

A man in a light blue shirt is shown from the side, holding a tablet computer. The tablet screen displays a software interface with various charts and data. The background is a blurred industrial factory setting with machinery and equipment.

SIEMENS

Background and System Description • 04/2015

Rockwell Automation PLC-5 to SIMATIC S7 Application Conversion Guide

STEP 7 (TIA Portal)

<http://support.automation.siemens.com/WW/view/en/104206213>

Warranty and liability

Note

The Application Examples are not binding and do not claim to be complete regarding the circuits shown, equipping and any eventuality. The Application Examples do not represent customer-specific solutions. They are only intended to provide support for typical applications. You are responsible for ensuring that the described products are used correctly. These application examples do not relieve you of the responsibility to use safe practices in application, installation, operation and maintenance. When using these Application Examples, you recognize that we cannot be made liable for any damage/claims beyond the liability clause described. We reserve the right to make changes to these Application Examples at any time without prior notice.

If there are any deviations between the recommendations provided in these application examples and other Siemens publications – e.g. Catalogs – the contents of the other documents have priority.

We do not accept any liability for the information contained in this document.

Any claims against us – based on whatever legal reason – resulting from the use of the examples, information, programs, engineering and performance data etc., described in this Application Example shall be excluded. Such an exclusion shall not apply in the case of mandatory liability, e.g. under the German Product Liability Act (“Produkthaftungsgesetz”), in case of intent, gross negligence, or injury of life, body or health, guarantee for the quality of a product, fraudulent concealment of a deficiency or breach of a condition which goes to the root of the contract (“wesentliche Vertragspflichten”). The damages for a breach of a substantial contractual obligation are, however, limited to the foreseeable damage, typical for the type of contract, except in the event of intent or gross negligence or injury to life, body or health. The above provisions do not imply a change of the burden of proof to your detriment.

Any form of duplication or distribution of these Application Examples or excerpts hereof is prohibited without the expressed consent of Siemens Industry Sector.

Security information

Siemens provides products and solutions with industrial security functions that support the secure operation of plants, solutions, machines, equipment and/or networks. They are important components in a holistic industrial security concept. With this in mind, Siemens' products and solutions undergo continuous development. Siemens recommends strongly that you regularly check for product updates.

For the secure operation of Siemens products and solutions, it is necessary to take suitable preventive action (e.g. cell protection concept) and integrate each component into a holistic, state-of-the-art industrial security concept. Third-party products that may be in use should also be considered. For more information about industrial security, visit <http://www.siemens.com/industrialsecurity>.

To stay informed about product updates as they occur, sign up for a product-specific newsletter. For more information, visit <http://support.automation.siemens.com>.

Table of contents

Warranty and liability	2
1 Preface	5
1.1 Purpose.....	5
1.2 Conversion versus Translation.....	5
1.3 Terminology.....	6
1.4 Additional Resources.....	7
1.5 PLC Conversion phases	9
2 Hardware Conversion.....	11
2.1 Introduction	11
2.2 Controllers.....	12
2.3 I/O-Systems.....	13
2.4 Networks	17
2.5 Conversion of HMI.....	19
2.6 Conversion of Systems with Distributed Intelligence.....	20
2.7 Connecting SIEMENS and Rockwell Automation Devices	21
3 Rockwell features and their SIMATIC S7 equivalents.....	25
3.1 Introduction	25
3.2 Architectural differences between PLC-5 and SIMATIC S7	25
4 Converting the Program Structure	27
4.1 Organization Blocks.....	27
4.2 Function Blocks	27
4.3 Functions.....	27
4.4 Dividing the Logic	27
4.5 Creating a continuously executed program	28
4.6 Creating an event triggered program.....	28
4.7 Creating a periodically executed program	28
4.8 Creating a Status File	28
5 Converting the Data	29
5.1 Converting Input (I) and Output (O) data	30
5.2 Converting the PLC-5 Status file (S)	31
5.3 Converting the Binary (B) file type.....	33
5.4 Converting the Timer (T) file type.....	33
5.5 Converting the Counter (C) file type.....	35
5.6 Converting the Control (R) file type	36
5.7 Converting the Control (R) type for Serial Port Control	36
5.8 Converting the Integer (N) file type.....	37
5.9 Converting the Floating Point (F) file type.....	37
5.10 Converting the ASCII (A) file type	38
5.11 Converting the Decimal (D) file type.....	38
5.12 Converting the Block Transfer (BT) file type	39
5.13 Converting the M0 and M1 file types.....	39
5.14 Converting the Message (MG) file type	39
5.15 Converting the PID (PD) file type	40
5.16 Converting the SFC status (SC) file type.....	40
5.17 Converting the ASCII String (ST) file type	40
5.18 Converting the ControlNet (CT) file type.....	41
5.19 Converting the Constant Values.....	41
5.20 Converting Indirect Addresses	42
5.21 Converting Index Addresses	43

