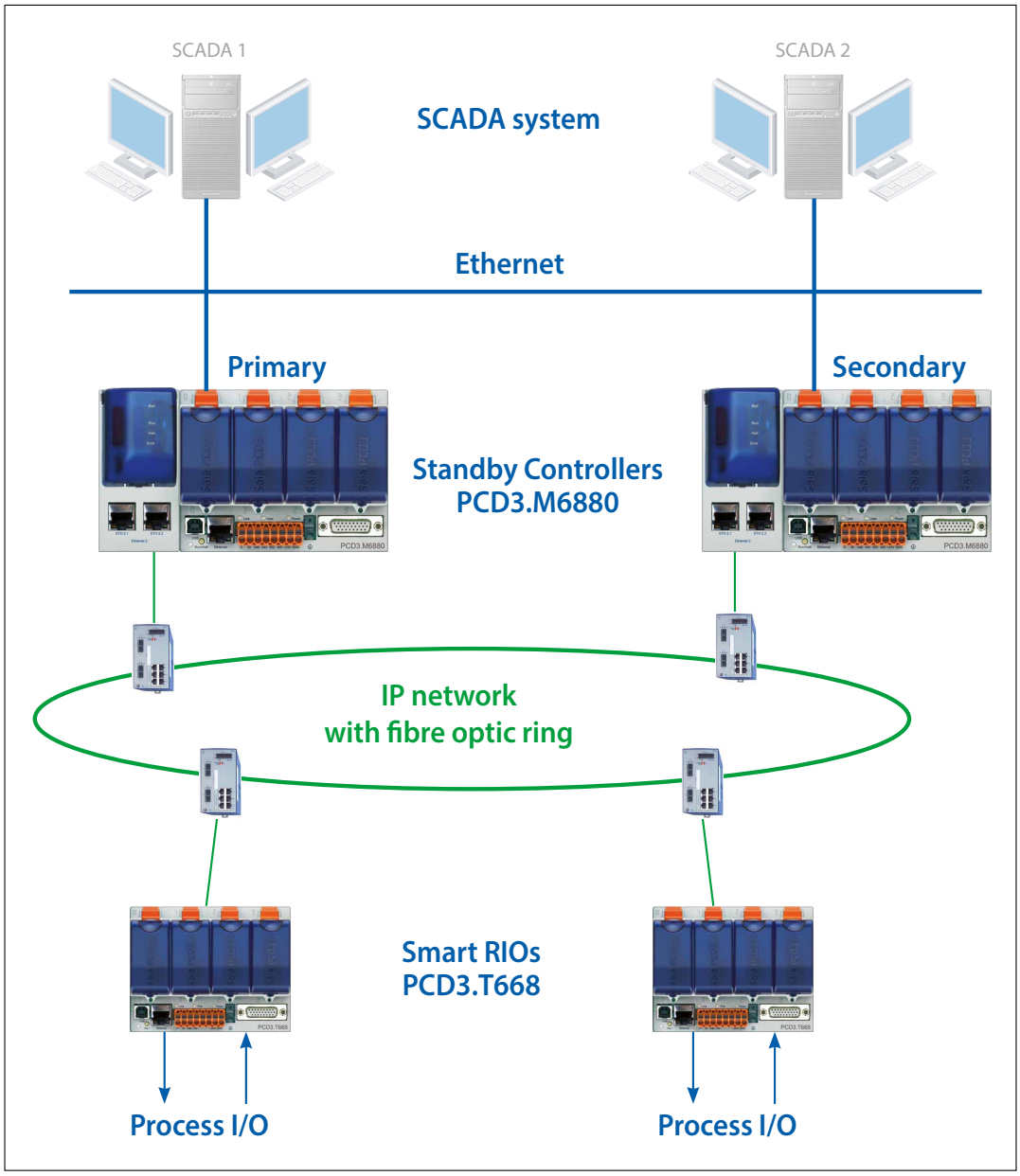
1.1 Introduction

The PCD3.M6880 Standby Controllers are for creating redundant automation solutions, to ensure the uninterrupted operation of systems and processes.

Standby (redundant automation) systems from SBC have the following characteristics:

- Based on the modular and robust PCD3 family, using standard modules.
- Simple system architecture to reduce costs.
- Standby processors with shared Ethernet Remote I/Os
  avoids the duplication of the inputs/outputs and the sensors/actuators.
- Programmable remote I/Os create intelligent decentralized nodes to provide additional reliability.
- The network uses standard Ethernet components, and can run over a standard Ethernet TCP/IP network along with other services.
- Easy engineering and commissioning, using the PG5 Project Manager to automatically generate the project.
- Uninterrupted switching from Standby to Active device.
- Standby controllers contain two processors. One processor runs the redundant program and monitors the active PCD. The second independent processor runs other non-redundant processes. This significantly increases the performance and flexibility of the system.
- Comprehensive diagnostic features to aid commissioning and fault finding.

1



Typical layout of a redundancy system with two PCD3.M6880 Standby devices and PCD3.T668 Ethernet Smart RIOs.

1.2 Terminology

The following definitions will provide a better understanding of the properties and operating principles:

Standby Controller — The PCD3.M6880 controller which supports the standby feature.

Primary PCD — The PCD which becomes the active device by default when the system is powered up, depending on the configuration.

Secondary PCD — The PCD which becomes the standby device on power up, and only takes over active control in the event of a fault on the active device.

Active PCD — The PCD whose CPU1 is in Active Mode, running the redundant program and controlling the inputs/outputs (PCD3.T668 RIOs).

Standby PCD — The PCD whose CPU1 is in Standby mode. It does not run the redundant program and the outputs (PCD3.T668 RIOs) are not controlled by this device.

Main CPU — CPU0 of the Primary or the Secondary PCD, which runs the non-redundant program. This program may be different on the Primary and Secondary devices.

Redundant CPU — CPU1 of the Primary or Secondary PCD, which contains the Redundant program. This program must be the same on the primary and Secondary devices. This CPU is either in Active mode and running the Redundant program, or in Standby mode and monitoring the Active PCD.

Redundant control solutions are created using two PCD3.M6880 Standby Controllers. The input/outputs (process signals) are connected and controlled via PCD3.T668 Ethernet smart RIOs. The RIO stations are connected to both controllers via an Ethernet connection. This means there is no need to have duplicate inputs, outputs, sensors and actuators. The two PCDs (primary and secondary) monitor each other. If the active PCD fails, the standby PCD takes over processing and control of the connected RIO stations. The process image (I/O) and the internal PCD media (F, R, T, C, DB) - the synchronization data - are continuously transferred from the active PCD to the standby PCD via the Ethernet connection. This ensures uninterrupted switching from the active to the standby PCD.

The Redundant CPU1 has two independent Ethernet interfaces. The ETH 2.× interface is reserved exclusively for operating the PCD3.T668 RIO stations. The PCDs also synchronize their process data via the same interface. For security reasons, we recommend setting up this network as a ring structure with specific network components from third-party providers. We have had good experiences with the industrial Ethernet switches from Hirschmann.

Ethernet 2
(2 port switch)

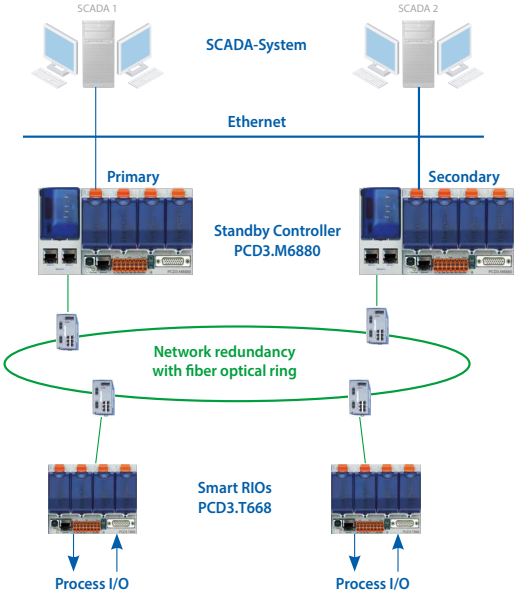
The ETH 1 interface on CPU0 is available for connecting and operating other systems and devices. For example, SCADA systems can be connected via this interface. SBC does not provide its own SCADA system for redundant automation solutions, but almost any system can be used. A single SCADA system, or an additional redundant SCADA system can be used if it supports redundant controllers. The PCD3. M6880 controllers provide detailed status and diagnostic information which can be evaluated by the SCADA systems.

Ethernet 1

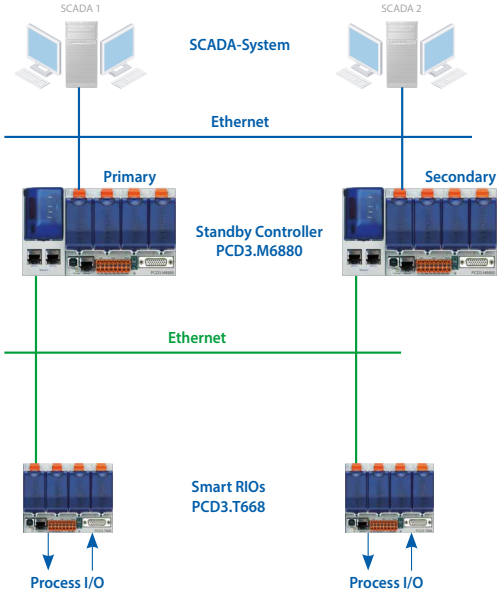
1.3 Designing the System

Redundant automation solutions can be achieved with various network topologies.

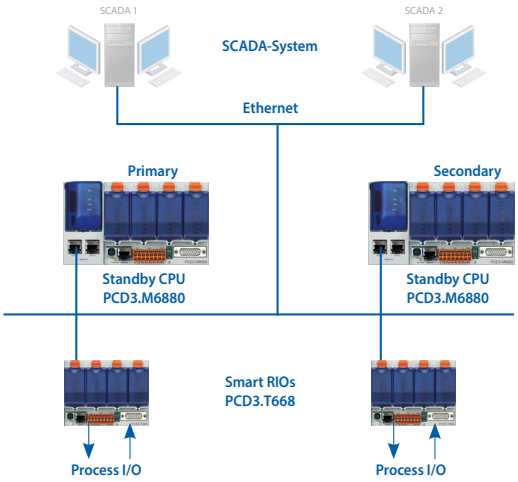
Physically separating the management network (SCADA systems) and the network for the remote I/Os is recommended. We also recommend setting up the remote I/O network in a ring structure using fibre-optic network components. This significantly increases the performance, security and, above all, the network availability and thus the system reliability. Standard devices from third-party providers can be used for the network components (switches). We have had good experiences with the switches (RS30) from Hirschmann. However, the networks can also be set up with standard components in a star structure. A shared physical network for the remote I/Os and management systems is also possible, but availability of the system will be reduced accordingly.



Recommended network topology with physically separate networks and a fibre-optic ring



Physically separate networks in a star topology with standard components



Shared physical network in a star topology with standard components

1.4 Order details

PCD3.M6880	Modular PCD3 standby controller with 2 Ethernet TCP/IP ports and a coprocessor for standby operation
PCD3.T668	Smart RIO for standby system, for connection to the PCD3.M6880 CPU1

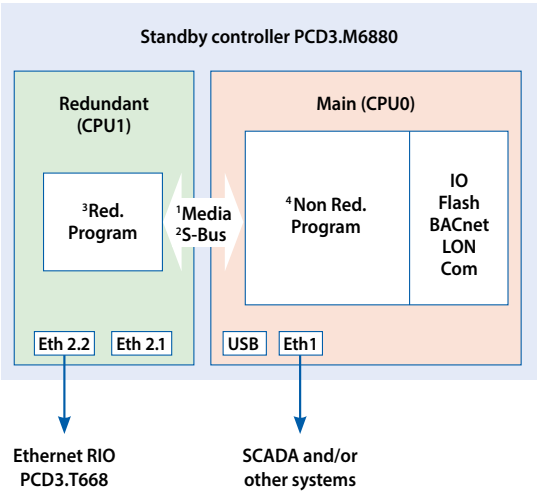
1.5 PCD3.M6880 Standby Controller

1.5.1 Architecture of the PCD3.M6880

The PCD3.M6880 standby controller has two independent processors (CPU0 and CPU1). Both processors have their own independent PCD media (F, R, T, C, DB/TX).
The redundant CPU1 runs the redundant user program and controls the shared inputs/outputs of the PCD3.T668 remote I/Os. The redundant programs in the primary and secondary PCD3.M6880s are identical. During normal operation, only the active PCD runs the redundant program. CPU1‘s internal used PCD media (F, R, T, C, DB/TX) are transferred from the active to the standby PCD via the Ethernet interface 2 (ETH2.x). In the event of a fault, the standby PCD takes over operation without interruption, and runs the redundant program using the last process image from the active PCD.

Depending on requirements, the user programs of the main CPU0 can be different in the primary and secondary PCD3.M6880. CPU0 has the same capabilities as a standard PCD (e.g. PCD3.M5560). Local I/Os in the PCD‘s slots, and the I/O expansion modules, are controlled by CPU0. External systems and devices (SCADA systems, web browsers and other external devices) communicate only with CPU0. CPU0‘s internal PCD media (F, R, T, C, DB) are not synchronized between the active and standby PCD.