6	Conversion of System Software and Standard Functions	45
6.1	Introduction	45
6.2	System Functions	46
6.3	Copy Instructions.....	47
6.4	Reading and Setting System Date and Time.....	49
6.5	Interrupt Handling.....	51
6.6	Error- and Fault Handling.....	52
6.7	Diagnostics.....	52
6.8	Timers and Counters	52
6.9	Conversion Routines	54
6.10	Closed Loop Control	55
6.11	Examples of System Function Calls	56
7	Common misunderstandings when converting to SIMATIC S7	60
7.1	Introduction	60
7.2	Not Selecting Appropriate Hardware	60
7.3	The Impact of Task Scheduling.....	60
7.4	Drawbacks of straight code translation vs. conversion.....	61
7.5	Not using the most appropriate Programming Languages	61
7.6	Implementation of Incorrect Data Types – INT vs. DINT	61
7.7	User Code Emulating Existing Instructions.....	62
7.8	Extensive Usage of Jumps.....	63
7.9	Not using Symbolic Addressing.....	63
7.10	Using indexing instead of indirect addressing.....	64
8	PLC-5 to S7 Glossary	66
8.1	Introduction	66
8.2	Hardware Terminology.....	66
8.3	Software Terminology.....	67
9	Appendix A – PLC-5 parts and their S7 Equivalents.....	68
9.1	Digital input modules	68
9.2	Digital output modules	69
9.3	Analogue input modules	70
9.4	Analogue output modules	70
9.5	Analogue mixed modules.....	71
9.6	Specialty modules	72
9.7	Conversion of remote Point I/O modules.....	73
9.8	PLC-5 CPU modules	74
10	Appendix B – Panel Builder HMI Cross Reference Table	76
11	Appendix C – Figures.....	77
12	Appendix D – Tables	78
13	History	78

1 Preface

1.1 Purpose

This document provides guidance for users who have used legacy control systems based on Rockwell Automation PLC-5 and in Addition:

- have the desire or the need to take advantage of the features provided by SIMATIC S7 controllers.
- have specific Rockwell code that they wish to effectively convert into a more efficient STEP7 code.

Use this manual to adopt best practice and to avoid the commonly made mistakes, when converting projects to STEP7.

NOTE

Since the differences to other legacy Rockwell Automation control systems are minor, it is likely that the presented techniques and methods may be applicable to these as well. However there is no guarantee for it.

1.2 Conversion versus Translation

An understanding needs to be developed for the difference between conversion and translation in order to avoid the first and most commonly made mistake.

A simple translation is simply replacing a single instruction or sequence of such on one system by another instruction or sequence on the other system. This is typically done on a line by line base. Doing so, often results in missing out on additional benefits offered by the target system. This in turn results in lost opportunity for higher operational efficiencies and system availability.

Conversion on the other hand focuses on the higher level design of the application and the knowledge of both the technology and the target system. The conversion uses translation, but is not limited by a strict line by line rewriting of the existing code. More than simple line by line translation, a conversion provides the freedom to solve a certain task in another way to fully utilize the features of the new system.

If you decide to do the conversion from one of the aforementioned Rockwell systems to a SIMATIC S7 system, you need to understand three major parts:

- The original Rockwell Software
- Your application/ technology or process
- The new target system SIMATIC S7

1.3 Terminology

Understanding the Rockwell software can be achieved by either having been involved yourself in its development, or by reading the documentation of the program and of the process it controls. If the documentation is poor or missing at all, the conversion is less likely to succeed. It will most likely end up with as a mere translation.

In this case, it may be best from the cost perspective to redevelop the whole application from scratch with minimal time spent on conversion.

Understanding the application/ technology or the process is typically achieved by reading the specification or having worked in that area for a while, so certain requirements and behaviors are fully understood.

The target system can be understood by taking classes and reading manuals, however it is beneficial to understand the differences between the systems, so that you may better plan your strategy for conversion.

1.3 Terminology

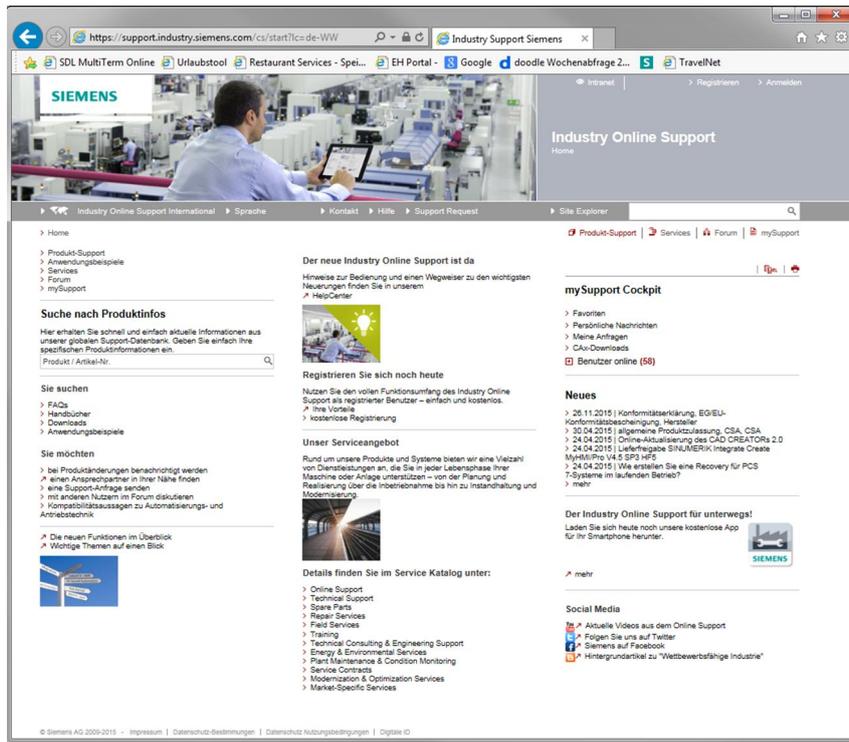
Table 1-1 – Terminology

System	Description
TIA Portal	Is the engineering framework being used to configure, program, network, commission and diagnose SIMATIC S7 controllers and operator panels. It is the single entry point to access all required functions.
RSLogix 500	Is the engineering software provided by Rockwell Automation to program the aforementioned CPU modules as well as configure and diagnose some periphery.
SIMATIC S7	Refers to the family of S7 controllers and their respective periphery modules.
PAC	“Programmable automation controller”, a term used by Rockwell Automation to describe it’s range of “Logix” controllers.

1.4 Additional Resources

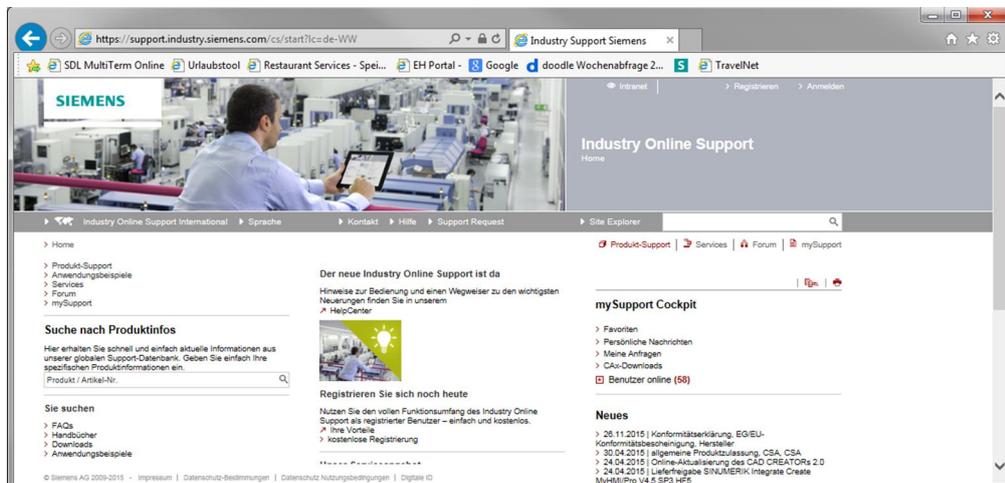
The Customer Support web pages (<http://automation.siemens.com/support>) are a vast resource for all kinds of information regarding the SIEMENS Automation product range.