CPU1‘s program cannot directly access the local IOs or CPU0‘s media (and vice versa). Data is exchanged between CPU0 and CPU1 using a data exchange mechanism. The data to be exchanged (PCD media) are define in global symbol files. This data is automatically exchanged between CPU0 and CPU1 on each program cycle.



PCD3.M6880

1 Data Media Transfer (Exchange Range or/and CSF/FBox)
2 S-Bus GWY CPU0 to CPU1 (2 different S-Bus address)
3 Redundant program on CPU1 runs only if active. Same program on both PCDs.
4 Non-redundant program can be different in both PCDs.

1

1.5.2 PCD3.M6880 Technical Data

Property/function	PCD3.M6880 Main CPU0	PCD3.M6880 Redundant CPU1
Number of inputs/outputs	1023	—
or I/O-module slots	64	—
I/O expansion connection for PCD3.C module holder	Yes	—
Processing time [µs] bit operation	0.1…0.8 µs	
word operation	0.3 µs	
Real time clock (RTC)	Yes	

On-Board memory

Program memory, DB/TEXT (Flash)	2 MByte	
User memory, DB/TEXT (RAM)	1 MByte	128 KByte
Flash memory (Program, S-RIO and configuration)	128 MByte	
User flash file system (INTFLASH)	128 MByte	—
PCD media:		
Register	16384	16384
Flag	16384	16384
DB/TEXT	8192	8192

On-Board interfaces

USB 1.1	Yes	No
Ethernet 10/100 Mbit/s, full-duplex, auto-sensing/auto-crossing	ETH1	ETH2.x (2 port switch)
RS-485 on terminal block (Port 2) or RS-485 Profibus-DP Slave, Profi-S-Net on terminal block (Port 2)	up to 115 kbit/s up to 187.5 kbit/s	—
Optional communication interfaces		
I/O slot 0*: PCD3.F1xx modules for RS-232, RS-422, RS-485 and Belimo MP-Bus	Yes	No
I/O slot 0…3 up to 4 modules or 8 interfaces*: PCD3.F2xx modules for RS-232, RS-422, RS-485, BACnet® MS/TP, Belimo MP-Bus, DALI and M-Bus	Yes	No

Other features

Communication protocols/systems (BACnet, Modbus, LonWorks®, DALI, M-Bus…)	As PCD3.M6860 without 2nd Ethernet	No
Automation server (web server, FTP server, e-mail, SNMP, flash file system…)	Yes	No
Connection and operation of PCD3.T668 remote I/O Number of supported RIO stations	No —	Yes 64
Connection and operation of PCD3.T665/T666 remote I/O Number of supported RIO stations	Yes 64	No —
Access to the I/O slots in the basic housing as well as to the PCD3.Cxxx I/O terminal bases	Yes	No

*see manual 26-857

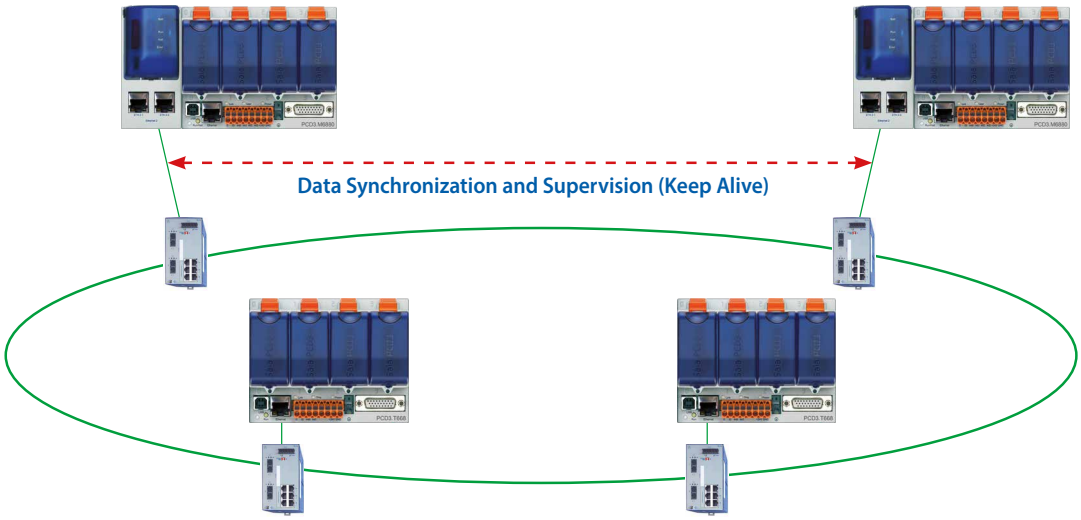
1

1.5.3 Switchover Criteria

Each of the Standby PCDs (CPU1) sends a „Keep Alive“ telegram to its partner for supervision.

The STANDBY PCD switches to ACTIVE when:

- No Keep Alive telegram has been received within the „Keep alive timeout“ period defined with the Redundant CPU‘s Device Configurator. The “Keep Alive Timeout” can be adjusted between 100…500 ms. By this the max. switchover latency is <100…500 ms.
- The ACTIVE PCD‘s state is not RUN or STOP (stops sending Keep Alive).
- A manual switchover command is executed. This is only possible if the Primary device does not have priority, the „Primary device has priority“ option must be „No“.



Data Synchronisation and Program Cycle:

The used PCD medias (R, F, T/C, DB/TX) in the redundant CPU1 are cyclically synchronized between the active and the standby PCD. The synchronization time for all PCD media is normally less than 200 ms. This time is reduced accordingly if only a part of the PCD media is used. The total program cycle time is calculated as follows:

Total cycle time = program execution time + data synchronization time

The max. value for a large application can be calculated as follows: 100 ms + 200 ms = 300 ms max.

For smaller applications where less PCD media are used the cycle time is reduced correspondingly.

1.6 PCD3.T668 Standby RIO

1.6.1 Architecture of the PCD3.T668

The PCD3.T668 remote I/Os are exclusively for use with the PCD3.M6880 Standby Controllers. With the exception of the redundancy function, they support the same properties/functions as the PCD3.T666 remote I/O station. The PCD. T665 and PCD3.T666 standard remote I/Os cannot be used with Standby Controllers.

- Can be used as a simple local I/O station or an intelligent programmable I/O station
- Can be programmed with the PG5. Important or timecritical tasks can be processed directly in the RIO
- The RIO‘s user programs are managed centrally by the Smart RIO Manager (PCD) and downloaded to the RIOs automatically
- Data exchange uses the efficient Ether-S-IO protocol. Simple configuration with the RIO Network Configurator
- Cross-communication with other PCD systems using Ether-S-Bus (FBoxes)
- Intelligent communication modules (e.g. M-Bus, DALI) are supported
- Other communication protocols (e.g. Modbus) via Ethernet TCP/IP and also by the onboard RS-485 interface
- Integrated Web Server

1.6.2 Technical Data

Property		PCD3.T668
Number of inputs/outputs		64 in base unit, extensible to 256
I/O-module slots		4 in base unit, extensible to 16
I/O-modules supported		PCD3.Exxx, PCD3.Axxx, PCD3.Bxxx, PCD3.Wxxx
Max. number of RIO stations		128
Protocol for data transfer		Ether-S-IO
Ethernet connection		10/100 Mbit/s, full-duplex, auto-sensing, auto-crossing
Default IP configuration		IP address: 192.168.10.100 Subnet mask: 255.255.255.0 Default gateway: 0.0.0.0
USB port for configuration and diagnostics		yes
Program memory		128 kByte
Web server for configuration and diagnostics		yes
Web server for user pages		yes
On-Board file system for web pages and data		512 kByte
BACnet® or LonWorks®		no
On-Board interrupt inputs		2
On-Board RS-485 interface		yes
Special modules	for I/O-slot 0 only*	PCD3.F1xx
	for I/O-slots 0…3* (up to 4 modules)	PCD3.H1xx counter PCD3.F26x DALI PCD3.F27x M-Bus
S-Web alarming/trending		no
Watchdog		no
Real-time clock		no
Software clock (not battery-powered)		yes, synchronized by the Manager
Battery		no

*see manual 26-857

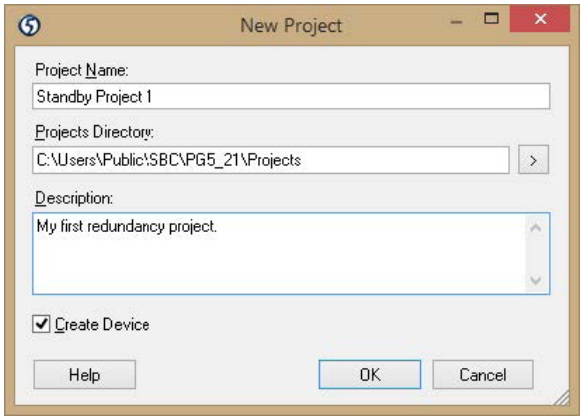
2 Configuration and Programming

2.1 Creating a Standby System Project in PG5

The Saia PG5® has many features to help you create a standby system project. It will create the three standby devices for you (primary, secondary and redundant), and creates the symbol files for defining the data to be exchanged between the CPUs.

1. Create a new Project

Use Project Manager's "Project / New Project" command to create a new project. Check the "Create Device" option.

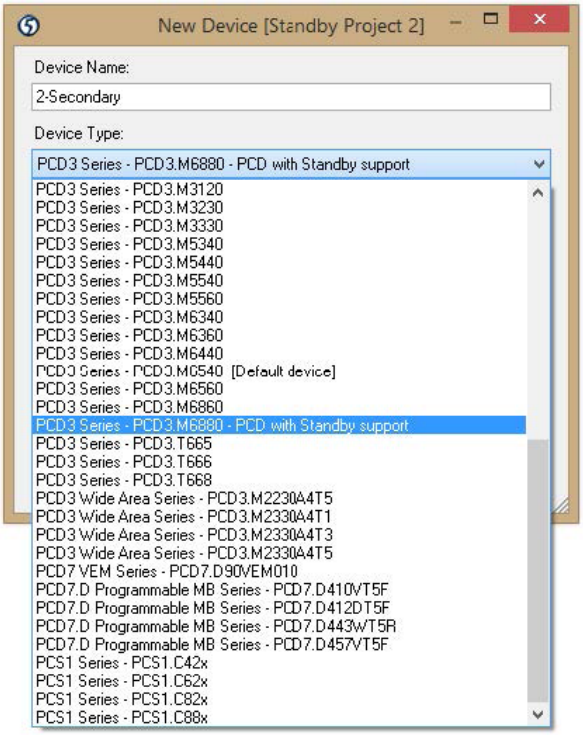


A single PG5 project can contain one standby system (one primary and one secondary device). If you have more than one standby system on the same network, then you must create a new PG5 project for each standby system.

If you are using Ether-S-Bus to communicate between the PCDs of two or more standby systems on the same network, then you can use the Export and Import features of the "TCP/IP Settings Table" window to transfer the settings between the projects.

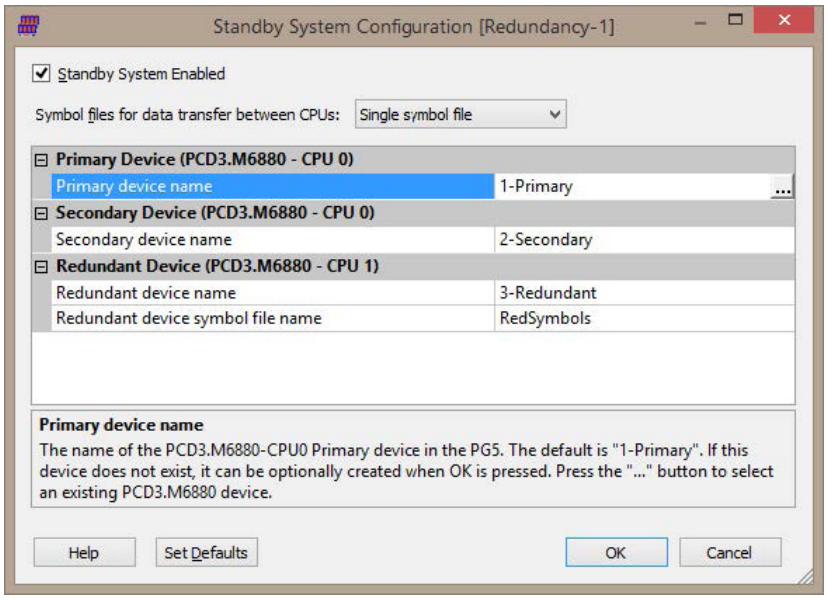
2. Add the Primary standby device

The “New Device” dialog box will be displayed next. Select the device type “PCD3.M6880 - CPU with Standby support”. This is the “primary” device, so it is assigned the name “1-Primary” by default, but you can rename it if you like. A standby project contains 3 devices, by default these are named “1-Primary”, “2-Secondary” and “3-Redundant”. The “1-”, “2-” and “3-” prefixes are used only to define the order of the devices in the Project Tree. The devices can also be renamed later.