Figure 1-1 – SIEMENS Industry Online Support Web Page



It contains manuals and technical specifications as well as a wide range of application examples, which include fully functioning code and well written documentation.

Figure 1-2 – SIEMENS Service & Support Web Page Heading



1.4 Additional Resources

On this web site you can search für any entry ID. Once you are registered you have access to the forum and can exchange opinion ask questions and share ideas with large community of acrive users. You may also sign up for newsletters containing links to the latest information about product releases and updates. Also you can access to the Application & Tools section, providing code samples and documentation for your quick start into your solution.

Throughout this document hyperlinks will be provided to such documents. The most relevant documents are listed below.

Table 1-2 – Referenced documents

Online Support ID	Document Title
59191792	SIMATIC S7-1500 Automation System
59191792	SIMATIC S7-1500/ ET200 MP Automation system

1.5 PLC Conversion phases

Legacy systems are often hard to repair and difficult to support. This increases the risk of unplanned downtimes leading to a decrease in productivity. The cost involved in getting such a system back up and running again can be high.

Siemens splits up the conversion process into two phases with a few sub steps.

- Conversion planning
 - Migration/ Conversion Consulting
 - Site Survey and Follow Up
- Conversion execution
 - Hardware Translation
 - Software Conversion
 - System/ Product Training

These sub steps are designed to help you convert your equipment, as effortless as possible.

Migration and Conversion Consulting

The very first step after you decide to migrate from a Rockwell Automation platform, is a concept which suits your requirements and needs. This concept addresses question such as:

- the focus of migration, e.g. Controller only, Periphery only, HMI only or any combination of the mentioned and the scope of migration, e.g. is it a single machine, a manufacturing line or a whole plant
- the acceptable downtimes
- the timeline for migration

To help answer all pertinent questions, a standardized checklist is provided as part of the migration and conversion consulting.

Site Survey and Follow Up

The next step after the initial consultation should be a site survey to get a detailed list of the installed base. This information is then used to refine the original concept. The result will be discussed with you in order to answer open questions and to inform about the results. Once again a standardized checklist will be provided to you.

Hardware Conversion

After the site survey has been completed, an equivalent SIEMENS hardware will be identified based on the existing components. Already at this time, it may be possible to identify the first benefits of moving to a Siemens solution. Examples of which could be the use of fewer modules, providing cost savings, or it may be the use of remote peripheral systems instead of the current centralized periphery, allowing for a more agile, often space saving decentralized approach.

All this can be decided during the conception phase.

Software Conversion

Well trained personnel knowing the target and legacy systems, assist you in this part of the migration process. It addresses the code used in the legacy system responsible for process control. This code will be migrated and adopted to the modern SIEMENS target system. The formerly used file formats by the Allen-Bradley legacy systems (6200, APS or AI) can be used as input. The result is automation software in the latest SIEMENS Automation Software format, which is capable of controlling your process.

System/ Product Training

Operation and maintenance of the new system is essential for a successful migration. To make this an easy task for your personnel, SIEMENS offers product and system training to familiarize operations- and maintenance personnel with the new system.

You may use any combination of the above mentioned services. There is no obligation using one service to use another service as well.

To schedule a conversion project, or learn more about the services offered, contact your local SIEMENS Automation sales office or authorized distributor.

NOTICE	All the above mentioned services are subject of extension, cancellation or local availability. The costs for some of these services are on a case-by-case basis, depending on complexity and project size.
---------------	--

2 Hardware Conversion

2.1 Introduction

In this chapter the document provides guidance, on how to determine the best fitting hardware as a replacement for the existing Rockwell equipment.

After studying this chapter, you should be able to select the most appropriate hardware. It describes how to select controllers, local and remote I/O, select networks and protocols as well as select appropriate HMIs. It also provides conversion examples for the most common Rockwell Systems.

Table 2-1 – Topics of chapter “Introduction”

Topic	Page
Controllers	12
I/O Systems	13
Networks	18
Conversion of HMIs	20
Conversion of systems with distributed intelligence	21
Connecting SIEMENS and Rockwell Automation Devices	22

2.2 Controllers

The following table lists an incomplete selection of the Allen- Bradley legacy controllers and a possible SIEMENS S7 controller replacement.