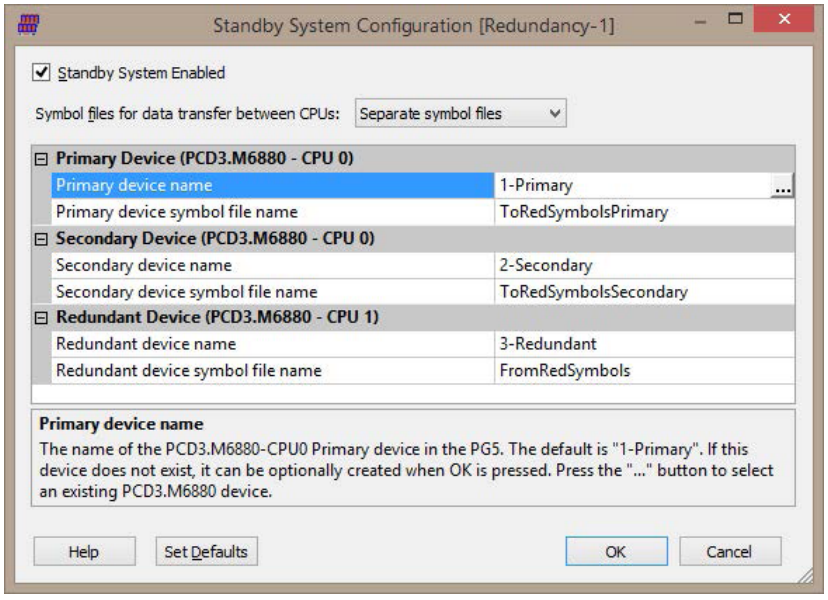
3. Configure the standby system

When OK is pressed to create the device, Project Manager creates the primary device, then displays the Standby System Configuration dialog box with the default configuration:



By default, a single symbol file is used to define the media for transfer between CPU0 and CPU1 on the Primary and Secondary devices. These media are also transferred between the devices, see Media Exchange.

According to the application, you may want to use separate “to” and “from” symbol files for the data transfer between CPU0 and CPU1. To do this, select the “Separate symbol files” option. It will then use three symbol files, ToRedSymbolsPrimary.sy5, ToRedSymbolsSecondary.sy5 and FromRedSymbols.sy5.



You can leave the settings as their defaults, or change the device names or symbol file names. The configuration can always be changed later, so using the

defaults is good for starting the project.

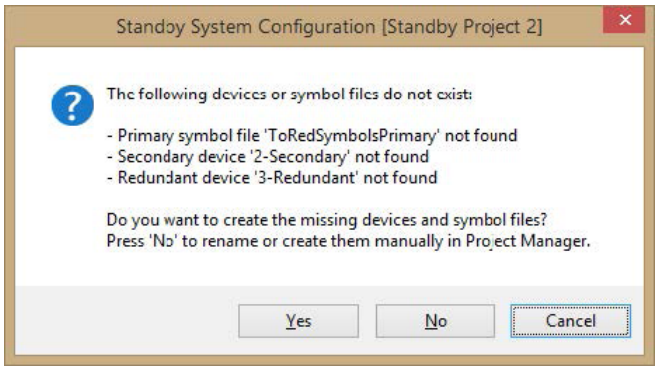
See the Standby System Configuration dialog box for more details.

Renaming or deleting devices and files

If you use Project Manager to rename or delete one of the standby devices or symbol files, then you must open the ‘Standby Configuration’ dialog box and modify the names there too, Project Manager will not do this automatically. You will see a configuration error or warning message if a device or symbol file is missing.

4. Create the Secondary and Redundant devices

Press OK. The Project Manager will see that the required devices and symbol files do not exist, and will ask if you want to create them:



Press ‘Yes’ to create the default devices and files. Press ‘No’ if you want to create these files yourself, or restore, copy or import them from another project. If there are any files missing, messages will be shown in Project Manager’s “Messages” window, see Configuration Error and Warning Messages.

This message box is also shown if you later rename a device or file, it will always ask if you want to create a missing device or file.

You can always rename or create the devices and files later if you need to.

5. Configure and download the Device Configuration to the Primary and Secondary devices

Open the Device Configurator for the 1-Primary and 2-Secondary devices (CPU0). Set the S-Bus station number, usually Primary=0 , Secondary=1. Enter the IP addresses of each device in the “Onboard Communications - Ethernet” slot.

Open the Device Configurator from the 3-Redundant device (CPU1), go to the “Onboard Communications - Ethernet” slot, and enter the same the IP Addresses for the Primary and Secondary devices, see Redundant Device Configuration.

The new configuration must now be downloaded into the primary and secondary devices, see Downloading the Configuration.

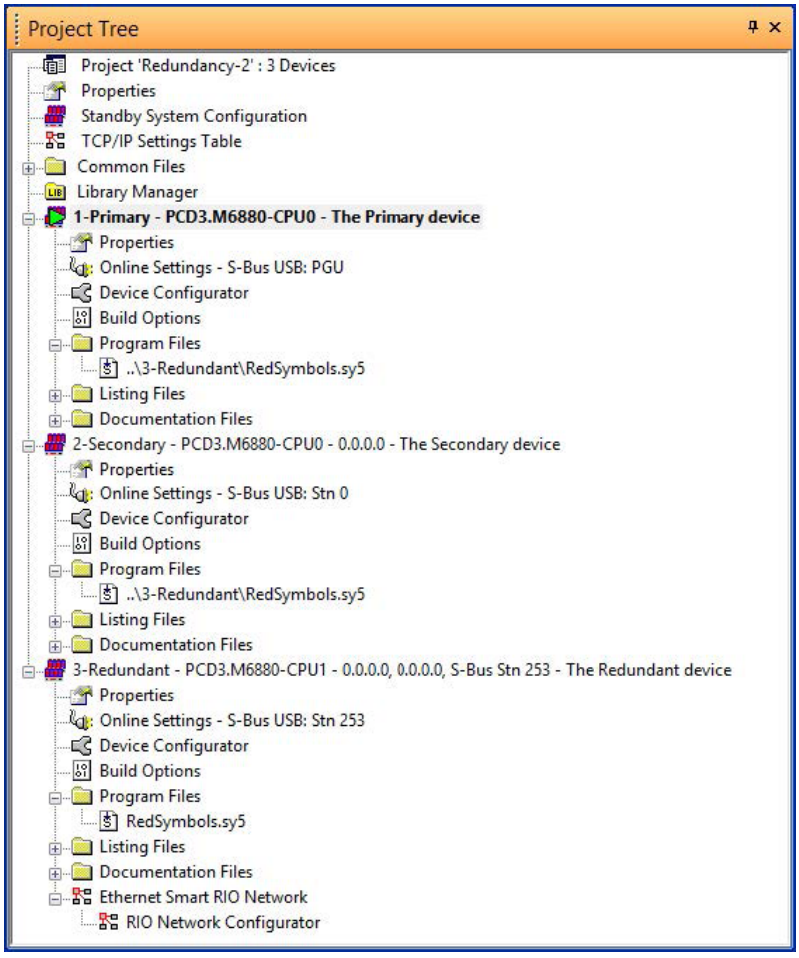
2.2 PG5 Project Manager View

The three devices and their symbol files are shown in the PG5 Project Manager's "Project Tree". If the Redundant device connects to programmed RIOs, you will also see the RIOs in the Project Tree.

Each device defines its own symbol file(s), see Media Exchange.

The Primary and Secondary devices reference the symbol files of the Redundant device. The Redundant device references the symbol files of the Primary and Secondary devices. The symbol file references are shown using a relative path name "..\directory_name\file_name.sy5", where ".." means the parent directory.

Here's the view with the "Single symbol file" option selected:



If the "Separate symbol files" option is chosen, then three symbol files are shown.

2.3 Process Data Synchronization

All media used by the CPU1 Redundant program are synchronized between CPU1 of the Active and the CPU1 of the Standby device.

The data can be synchronized cyclically (e.g. at the end of every cycle, all the data are transferred and the next cycle beings only after all data has been sent), or it can be sent asynchronously without slowing the program cycles. If sent asynchronously, a minimum sync time can be specified, and the Standby CPU's data may be some cycles behind the Active CPU. See Data Synchronization and Switchover (Synchronous) and Data Synchronization and Switchover (Asynchronous) for timing details. The data synchronization behavior is defined in the Device Configurator for the Redundant CPU1, see Redundant Device Configuration.

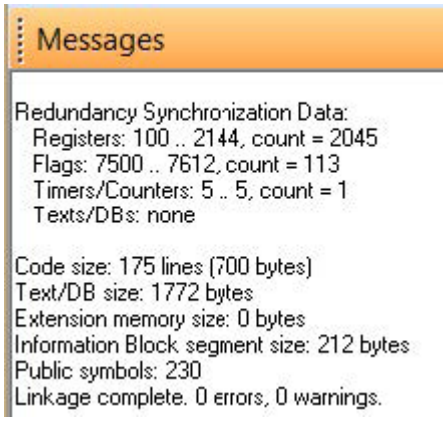
The PG5's "build" generates a table of all the used media addresses, and saves the first and last used addresses for each media type.

All media between the first used address and the last used address are transferred, even if the data is not used by the program, so try to keep the addresses consecutive.

If using dynamic addressing, keep the dynamic address ranges to a minimum the device's "Build Options" to assign the address ranges. Use the "Device / Advanced > Clean Files" command to reset the dynamic address allocator so the addresses are assigned consecutively and no addresses are unused.

If you use absolute addresses in the program, use addresses immediately before or after the dynamic address ranges so that absolute and dynamic addresses are consecutive. Synchronization data also includes the data in the "[From] RedSymbols.sy5" file, see Media Exchange, so the addresses in this file should also be consecutive.

The number and addresses of the media to be transferred are shown in Project Manager's "Messages" window when the Redundant device's program is built:



The worst-case data synchronization time for 16K Flags, 16K Registers, 1600 T/Cs, 128K RAM Text/DB (200KB data) is about 200ms. But usually the amount of used media will be much smaller, so the sync time will be faster.

Indexed Addressing

The media addresses of the synchronization data are taken from the addresses actually used by the program. If media are accessed using the Index Register, these addresses may not be taken into account if they are at the end of the address range. If the Index Register is used, always ensure that the indexed addresses are within the ranges shown in the ‘Redundancy Synchronization Data’ table. If not, define and reference an address at or beyond the end of the indexed range, to ensure the indexed media are also transferred.

2.4 Media Exchange

To communicate between CPU0 and CPU1 on each device, and between the programs on the Primary and Secondary devices, media values can be transferred cyclically between each CPU.

Special symbol files are used to define the media to be transferred between CPU0 and CPU1 of the Primary and Secondary devices and the media to be transferred between the CPU1s of the Primary and Secondary devices. Two options are available: “Single symbols file” and “Separate symbol files”, see the Standby System Configuration dialog box.

If the same program is to be used in CPU0 of both devices (Primary and Secondary), then the “Single symbols file” option must be used, so that the symbol names are always the same in both programs.

Single symbol file

One symbol file (default name “RedSymbols.sy5”) is used. When a value is written to one of the media addresses defined in this file, the value is transferred to the other CPUs in the ACTIVE and the STANDBY devices.

The procedure is as follows:

1. CPU0 of the ACTIVE device modifies the media values of symbols defined in “RedSymbols.sy5”, either by the user program, an S-Bus telegram, a Web page, etc.

2. CPU0 transfers the modified media values to CPU1 of the ACTIVE device at the end of the CPU1 cycle. (If CPU1’s program subsequently modifies the same value before the end of the cycle, then the new value will be used instead of the value modified by CPU0.)

3. All media values defined in ‘RedSymbols.sy5” are then cyclically copied back to CPU0.

4. For data synchronization between the Active and Standby devices, all media values are transferred from CPU1 of the ACTIVE device to CPU1 of the STANDBY device. This also transfers the media values that were modified by CPU0 of the ACTIVE device.

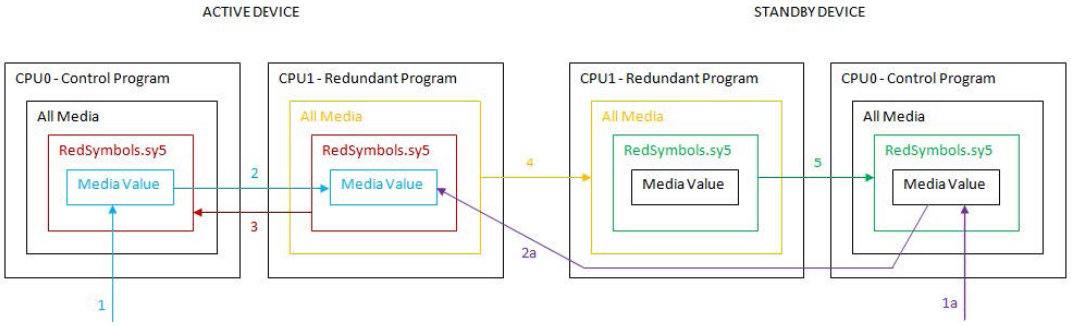
5. All the “RedSymbols.sy5” media are cyclically transferred from CPU1 to CPU0 of the STANDBY device. This also transfers the media values that were modified by CPU0 of the ACTIVE device.

For the STANDBY CPU0 the process is almost the same:

1a. CPU0 of the STANDBY device modifies the media values of symbols defined in “RedSymbols.sy5”.

2a. The modified media values are transferred through CPU1 to the corresponding media in CPU1 of the ACTIVE device.

3..5. The same as above. Note that the modified media values are updated in CPU1 of the STANDBY device only when the media are copied from CPU1 of the ACTIVE device in Step 4.



Separate symbol files

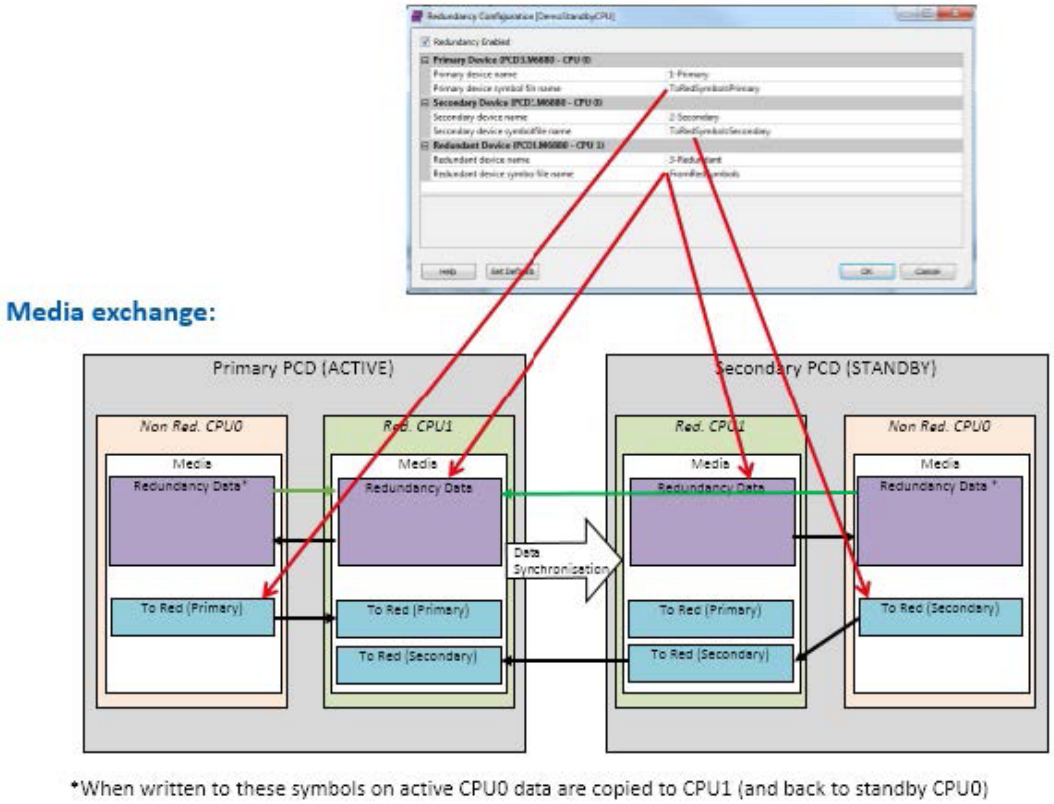
Three global symbol files (.sy5) are needed to define the data to be transferred between the CPUs.These are the default file names, which are created for you automatically when the project is created:

ToRedSymbolsPrimary.sy5	Defines data transferred to the Redundant CPU1 from the Primary CPU0
ToRedSymbolsSecondary.sy5	Defines data transferred to the Redundant CPU1 from the Secondary CPU0
FromRedSymbols.sy5	Defines data transferred to both the Primary and the Secondary CPU1 from the Redundant CPU1

The FromRedSymbols (from Redundant) are also transferred from the Active to the Standby device. This allows processes in the Active CPU0 to be monitored by the program in CPU0 of the Standby device. This is useful for SCADA control and monitoring where each device is connected to its own SCADA system.

These symbol files are referenced by each CPU via links in the Project Manager’s “Project Tree”. Synchronous transfer of the media values between CPU0 and CPU1 is done automatically at the start/end of each cycle.

2



*When written to these symbols on active CPU0 data are copied to CPU1 (and back to standby CPU0)

NOTES

Indexed Addressing

The media addresses of the synchronization data are taken from the addresses actually used by the program. If media are accessed using the Index Register, these addresses may not be taken into account if they are at the end of the address range. If the Index Register is used, always ensure that the indexed addresses are within the ranges shown in the 'Redundancy Synchronization Data' table. If not, define and reference an address at or beyond the end of the indexed range, to ensure the indexed media are also transferred.

Absolute addresses are recommended

It is recommended to use sequential absolute addresses (e.g. R 1000, R 1001, R 1002 ...) for media exchange symbols instead of dynamic addresses, so the addresses will never change when a build is done. The data to be transferred are treated as an array of values between the first and the last address. Dynamic addresses can be used, but this is less efficient because the addresses may not be consecutive, so single addresses instead of an array are transferred, and there may be alignment problems which require a "Device / Advanced > Clean Files - Symbol Information Files" operation. If there is an alignment problem, you will receive the "Redundancy media exchange dynamic addresses are incompatible" error (see below).

Error 939: Standby system media exchange dynamic addresses are incompatible (ToRed/FromRed)

You may see this build error if you have used dynamic addresses for the media exchange data in the global symbol files. It is caused by the optimization of addresses to be transferred into arrays, to reduce the number of transfers. With dynamic addresses, addresses in one CPU may be consecutive, and in the other they may be separate. This means that the optimized arrays will not be the same, and the data cannot be transferred. The solution is to do the "Device / Advanced > Clean Files - Symbol Information Files" operation, or to edit the symbols to use consecutive absolute addresses.

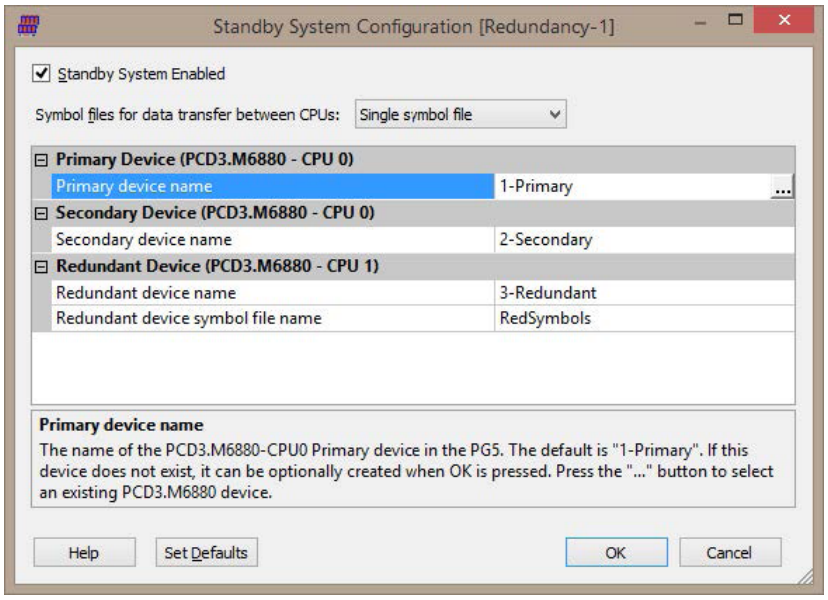
2.5 Configuration

2.5.1 Standby System Configuration dialog box

If the project contains a PCD3.M6880 redundancy device, then the “Standby System Configuration” branch is shown at the top of the Project Tree. Double-click on this to open the “Standby System Configuration” dialog box. This dialog box is also shown when you first add a PCD3.M6880 Standby Device to your project.

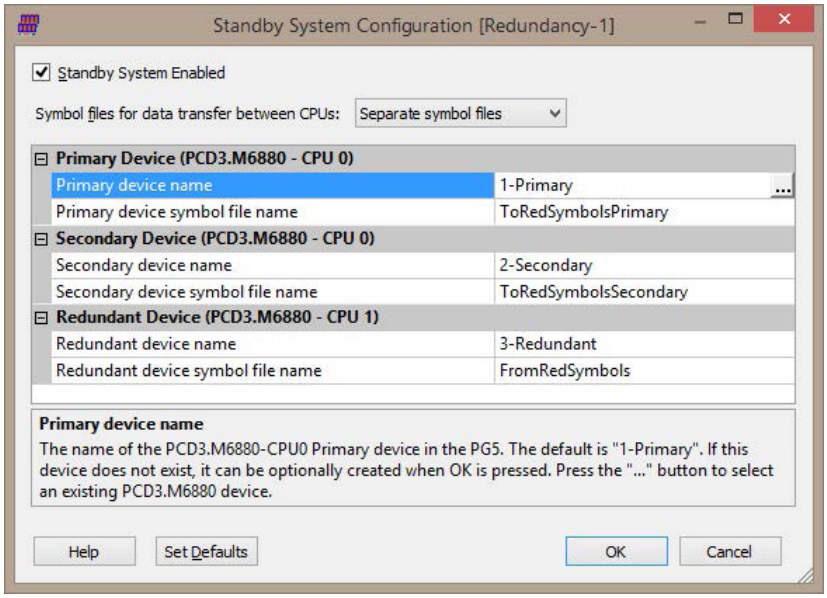
This is where the device and symbol file names for the Standby project are defined.

With the “Single symbol file” option selected:



2

With the “Separate symbol files” option selected:



Standby System Enabled checkbox

Enables or disables the standby feature for the project.

Symbol files for data transfer between CPUs

Depending on your application, you can use either one or three symbol files for defining the media to be transferred between CPU0 and CPU1 of each device, and between the two devices. See Media Exchange for more details.

Primary Device (CPU0)

This is CPU0 of the first PCD3.M6880.

Primary device name

The name of the PCD3.M6880 Primary device. The default is “1-Primary”. Press the “...” browse button to select an existing PCD3.M6880 device.

Primary device symbol file name

For “Separate symbol files” only. The name of the global symbol file that defines the media transferred between CPU0 and CPU1 of the Primary device. This is shown as “To Red (Secondary)” in the picture above. The default name is”ToRedSymbolsSecondary”.

Secondary Device (CPU0)

This is CPU0 of the second PCD3.M6880.

Secondary device name

The name of the PCD3.M6880 Secondary device in the PG5. The default is “2-Secondary”. Press the “...” browse button to select an existing PCD3.M6880 device.

Secondary device symbol file name

For “Separate symbol files” only. The name of the global symbol file that defines the media to be transferred between CPU0 and CPU1 of the Secondary device. The default name is “ToRedSymbolsSecondary”.

Redundant Device (CPU1)

This represents CPU1 in both the primary and the secondary PCD3.M6880 devices. It runs the same program in both devices. This program is the process which has redundancy.

Redundant device name

The name of the CPU1 device in the PG5. This device’s program is loaded into CPU1 of the primary and the secondary devices.

Redundant device symbol file name

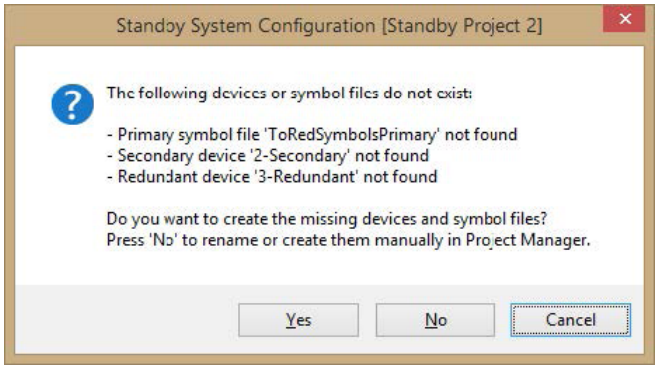
The name of the global symbol file that defines the media to be transferred between CPU1 and CPU0 of the primary and the secondary devices. For “Single symbol files”, the default file name is “RedSymbols”. For “Separate symbol files”, the default file name is “FromRedSymbols.sy5”.

Set Defaults button

Press this to set the default device and file names. The same defaults are used when a new redundancy system is first created by adding a PCD3.M6880 device to the project.

OK button

When OK is pressed, Project Manager checks for the presence of the Primary, Secondary and Redundant devices and symbol files. If any devices or files are missing, Project Manager will ask if you want to create them:



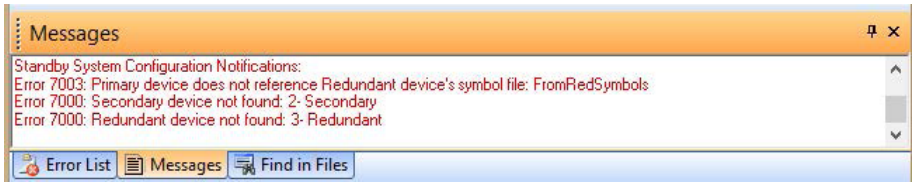
Press ‘Yes’ to automatically create the indicated devices and files. If the devices or files already exist but have different names, you can press ‘No’ and then rename them manually using the device’s or file’s ‘Properties’. Project Manager will not rename them automatically.

Press ‘No’ if you want to create these devices or files manually, restore them from a backup, or import them an existing project.

Validation

Project Manager will validate the standby system configuration when the dialog box is closed, when certain changes are made, and before a build is done. If any errors are found they are reported in the Messages window, and they should be corrected before continuing.