Table 2-2 – PLC-5 Controller Replacement

Controller	Catalogue Number	SIMATIC equivalent	Part Number
PLC-5/11	1785-L11B	CPU 1511-1 PN	6ES7 511-1AK00-0AB0
PLC-5/20	1785-L20B/ -L20E/ -L20C15	CPU 1511-1 PN	6ES7 511-1AK00-0AB0
PLC-5/30	1785-L30B	CPU 1511-1 PN	6ES7 511-1AK00-0AB0
PLC-5/40	1785-L40B/ -L40E/ -L40C15	CPU 1513-1 PN	6ES7 513-1AL00-0AB0
PLC-5/40L	1785-L40L	CPU 1513-1 PN	6ES7 513-1AL00-0AB0
PLC-5/60	1785-L60B	CPU 1515-2 PN	6ES7 515-2AM00-0AB0
PLC-5/60L	1785-L60L	CPU 1515-2 PN	6ES7 515-2AM00-0AB0
PLC-5/80	1785-L80B/ -L80E/ -L80C15	CPU 1515-2 PN	6ES7 515-2AM00-0AB0
PLC-5/26	1785-L26B	CPU 1511-1 PN	6ES7 511-1AK00-0AB0
PLC-5/46	1785-L46B/ -L46E/ -L46C15	CPU 1513-1 PN	6ES7 513-1AL00-0AB0
PLC-5/86	1785-L86B	CPU 1515-2 PN	6ES7 515-2AM00-0AB0

NOTE The above mentioned range of PLC-5 controllers can be replaced by a S7-1500 controller of type 1511, when the selection is purely based on possible I/O count, memory and execution speed for instructions.

NOTICE Keep in mind that the above table is only an incomplete list of Allen- Bradley controllers. Safety and redundant configurations are not part of this list.
Suitability of a certain controller must be evaluated on a case-by-case basis in order to select the most appropriate CPU.

It should be noted, that the advanced memory protection of the PLC-5/x6 controllers are an integral part of all SIMATIC S7-1200 and S7-1500 series controllers.

The selection of a specific controller is typically based on the following criteria, which a given by the existing application.

- Required response times (terminal to terminal response time)
- Memory requirements for both
 - Code
 - Data
- Number of required I/O's

NOTE Code efficiency is different for the SIMATIC S7 and the Rockwell PLC-5 controllers. That is the memory required for the program will be different in SIMATIC S7 and PLC-5.

2.3 I/O-Systems

Allen- Bradley controllers have access to a wide range of periphery modules, which are provided either by Allen- Bradley or third party vendors from Rockwell's Encompass partner program. SIEMENS provides an equivalent for the vast majority of these modules.

Allen- Bradley's controllers are placed into a rack, which also offers slots for periphery modules. Should the central rack not provide sufficient space to accommodate all peripheral modules, it may be extended through the use of extension racks. Special controllers, so called Ethernet and ControlNet controllers, offer additional interfaces to connect to Ethernet or ControlNet in order to extend the functionality further.

SIEMENS does not offer such special controllers, since the standard controllers already offer connectivity to Ethernet or PROFIBUS or both.

Local I/O

SIEMENS offers a wide range of periphery modules for the different families. The periphery modules of the S7-1500 series are also usable with the decentralized (remote) periphery system ET200MP, which reduces the number of required spare parts and thus reducing storage costs.

The S7-1500 is a truly modular system. It does not require a specially designed Power Supply module; neither does it have a fixed number of slots occupying space in the cabinet even when not in use.

The mounting takes place by placing the modules onto a metal profile rail and fix them onto it using provided screws. These rails can be cut to length, optimizing the cabinet space down to the bare minimum. The electrical connection is realized using supplied U-connectors. These U-connectors provide enough mechanical stability for office or laboratory use. However when installed in the field, cabinet installation onto the metal rail is recommended. The rail adds mechanical stability, provides a proper grounding point as well as helps with heat dissipation.

In the SIMATIC S7 1500 the maximum number of periphery modules in a central configuration is 32 modules. There is no need to extend the backplane beyond the maximum of 32 modules.