For example:

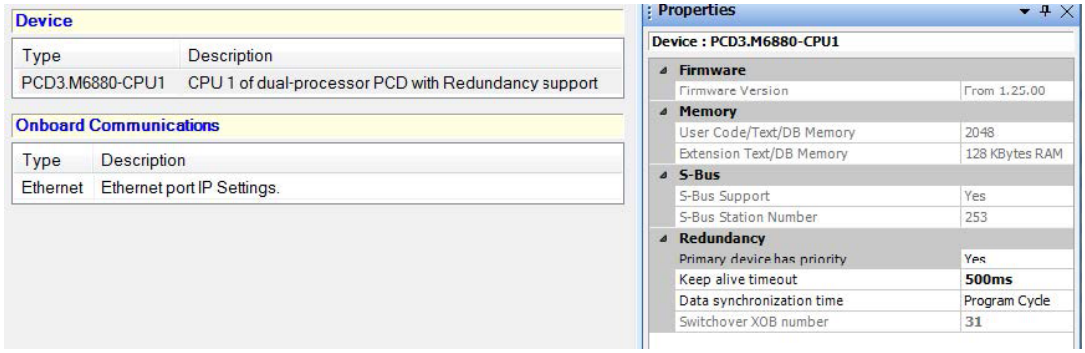


See Standby System Configuration Error Messages for a description of these errors.

2.5.2 Redundant Device Configuration

The Redundant device is configured using the Device Configurator. The configuration includes the IP addresses of the Redundant CPU which are used by the Primary and Secondary devices.

There is a “Redundancy” section in the PCD3.M6860-CPU1 properties:



Primary device has priority

If ‘Yes’, then the primary device will be the active device whenever possible, and manual switchover is disabled. If ‘No’, then either device can be the active device, and manual switchover is enabled.

Keep alive timeout (ms)

If the standby CPU1 device has not received any communications from the active CPU1 device for this number of milliseconds, then it will assume the active device is down or communications has failed, and will take over control by becoming the active device. Selectable timeouts are 100 ms, 250 ms and 500 ms.

Data Synchronization Time

The speed at which data is synchronized between the two CPU1 devices. The time needed to transfer synchronization data will often be longer then the cycle

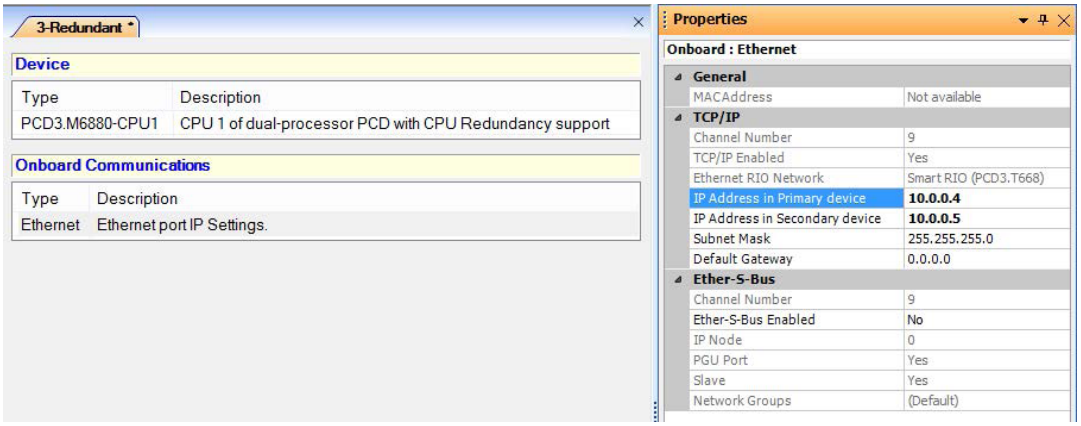
time, depending on the amount of data and the complexity of the program. This means that the standby CPU's data may be some cycles behind the active CPU. "Program Cycle" will transfer the data on every program cycle and wait until the data is sent. 100ms..1000ms specifies the minimum times between data transfers. If the cycle time is longer than this time, then the cycle time is used instead. Actual data synchronization times can be seen by using the web page of the active CPU0.

Switchover XOB number

Fixed as XOB 31. If programmed, this XOB is executed on CPU1 when the Standby PCD takes over control of the system and becomes the Active PCD.

Ethernet Port IP Settings

The Redundant CPU1's Ethernet IP ports are also configured from here.



Primary CPU1 IP Address

The IP address of CPU1 in the Primary device. This IP address is used to transfer the synchronization data between CPU1 in the Secondary device, and CPU1 in the Primary device.

Secondary CPU1 IP Address

The IP address of CPU1 in the Secondary device. This IP address is used to transfer the synchronization data between CPU1 in the Primary device, and CPU1 in the Secondary device.

Subnet mask and Default gateway

CPU1's subnet mask and default gateway. The Ethernet settings for CPU0 are done from the Device Configurator of the primary and secondary devices.

2.5.3 Creating a Default RIO Configuration

A Standby system uses the RIO type PCD3.T668. This type is automatically used by the RIO Network Configurator if the manager device is a PCD3.M6880. These RIOs can be programmed to reduce the amount of processing done by the manager device (CPU1). If programmed, they will also run autonomously if communications with the active manager device has failed.

You can create a default PCD3.T668 configuration using the RIO Network Configurator (Tools / Default RIO Configuration), and re-use this configuration as a template for each new RIO. See Project Manager's help "How to create a RIO network" for details.

2.5.4 Error and Warning Messages

The following errors may be reported in Project Manager's "Messages" window when the standby system configuration is modified, and before a build or download is done. Errors should be corrected before continuing.

Error Messages

Error 7000: Primary device not found: <device_name>
Error 7000: Secondary device not found: <device_name>

The device does not exist in the project. If the name is incorrect, rename it or change its name in the Redundancy Configuration. If the device is missing, create a device with the correct name and the type PCD3.M6880. To create the device automatically, open the Redundancy Configuration dialog box, press OK, and answer 'Yes' to the "Do you want to create..." message box.

Error 7000: Redundant device not found: <device_name>

The device does not exist in the project. You cannot create a device with the type PCD3.M6880-CPU1, so open the Redundancy Configuration dialog box, press OK, and answer 'Yes' to the "Do you want to create..." message box.

Error 7001: Primary device is not a PCD3.M6880
Error 7001: Secondary device is not a PCD3.M6880
Error 7001: Redundant device is not a PCD3.M6880-CPU1

The device whose name is defined in the Redundancy Configuration has the wrong PCD type. If it's the primary or secondary device, then you can use the Device Configurator to select the correct type or create a new device with the right type and change the device name in the Redundancy Configuration. If it's a PCD3.M6880-CPU1 device, open the Redundancy Configuration, change the redundant device name and press OK to create the new device.

Error 7002: Primary device symbol file not defined: <file_path>
Error 7002: Secondary device symbol file not defined: <file_path>
Error 7002: Redundant device symbol file not defined: <file_path>

A global symbol file with the specified name does not exist. Create it manually or use the Redundancy Configuration dialog to create it for you.

Error 7003: Primary device does not reference Redundant device's symbol

file: <file_name>

Error 7003: Secondary device does not reference Redundant device's symbol file: <file_path>

Error 7004: Redundant device does not reference Primary device's symbol file: <file_path>

Error 7004: Redundant device does not reference Secondary device's symbol file: <file_path>

The symbol file references should be as shown in the Project Tree picture above. The primary and secondary devices should reference the redundant device's symbols (e.g. FromRedSymbols.sy5), the redundant device should reference the primary and secondary devices symbol files (ToRedSymbolsPrimary.sy5 and ToRedSymbolsSecondary.sy5)

Error 7005: Invalid or missing Primary IP address for Redundant device

Error 7005: Invalid or missing Secondary IP address for Redundant device

Error 7005: Invalid or missing Subnet Mask for Redundant device

Error 7005: Invalid or missing Default Gateway for Redundant device

The IP address is not valid. open the Redundancy Configuration dialog box and correct it.

Error 7006: Primary and Secondary IP addresses for Redundant device are the same

Different IP addresses must be used. Open the Redundancy Configuration dialog box and correct it.

Error 7007: Invalid 'Online Settings' for Redundant device (needs Station=253, Auto Station=No): <device_name>

This is possible because the 'Online Settings' does not know what the device type is, and cannot stop the user changing the settings. Open the 'Online Settings' for the redundant device and correct them.

Error 7008: Primary device symbol file has wrong extension: <file_name>

Error 7008: Secondary device symbol file has wrong extension: <file_name>

Error 7008: Redundant device symbol file has wrong extension: <file_name>

A file with the correct name exists, but the existing file does not have the '.sy5' file extension. Manually create a new Global Symbol File (.sy5) and change the file name in the Redundancy Configuration, or use OK to create a new file for you.

Error 7009: Primary device symbol file is not a 'Symbols file', see Properties: <file_name>

Error 7009: Secondary device symbol file is not a 'Symbols file', see Properties: <file_name>

Error 7009: Redundant device symbol file is not a 'Symbols file', see Properties: <file_name>

The file exists, but it has not been flagged as being a 'Symbols file". Open the file's 'Properties' and check the 'Symbols file' checkbox.

Error 7011: Redundant device's S-Bus station number must be 253

The S-Bus station number for the Redundant device (CPU1) must be 253. This

is hard-wired in the firmware. Communications will not work unless the station number is 253.

Error 7012: Primary and Secondary devices have the same S-Bus station number

The Primary and Secondary devices must have different S-Bus station numbers. You will see this error if a new project has been created and the station numbers have not yet been assigned (both are 0).

Warning Messages

Warning 7101: Invalid 'Channel Type' in Online Settings, use S-Bus, S-Bus USB or SOCKET

Only these channels can be used to communicate with a standby controller.

Warning 7102: Invalid 'S-Bus Station Number' in Online Settings (is n, should be n)

The S-Bus station number on the Online Settings does not match the S-Bus station number on the device configuration.

Warning 7103: Invalid 'IP Address' in Online Settings (is xxx.xxx.xxx.xxx, should be xxx.xxx.xxx.xxx)

The IP address in the Online Settings does not match the IP address in the device configuration.

Warning 7104: Invalid Online Settings, 'PGU Mode' should be 'No'

PGU mode cannot be used to communicate with a standby controller. It may cause connection to the wrong device or CPU. Open the Online Settings and ensure 'PGU' mode is 'No'.

Warning 7105: Invalid Online Settings, 'Auto Station' should be 'No'

Automatic station number detection mode cannot be used to communicate with a standby controller. It may cause connection to the wrong device or CPU. Open the Online Settings and ensure 'Auto Station' is 'No'.

2.5.5 Downloading the Device Configuration

The device configuration must first be downloaded into the Primary and Secondary devices using a local “S-Bus USB” channel, to configure the S-Bus station numbers and IP addresses. After this step you can use a n “S-Bus TCP/IP” Ethernet channel.

Downloading the configuration to CPU0 of the Primary or Secondary device also updates the configuration of CPU1 at the same time. It is not possible to update the configuration of only CPU1, the Download Configuration command is disabled for CPU1.

Configure the Online Settings

Before you can download the new configurations to the Primary, Secondary and Redundant devices, you must ensure that the Online Settings are correct. First use an S-Bus USB channel to download the configuration. Connect a single device with the USB cable, select an “S-Bus USB” channel, PGU mode = ‘Yes’. Open the device Configurator, and press the ‘Download Configuration’ button. This updates the configuration for both CPUs (0 and 1) in the device. Repeat this for the primary and secondary devices.

Once the configuration has been downloaded, open the Online Settings and set PGU mode = ‘No’, and Auto Station = ‘No’.

Now you can either use two local S-Bus USB connections or an S-Bus TCP/IP connection for communications with the two devices.

Using two USB connections

Two USB connections can be used to connect locally to both devices at the same time, but normally you would use an S-Bus TCP/IP channel for this.

Connect the two USB cables to the Primary and Secondary CPU0. Open the Online Settings for the Primary and Secondary each devices, select the S-Bus USB channel, ensure that PGU and Auto Station are both ‘No’. Enter the correct S-Bus station number. To select the right USB channel, press the ‘Refresh USB list’ button, then open the ‘USB Serial Number’ list and choose the correct device. The USB serial number is created from the device type and the device’s PCD serial number, e.g. PCD3.M6880 8A56C170000.

If using Ethernet, enter the IP addresses you used in the device configuration for the 1-Primary and 2-Secondary devices.

Test you connection by opening the Online Debugger (S-Bug) for each device.

You can also open the Download Program to Standby System dialog box to see the status of each device.

2.6 Programming

The programs “1-Primary” and “2-Secondary” are for CPU0 of the dual-processor PCD3.M6880. These programs are often the same, but they can be different in each device. Unlike the Redundant device, the programs in these devices can do communications and direct I/O accesses. See CPU0 Control Program.

To use the same program in the Primary and Secondary devices, first develop and test the Primary program, then use ‘Add Files’ to add all the program files to the Secondary device. Alternatively you can use the “Common Files” branch to hold the program files for both devices. You must use two devices because the device configuration will be different (S-Bus station, Ethernet address, serial number, MAC address, Online Settings etc). See Using the same program in the Primary and Secondary devices.

“3-Redundant” is the program for CPU1 in both the Primary and the Secondary devices. CPU1’s program must interface to the outside world using only the process image, so that execution of the two Redundant CPU’s programs can be synchronized. CPU1’s program cannot do any communications or any local I/O accesses - asynchronous operations are not allowed. I/O is done only via the Smart Ethernet RIOs, or by media exchange via dual-port RAM, because these both use the process image. See CPU1 Redundant Program.

2.6.1 CPU0 Control Program

The CPU0 program is a non-redundant program that always runs on both PCDs. There are no restrictions on what this program can do. It can access local I/Os, use asynchronous communications and connect to a SCADA system. The execution of this program is not synchronized with the Standby device For data exchange between CPU0 and CPU1 see Media Exchange.

The CPU0 control program can be the same in both devices, or can be different, depending on the application. See Using the same program in the Primary and Secondary devices.

2.6.1.1 Using the same program in the Primary and Secondary devices

In some cases, the control programs in both the Primary and Secondary devices will be the same, but with different IP addresses, S-Bus station numbers, and other configuration items. Or maybe the devices will share some of the same program files, but not all.

Program files can be shared by the Primary and Secondary devices either by placing the files in the “Common Files” branch and referencing these files from each device, or by using “Add Files” to add references to the program files of other devices.

The “Single symbol file” option for CPU0 <=> CPU1 data transfer should be used

In a Standby system, symbols which are transferred between CPU0 and CPU1 are defined in special symbol files. From the “Standby System Configuration” window, you can selected “Separate symbol files” (e.g. ToRedSymbolsPrimary.sy5, ToRedSymbolsSecondary.sy5 and FromRedSymbols.sy5), or a “Single symbol file” (e.g. RedSymbols.sy5) in which to define these symbols. If you want the same

program in both the Primary and the Secondary devices, then the symbol names must also be the same. This means that you cannot use the “Separate symbol files” option because there will be “multi-defined symbol” errors in CPU1 because the same symbol names must be used in ToRedSymbolsPrimary.sy5 and ToRedSymbolsSecondary.sy5. However, if you use “Single symbol file”, then the same symbols are used automatically.

If you really want to use “Separate symbol files”, you will need to use different symbol names in ToRedSymbolsPrimary.sy5 and ToRedSymbolsSecondary.sy5, probably just different group names, and then create alias symbol names from them, which are the same in both devices.

Using the “Common Files” branch

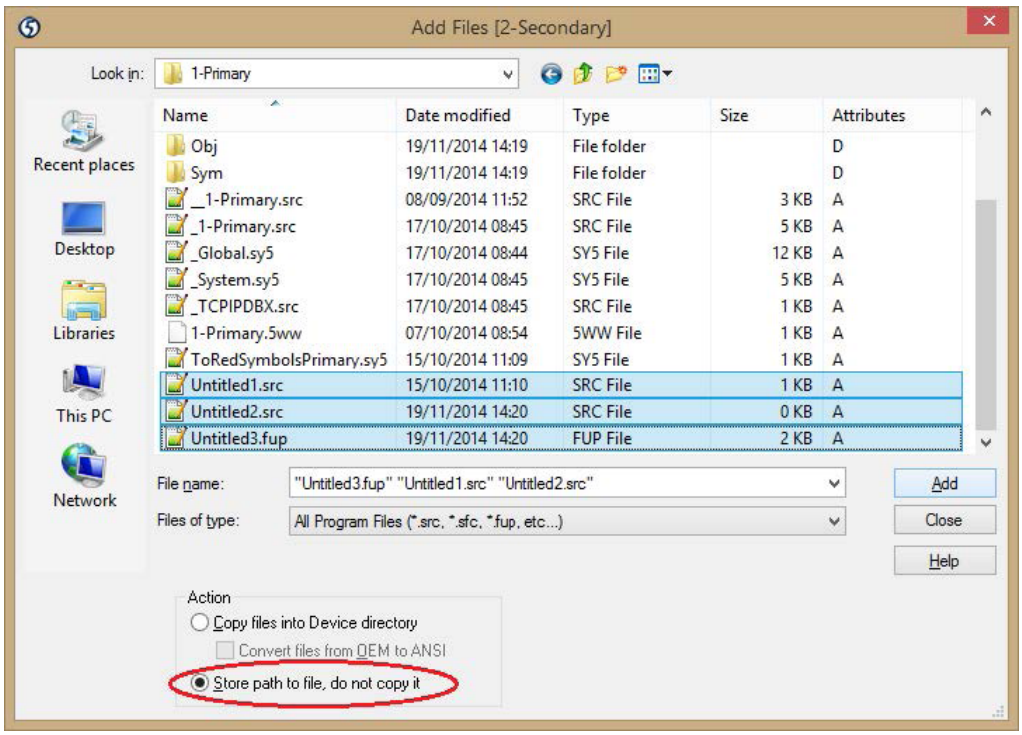
In Project Manager, create the files, e.g. the IL or Fupla program files, in the Common Files branch instead of the device branch. Then drag-and-drop each file onto the Primary or Secondary device’s “Program Files” branch. This creates a reference to the file in the Common Files directory. Common Files are located in the parent directory of the devices, so the path is indicated by “..\” in the file name. See the picture below. When you edit a Common File, it changes the code for both devices, so both devices will need a rebuild and download.

Even though the programs are the same in both devices, there may be other differences, such as different IP addresses, SCADA configurations etc. This can be managed by using differen “Gloabl Symbol Files”, in each device. These can contain Public symbols with the same names, but different values.

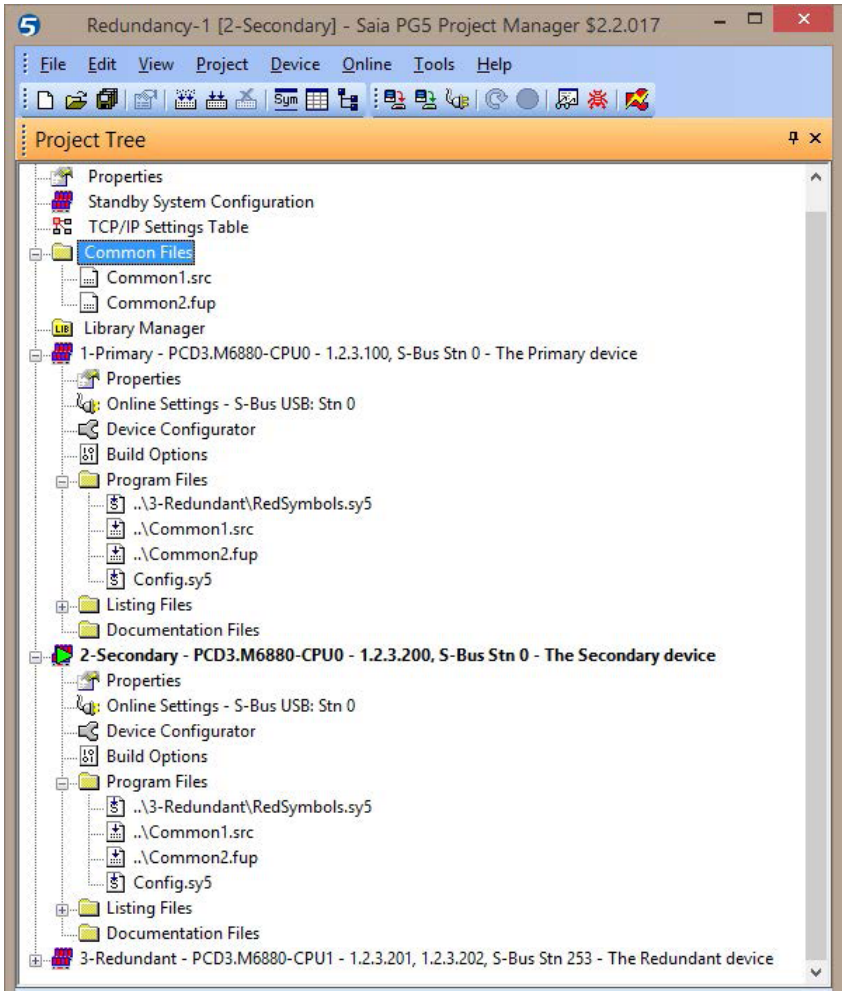
Referencing the program files of another device

Alternatively, you can create the file in the Program Files branch of the Primary device, and reference them from the Secondary device (or vice-versa).

Create the program files in the Primary device, then right-click on the Secondary device's Program Files branch and select the "Add Files.." command. To use the same program files, first select the Action " Store path to file, do not copy it" - see the picture below. If this is not selected, then "Add Files" will make new copies of the files, which is not what you want. From the "Add Files" dialog box, navigate to the Primary device's directory and select the program file (or select multiple program files by holding Shift or Ctrl key down before clicking on each file), then press "Add". This adds references to the selected files to the Secondary device's "Program Files" branch. You can also add individual files by selecting a single file and pressing the "Add" button. Press "Close" when all the required files are added.



Here is a project with the same programs for the Primary and Secondary devices, that uses two Common Files and global symbol files for symbols with different values in each device.



2.6.2 CPU1 Redundant Program

CPU1 of the Active device runs the Redundant program, and its process image must be synchronized with the process image of CPU1 in the Standby device. This means that no features that are asynchronous, use non-process data or communications stacks can be used on CPU1. Only the process image can be used for incoming and outgoing data. Because of this, there are some restrictions on CPU1's redundancy program.

- The CPU1 program runs only on the ACTIVE PCD and controls the remote IOs via PCD3.T668 RIOs.
- The redundant program is always the same on both PCDs.

Restrictions

- CPU1 cannot use any communications (S-Bus master, Modbus, BACnet, FTP etc.), except ESIO (for
- the RIO network) and data synchronization.
- CPU1 has no direct access to IO modules or Flash memory modules.
- Communications with a SCADA system or other external system can run only on CPU0. But commands and data can be transferred to CPU1 using Media Exchange.
- The CPUs can access each other's data only by using Media Exchange.
- CPU1 cannot run Graftec programs, they cannot be synchronized.
- FBox limitations: No communications FBoxes can be used in CPU1's program.
- A PG5 project can contain only one pair of redundancy devices, a new project is needed for each pair.

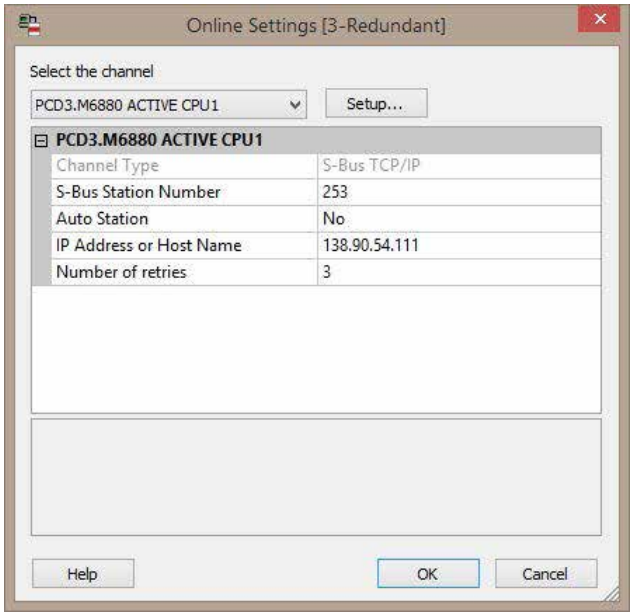
2.6.2.1 Going Online to Active CPU1

CPU1 is always S-Bus station number 253.

CPU1 has its own ‘Online Settings’ branch in the Project Tree, but CPU1 will usually be accessed via the Ethernet connection to CPU0 of the Primary or Secondary device. Therefore the same Online Settings of the Primary or Secondary CPU0 should be used, but with station number 253. Usually you will also want to go online with the ACTIVE CPU1 device, because only the ACTIVE device is executing the program.

If the channel named “PCD3.M6880 ACTIVE CPU1” is selected for CPU1, then when the Go Online command is done, the ACTIVE CPU1 is automatically connected using the Online Settings of the Primary or Secondary device. The ‘S-Bus Station Number’ is always set to 253, and the ‘IP Address or Host Name’ is filled in with the IP address of the Primary or the Secondary device, whichever is the ACTIVE device.

If another channel is used, then it will go online the actual Online Settings, so ensure that the IP address of the Primary or the Secondary device is used.



Tips

Connecting to CPU1 with S-Bug, S-Conf or the Watch Window

If you use the Online Debugger (S-Bug), the Online Configurator (S-Conf) or the Watch Window for CPU1, it will use the current Online Settings - it will not try to find the Active CPU1 even if the “PCD3.M6880 ACTIVE CPU1” channel is selected. To connect to the Active CPU1, you should first do a ‘Go Online’ for CPU1 with the “PCD3.M6880 ACTIVE CPU1” channel to select the correct Online Settings. You can then open S-Bug, S-Conf or the Watch Window.

To connect to CPU1 with S-Bug, you can select the Primary or Secondary device and open S-Bug. This connects to CPU0 first. You can then use S-Bug’s “cOnnect Cpu 1” command or “cOnnect Sbus-station 253” to connect to CPU1.