With all SIMATIC platforms, further I/O can be networked via

- PROFIBUS
- PROFINET
- ASi networks

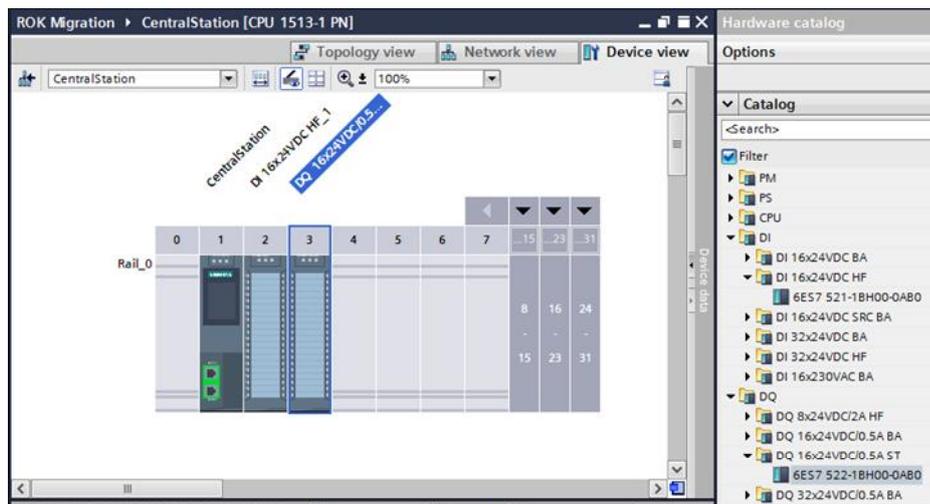
with a tight and seamless I/O integration into the system.

The following figure shows a partially done centralized configuration for a SIMATIC S7-1500. Please note also the hardware catalogue on the right hand side.

2 Hardware Conversion

2.3 I/O-Systems

Figure 2-1 – TIA Portal S7 central 1500 station



The PLC-5 is mounted into a rack with a power supply module. The rack provides a fixed number of slots usable for controller and peripheral modules.

Once the central rack is completely filled, the PLC-5 controllers can extend the backplane into multiple additional racks. They are connected as:

- Local I/O
- Remote I/O
- Extended-local I/O

The following table lists some of the more common I/O modules to be found in applications.

Table 2-2 – SIMATIC equivalents to PLC-5 I/O modules

PLC-5 modules	Description	SIMATIC equivalent	Part Number
1771-IB	8 channel digital input module 10..27V DC	SM 521 (16x DI)	6ES7 521-1BH00-0AB0
1771-IBD	16 channel digital input module 10..30V DC	SM 521 (16x DI)	6ES7 521-1BH00-0AB0
1771-IBN	32 channel digital input module 100.30V DC	SM 521 (32x DI)	6ES7 521-1BL00-0AB0
1771-OB	8 channel digital output module 10..27V DC	SM 522 (16x DQ)	6ES7522-1BH00-0AB0
1771-OB D	16 channel digital output module 10..30V DC	SM 522 (16x DQ)	6ES7522-1BH00-0AB0
1771-OB N	32 channel digital output module 10..30V DC	SM 522 (32x DQ)	6ES7522-1BL10-0AA0

NOTICE Keep in mind that the above table is only an incomplete list of the Allen- Bradley periphery modules. Suitability of a certain periphery module must be evaluated on a case-by-case basis in order to select the most appropriate SIMATIC periphery module.

A complete list of Allen- Bradley PLC-5 periphery modules and a SIMATIC equivalent can be found in the Appendix A.

Remote I/O

In most cases the use of a remote I/O system is the most cost efficient method to pick up signals from sensors in the field or drive actuators installed at the site. In the SIMATIC system this can be easily achieved using PROFIBUS or PROFINET as well as with ASi.

Extended central configurations

With Allen- Bradley Remote I/O, ControlNet, DeviceNet or EtherNet/IP networks for such remote periphery systems are common. The Allen- Bradley controllers can use periphery modules from the 1771 product range, which are the same plugged into the main rack.

The SIEMENS equivalent installation would be a central rack containing the CPU module as well as up to 32 periphery modules of any flavor. This configuration can be extended decentralized with ET200MP racks.

Decentralized configurations

In the Rockwell world decentralized configurations are realized using

- Remote I/O
- DeviceNet
- ControlNet
- EtherNet/IP

Each of those networks has a more or less equivalent in the SIMATIC systems.