Forcing a switch-over using S-Bug

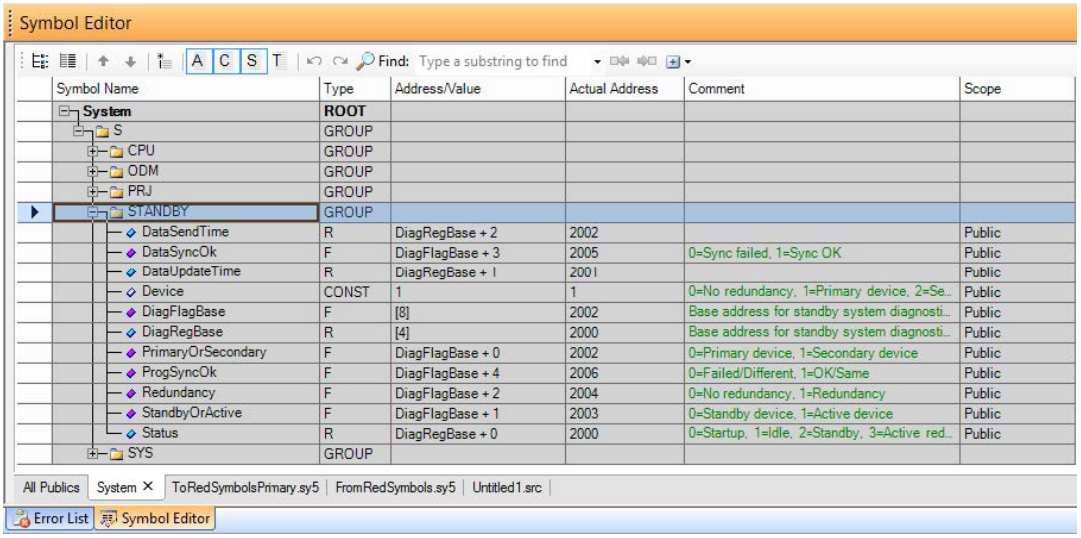
To test the switch-over, you can connect to the Active CPU1 with S-Bug, do a “Stop” command, then “Instruction HALT”. This Halts the Active CPU1 and the Standby CPU becomes Active. Note that the Standby CPU does not become Active when the Active CPU goes into Stop - this is because you would not be able to debug a program on the Active CPU1 without it causing a switch-over. (If the Active CPU1 ever fails, it will always go into HALT and cause a switch-over.)

2.6.3 Switchover XOB 31

XOB 31, if present, is executed on CPU1 when the Standby PCD takes over control of the system and becomes the Active PCD.

2.6.4 System Symbols

Each of the three redundancy devices has the following diagnostic System Symbols which can be used in the program. After the first build of the program, you can find the symbols on the “System” tab of the Symbol Editor. See the Diagnostics section for more details.



The T668 RIOs also have diagnostic symbols, see Ether-SIO Diagnostic Extension.

Defining your own absolute addresses for diagnostic media

By default, the diagnostic Flag and Register symbols have dynamic addresses (defined by the system), but you can assign your own fixed base addresses by defining these symbols in a global symbol file, for example called “BaseAddresses.src” :

Create the file as an Instruction List file (.src) using S-Edit. Open the Properties for the file, and check the define System symbols. It is important that the EQU statement is in the global symbol file, EXTNs cannot be used.

This works because the code that creates the symbols uses $IFDEF conditional directives, if you have defined the symbol S.STANDBY.DiagFlagBaseUsr and S.STANDBY.DiagRegBaseUsr then it will use these symbol values instead of dynamic addresses. See the listing file _device_name.lst:

```
                        ;Standby System Symbols
                        $GROUP S_STANDBY
                        $IFDEF .S.STANDBY.DiagFlagBaseUsr ;User-defined symbol for absolute address, see Help
F   1000                DiagFlagBase PEQU F .S.STANDBY.DiagFlagBaseUsr[8]
                        $ELSE
                      * DiagFlagBase PEQU F[8] ;Base address for standby system diagnostic flags
                      * $ENDIF
```

2.6.5 Download Programs to Standby System dialog box

A standby system project consists of two PCD3.M6880 devices (the Primary and the Secondary devices), and each of these devices contains two CPUs. This means that programs must be downloaded into four separate CPUs. To make this easier, all four programs can be downloaded in a single step by using the “Download Programs to Standby System” dialog box.

This dialog box is shown when the “Download Program” command is done for any of the standby system’s devices: the Primary, Secondary or the Redundant device.

The dialog box shows the four CPUs: the Primary CPUs 0 and 1, and the Secondary CPUs 0 and 1. The Primary and Secondary CPU 1 runs the redundant program, which is the same on both devices. The Primary and Secondary CPU 0 may run different programs.

For each CPU, the CPU Status (Run, Stop, Halt etc), the Build Status (Build OK, Build required), and the Program Status of the program already in the CPU (Program up-to-date, Program different etc) is shown. Error messages or invalid status texts are shown in red. The Program Info... button can be pressed to see more details of the Program Status.

Tip: The list of error messages and their meanings can be found at the end of this page.

The line below the Program Status shows any other errors that were detected, such as differences between the configuration in the PG5 and the actual device’s configuration. The Serial Number, S-Bus station numbers, IP addresses and program names are compared. An error message shown here indicates that the configuration is not up to date (use Download Configuration), or that you may be connected to the wrong PCD.

For advanced uses, there is a checkbox for each CPU so you can prevent the download if the program is already the same of if you want to test the behaviour when the PCDs contain different programs or program versions.

You will not be able to download a program if certain errors are present, such as “Build required”.

Program Info... buttons

Displays the “Program Information” dialog box, which shows the details of the PG5 program and the program within the connected PCD. Use this to see more details if the Program Status is not “up-todate”.

Download... buttons

Opens the standard “Download Program” dialog box for this CPU, from where you can download each program individually. This dialog box also provides some rarely used or advanced options. After the download, the status of all CPUs is refreshed in the same way as pressing the Refresh button.

Using the download method can cause the system to behave in unexpected ways, without any warnings or notifications, because many of the safety checks are not done.

Download with switchover checkbox

If checked, the programs are first downloaded into the Standby device’s CPU0 and CPU1. The Standby device then becomes the Active device, and the programs are downloaded into the previously Active device, which is now the Standby device. If not checked, the programs are downloaded into the Standby device, then to the Active device, and the Active device remains active after it is reset.

Clear Media (R F T C) checkbox

All Registers, Flags, Timers and Counters are reset to 0 before the downloaded program is run.

If using Media Mapping for Outputs on CPU 0, then checking this option will cause all Outputs to be reset.

Keep existing RAM DB and RAM TEXT data checkbox

The values in existing DBs and TEXTs in RAM memory are not changed by the download. New DBs or TEXTs are initialized with the data from the source file. If a DB or TEXT is increased in size, the existing data is unchanged, and the additional data is initialized with the data from the source file.

Refresh button

Refreshes all the status texts and messages. For example, if a CPU was not connected, and you have corrected the problem, pressing Refresh will connect and update the CPU status. Refreshing is not done automatically, the devices are not polled to see if their status has changed.

Start Download / Stop Download button

Validates and starts the download process. Programs are downloaded into all CPUs which have their ‘Download program’ checkboxes set. This button changes to ‘Stop Download’ during the download.

If you press ‘Stop Download’ to abort the download, the system will be left in an undefined state which depends on when the action was stopped.

Error Messages

These are the error messages and notifications that may be displayed by this dialog box, in alphabetical order. The message may also contain the device name and the expected and actual values.

Cannot download to Primary/Secondary CPU x, it is offline or using reduced protocol

If the PCD is password protected, enter the password when it is asked, before connecting to the PCD.
This error can also occur if a non-standard S-Bus serial connection is used, which does not support all telegrams.

Comms error: <details>

A communications error occurred, the operation was aborted. The system may be in an undefined state.

Configuration upload failed

You may be connected to a non-standard or unsupported PCD device.

Connected: Reduced protocol

To download programs, and full protocol connection is needed. This can occur if connected via a reduced protocol channel, or is Password Protection is active in the connected PCD.

Connect failed

Unable to connect to the CPU or PCD. Check it has power and is correctly connected to the PCD. This can also occur if the Online Settings are wrong, or if the PCD is incorrectly configured with a wrong station number or IP address etc. Try connecting in PGU mode using S-Bug or the Online Configurator, and display the current S-Bus configuration.

Do not use PGU mode for the downloader connection! Instead, put the correct S-Bus station number and IP Address in the Online Settings.

CPU x IP address not configured in PCD

The connected CPU does not contain an IP address. Open the Device Configurator, verify the IP address, then do a Download Configuration.

To download the configuration to the Redundancy CPU1, you must download the configuration to CPU 0, which also updates CPU 1.

Device not in Standby or Run

After a download and run operation, the device was found not be be in Standby or Run mode. This usually only occurs if a preceding fatal error has caused an abort, or if the PCD is in HALT. In in HALT, check the Halt Reason and look at the History messages using S-Bug's 'Display History'.

Device's Serial Number not defined in 'Downloader Options'

It is a good idea to check the destination PCD's serial number, to be extra-sure that you are downloading the program into the correct PCD, especially if it is one of many on a network. To do this, open the device's Downloader Options from the Device menu, check the option 'Warn if different Serial Number' and enter the device's serial number. The serial number can be seen on the Online Configurator's "Hardware Information" dialog, or with S-Bug's Display Sys-info command.

Different program names

If the device's Downloader Option "Warn if PCD contains program with different name" is checked, then you will see this error if the connect PCD contains a program with a different name. This may be because you are connected to the wrong PCD.

Failed to read program name

Caused by a communications error. The program name cannot be validated.

Invalid Primary CPU 1 IP address

Invalid Secondary CPU 1 IP address

The IP address of the Primary or Secondary CPU 1, used by each CPU 1 for the sync data transfer, is not the same as that configured in the Redundancy device's Device Configuration. This could be because of incorrect values on the Device Configurator, or because the connected CPU 1 contains a different configuration. Correct the IP address and/or download the new configuration.

To download the configuration to the Redundant CPU 1, you must download the configuration to CPU 0, which also updates CPU 1.

IP address not configured in Device Configurator

The configuration is incomplete, the IP address is blank or is 0.0.0.0. Open the Device Configurator, enter the correct address, then download it into the PCD.

IP address not configured in PCD

The device configuration has not be downloaded into the PCD.

Run failed

After downloading the programs to CPUs 0 and 1, the PCD is put into run with a 'run all CPUs' telegram. This error occurs if the command failed. This can occur if there is a communications error of if the PCD is in HALT.

S-Bus station number not configured in Device Configurator

S-Bus station number not configured in PCD

S-Bus support has not been correctly configured. Open the Device Configurator, correct the configuration, then download the new configuration into the PCD

Secondary CPU 0 and Secondary CPU 1 are the same device, check the Online Settings

If the Online Settings are incorrect, it may be possible to physically connect both channels to the same PCD. For example, if PGU mode or Auto Station is not ‘No’. Normally there will be other errors if the online settings are not valid.

Warning: Both devices are in STANDBY mode

Warning: Neither device is in STANDBY mode

Warning: Both devices are in ACTIVE mode (broken sync connection?)

Warning: Neither device is in ACTIVE mode

The redundancy control has failed. It could be caused by:

- Different firmware in each device (CPU0 and/or CPU1)

- Different programs in CPU1

- A failed data synchronization connection between the Primary CPU 1 and the Secondary CPU 1, via the Ethernet connector ETH 2.2. Ensure it is correctly wired and check the IP configuration is correct and has been downloaded. Check for another error message, such as “Invalid Primary CPU 1 IP address”.

Wrong PCD type

The connected PCD is not a PCD3.M6880 or PCD3.M6880-CPU1. Standby only works with these PCD models.

Wrong S-Bus station number

The connected PCD’s S-Bus station number is not the same as the station number configured in the Device Configurator. Either the PCD’s configuration is wrong, or you are connected to the wrong PCD. This cannot occur if S-Bus is being used for the communications channel, and PGU mode and Auto Station are both ‘No’ . Check the Online Settings.

Wrong Serial Number

The downloader option ‘Warn if different Serial Number’ is checked, but Serial Number in the Downloader Options dialog box does not match the connected PCD’s serial number.

Wrong IP address

The connected PCD’s IP address does not match the IP address in the Device Configuration. Either the configuration is incorrect, or you are connected to the wrong PCD.

Primary and Secondary CPU x are the same device, check the Online Settings

Primary CPU 0 and Primary CPU 1 are the same device, check the Online Settings

If the Online Settings or communications wiring is incorrect, it is sometimes possible to physically connect both channels to the same PCD. For example, if PGU mode or Auto Station is not ‘No’.
Normally there will be other error messages in the Messages window if the online settings are invalid

3 Diagnostics

3.1 Halt and History Messages

If a CPU halts due to a standby system error, a message is placed in the Halt Reason Register and an entry is made in the History log.

Message	HALT	Meaning
`Red FAIL xxx`	Y/N	Caused by Redundancy failure, xxx := 0 = General error 1 = Redundancy not configured 2 = Out of memory 3 = Media transfer error 4 = CPU “Keep alive” timeout 5 = CPUs have different configurations 6 = Size of media mapping incorrect 7 = No data transfer configured 8 = Incompatible data IDs 9 = Media transfer error 2
`Red Media Map FAIL`	Y	Redundancy media map configuration error
`Red CPU x HALT`	Y	CPU x goes into HALT because the other CPU is in HALT
`Red CPU REB00T`	Y	CPU rebooted due to CPU communication failure

3.2 Diagnostic Tags

There are some diagnostic tags which show the current state of the redundancy system. The section name is [SYS-Redundancy].

These diagnostic tags are also present in the redundancy CPU1.

Message	CFG	Access	Def.	Min/ Max	Description	Remarks
Type	NO	Read only	0	2	0: OFF 1: Primary PCD 2: Secondary PCD	
TypeText	NO	Read only	–	–	“OFF” “PRIMARY” “SECONDARY”	Same as Type but as text
Status	NO	Read only	0	4	0: STARTUP 1: IDLE 2: STANDBY 3: ACTIVE - Redundancy 4: ACTIVE- Non Redundancy	
StatusText	NO	Read only	–	–	“STARTUP” “IDLE” “STANDBY” “ACTIVE”	Same as Status but as text
Standby0rActive	NO	Read only	0	1	0: Standby 1: Active	
Redundancy	NO	Read only	0	1	0: No Redundancy 1: Redundancy	
DataSync0K	NO	Read only	0	1	Data Synchronization: 0: FAILED 1: OK	
ProgSync0K	NO	Read only	0	1	Program Synchronization 0: FAILED/Different Program 1: OK/Same Program	
DataUpdateTime	NO	Read only	0	65536	Data synchronization update time in ms	
DataSendTime	NO	Read only	0	65536	Data synchronization send time in ms	
SetActive	NO	Write only	0	1	1: Set this station as Active if possible	
SetStandby	NO	Write only	0	1	1: Set this station as Standby if possible	

3.3 Diagnostic Flags

An array of 8 flags is used for diagnostic flags on both CPUs. System symbols are assigned, which are shown on the “System” tab of the Symbol Editor.

The base flag symbol name is: `S.STANDBY.DiagFlagBase`

Offset	System Symbol Name	Values	Remarks
0	S.STANDBY. PrimaryOrSecondary	0 = PRIMARY 1 = SECONDARY	On CPU1 this flag is used to test if the program is running on the Primary or the Secondary device. This allows different code to be generated or executed in CPU1.
1	S.STANDBY. StandbyOrActive	0 = STANDBY 1 = ACTIVE	On CPU0 this flag indicates if CPU1 is active or standby.
2	S.STANDBY.Redundacy	0 = NO REDUNDANCY 1 = REDUNDANCY	1 = Data sync is working, both CPU1s have the same program, standby switching can occur. If 0 then standby control is not available.
3	S.STANDBY.DataSyncOK	0 = FAILED 1 = OK	0 = Sync communications between the two CPU1s has failed, 1 = it’s working
4	S.STANDBY.ProgSyncOK	0 = FAILED / Different 1 = OK / Same	0 = programs in both CPU1s are different and sync cannot occur, 1 = programs are the same
5	-	Reserved	
6	-	Reserved	
7	-	Reserved	

3.4 Diagnostic Registers

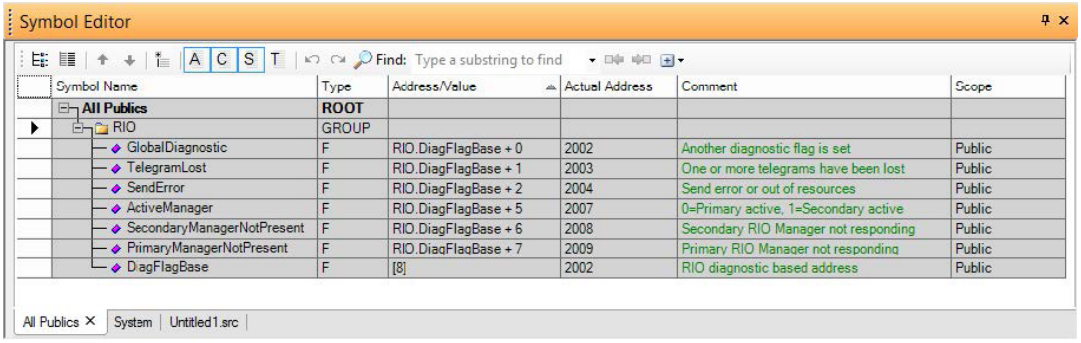
An array of 4 Registers is used for diagnostics on both CPUs. System symbols are assigned, which are shown on the “System” tab of the Symbol Editor.

The base register symbol name is: `S.STANDBY.DiagRegBase`

Offset	System Symbol Name	Values
0	S.STANDBY.Status	1: IDLE 2: STANDBY 3: ACTIVE - Redundancy 4: ACTIVE - Non Redundancy
1	S.STANDBY.DataUpdate-Time	The number of milliseconds between sync data updates
2	S.STANDBY.DataSendTime	The time in milliseconds for the sync data to be transferred (telegram time)
3	-	Reserved

3.5 Ether-SIO Diagnostics Extension

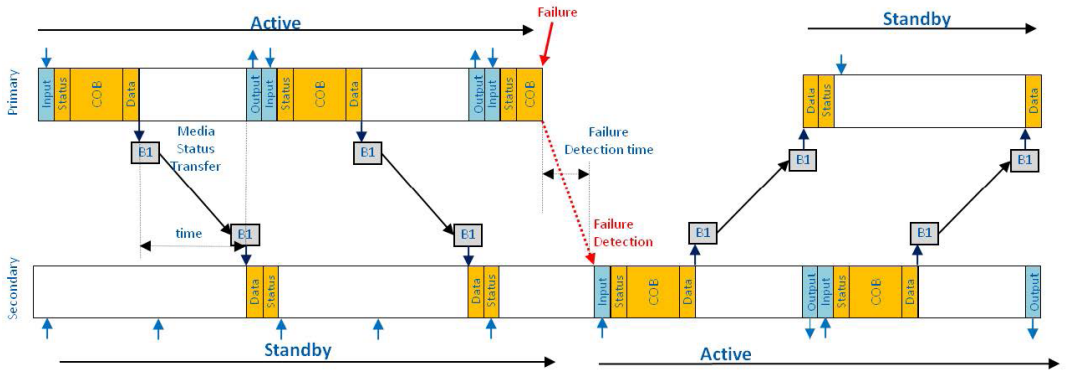
The diagnostics flags of the PCD3.T668 are extended to show the status of the two managers. These symbols are shown on the “All Publics” tab of the RIO program’s Symbol Editor, in the group RIO:



Offset	System Symbol Name	Values
0	`RIO.GlobalDiagnostic`	0 = no diagnostic flags set 1 = a diagnostic flag is set
1	`RIO.TelegramLost`	0 = no telegrams lost 1 = one or more telegrams lost
2	`RIO.SendError`	0 = no error 1 = failed to send, or out of resources*
3	-	Reserved
4	-	Reserved
5	`RIO.ActiveManager`	0 = Primary 1 = Secondary
6	`RIO.SecondaryManager-NotPresent`	0 = present 1 = not present
7	`RIO.PrimaryManager-NotPresent`	0 = present 1 = not present

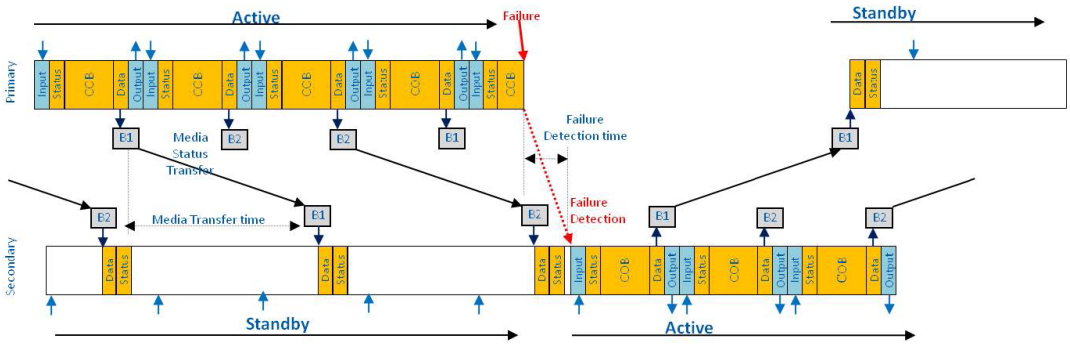
4 Technical Information

4.1 Data Synchronization and Switchover (Synchronous)



Data Synchronization: Program Cycle

4.2 Data Synchronization and Switchover (Asynchronous)



Data Synchronization: Program Cycle

4.3 LED States

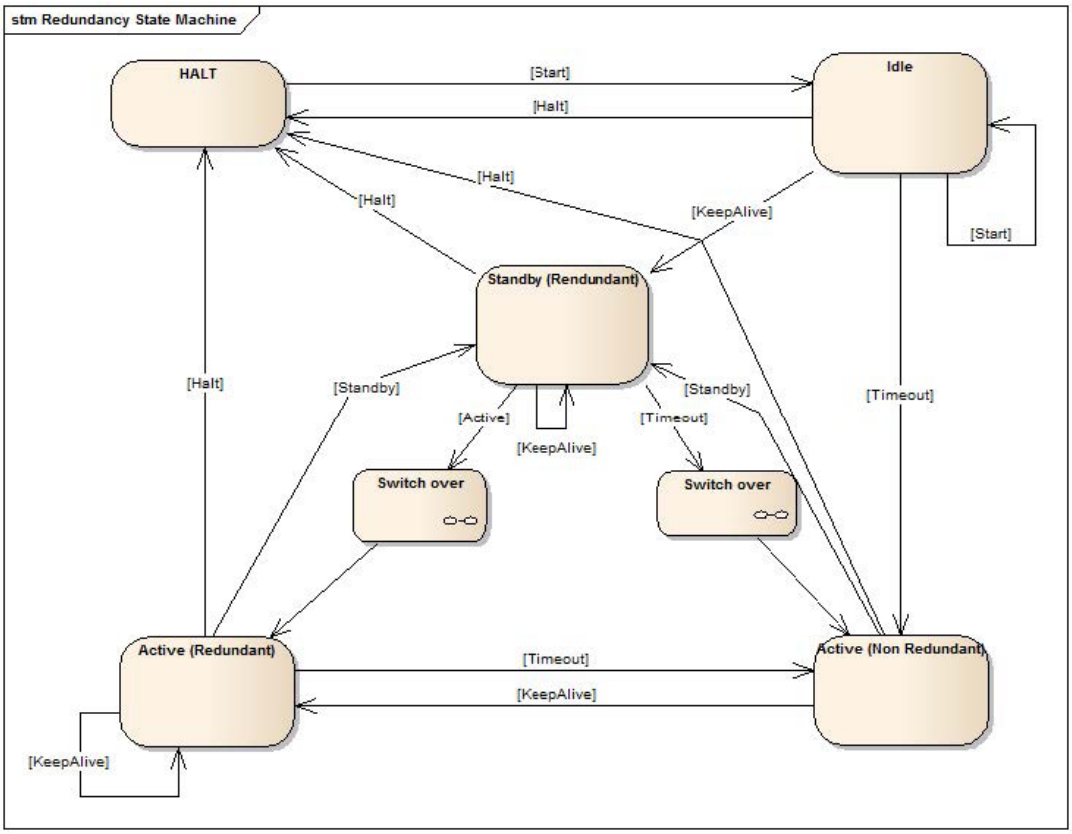
The RUN and Error LEDs on the PCD3.M6880 show the standby status:

Standby State	RUN LED State	Error LED State
IDLE	*	Blink 500ms
STANDBY	Blink 2.5s**	OFF
ACTIVE - Redundancy	*	OFF
ACTIVE- Non Redundancy	*	ON

* normal state of the RUN Led on a PCD

** If CPU0 is in: - Conditional RUN Led blinks 2.5sON/ 500ms OFF
- STOP Led blinks 500msON/ 2.5s OFF

4.4 State Machine



4.5 Troubleshooting

Problem	Cause	Remedy
The Active PCD is in RUN but the Standby PCD is in HALT.	The redundancy configuration is not the same on both PCD's.	Download the same configuration to both PCD's.

A Annex

A.1 Icons

	In manuals, this symbol refers the reader to further information in this manual or other manuals or technical information documents. As a rule there is no direct link to such documents.
	This symbol warns the reader of the risk to components from electrostatic discharges caused by touch. Recommendation: Before coming into contact with electrical components, you should at least touch the system's negative pole (cabinet of PGU connector). However, it is better to use a grounding wrist strap with its cable permanently attached to the system's negative pole.
	This sign accompanies instructions that must always be followed.

A.2 Instructions for connecting Saia-PCD® controllers to the internet

When Saia PCD controllers are connected directly to the internet, they are also a potential target of cyber attacks. For secure operation, appropriate protective measures must always be taken.
PCD controllers include simple, built-in protection features. However, secure operation on the internet is only ensured if external routers are used with a firewall and encrypted VPN connections.
For more information, please refer to our support site:
www.sbc-support.com/security

A.3 Products disposal

	**ATTENTION**	Dispose of batteries according to local regulations.	
	**ATTENTION**	According to WEEE European Directive 2012/19/EU, this device cannot be disposed of as a domestic waste.	

A

A.4 Contact

Saia-Burgess Controls AG
Route Jo-Siffert 18
1762 Givisiez 4, Switzerland

Phone central ..... +41 26 580 30 00
Phone Saia-PCD Support ..... +41 26 580 31 00
Fax ..... +41 26 580 34 99

Email support: ..... support@saia-pcd.com
Supportsite: ..... www.sbc-support.com
SBC site: ..... www.saia-pcd.com

International Represetatives &
SBC Sales Companies: ..... www.saia-pcd.com/contact

Postal address for returns from customers of the Swiss Sales office

Saia-Burgess Controls AG
Service Après-Vente
Bahnhofstrasse 18
3280 Murten, Switzerland

A