Table 2-3 – Periphery network comparison

Allen- Bradley network	Used for what applications	SIEMENS equivalent
Remote I/O	With Remote I/O the controller can use further periphery modules not fitting in its local physical rack. It allows for a daisy chained connection of multiple racks.	The S7 1500 does not have an equivalent concept, but can easily resemble such configuration with any of the SIEMENS supported field busses.
DeviceNet	DeviceNet allows the controller to read inputs from and control outputs to a device with low data volume	ASi (Actuator Sensor Interface) provides an easy access to input and output data from sensors and actuators with a low data volume (see glossary).
ControlNet	ControlNet network provides both I/O control and peer-to-peer communications	PROFIBUS is able to provide deterministic behavior for I/O control as well as methods for peer-to-peer communications (see glossary).
EtherNet/IP	Ethernet network supporting communications for both <ul style="list-style-type: none"> • periphery • messaging exchange 	PROFINET can solve the same tasks plus the extra of being able to run isochronously to a program (see glossary).

Using any of the above mentioned networks other Allen- Bradley periphery systems may be connected.

Table 2-4 – Allen- Bradley periphery systems and SIEMENS equivalents

Allen- Bradley System	DeviceNet	ControlNet	EtherNet/ IP	SIEMENS equivalent
1746 series (SLC 500)	N/A	Usable	N/A	ET200M/ ET200MP
1794 series (FLEX I/O)	Usable	Usable	N/A	ET200L
1794 Ex series(FLEX Ex I/O)	Usable	Usable	N/A	ET200iSP
1791D series (CompactBlock)	Usable	N/A	N/A	ET200L
1734 series (POINT I/O)	Usable	Usable	N/A	ET200S/ ET200SP

NOTICE This table is not an exhaustive list of the Allen- Bradley peripheral systems. Suitability of a certain peripheral system must be evaluated on a case-by-case basis in order to select the most appropriate SIMATIC equivalent.

2.4 Networks

As already mentioned Allen- Bradley controllers use networks not only to extend reach in regards to periphery, they also use networks for data exchange. This data exchange can be between controllers (C2C communication) or between controllers and other computer systems, such as programming computers or higher level systems.

Network communications are configured with NetLinx and is focused on three primary networks

- EtherNet/IP
- ControlNet
- DeviceNet

All these networks are designed using the Common Industrial Protocol (CIP), which allows for configuration, control and collection of data over any of the above networks.

Ethernet/IP

This network uses TCP and UDP over an IP based network. It allows for

- General I/O control
- Data exchange between controllers (C2C)
- Connectivity to enterprise systems
- Integration of safety devices
- Motion Control systems

In a SIEMENS system an engineer would likely use the integrated PROFINET interface of the selected controller to achieve all the same.

ControlNet

This real-time control network provides transport methods for time- critical I/O and interlocking data as well as for programming and configuration data. Typical applications include:

- General I/O control
- Data exchange among controllers (C2C)
- Backbone to multiple distributed DeviceNet networks

PROFIBUS offers equivalent functionality, which can be used with the integrated interfaces of the SIMATIC controllers.

DeviceNet

With a design focused on low volume data from and to devices DeviceNet offers real time operation for the following applications:

- Distributed devices with few I/O points
- Data and power over the same physical media
- Direct connection of devices

For applications with such characteristics, ASi is the fieldbus of choice in the SIEMENS' world. Configuration is done with a simple push of a button and it provides the added benefit of deterministic control.

2.5 Conversion of HMI

The HMI represents the interface to a single machine or to the whole plant. It is therefore a must that the operators find it easy and intuitive to use.

As an engineer, who wants to migrate such system it is important to understand the operations philosophy behind the installed system. Otherwise the new system won't be accepted by operators. This is not limited to the HMI but spreads out to the whole system. For this reason it is important that the new HMI be operated using the established philosophy as much as possible.

The operations philosophy is represented by:

- the Look& Feel of the screens
- the location of operations elements, such as button, indicators
- the navigation between screens

SIEMENS offers support in migration of third party HMIs into SIEMENS systems.

This support includes

- Evaluation of used functionality
- Selection of appropriate SIEMENS hardware
- Conceptual design of the new HMI software

2.6 Conversion of Systems with Distributed Intelligence

This section covers:

- how a general discrete control application containing a group of functional units can be built using multiple controllers.
- how a similar method can be applied to a process control application that is designed to the S88 standard.

Hardware and Software Implementation

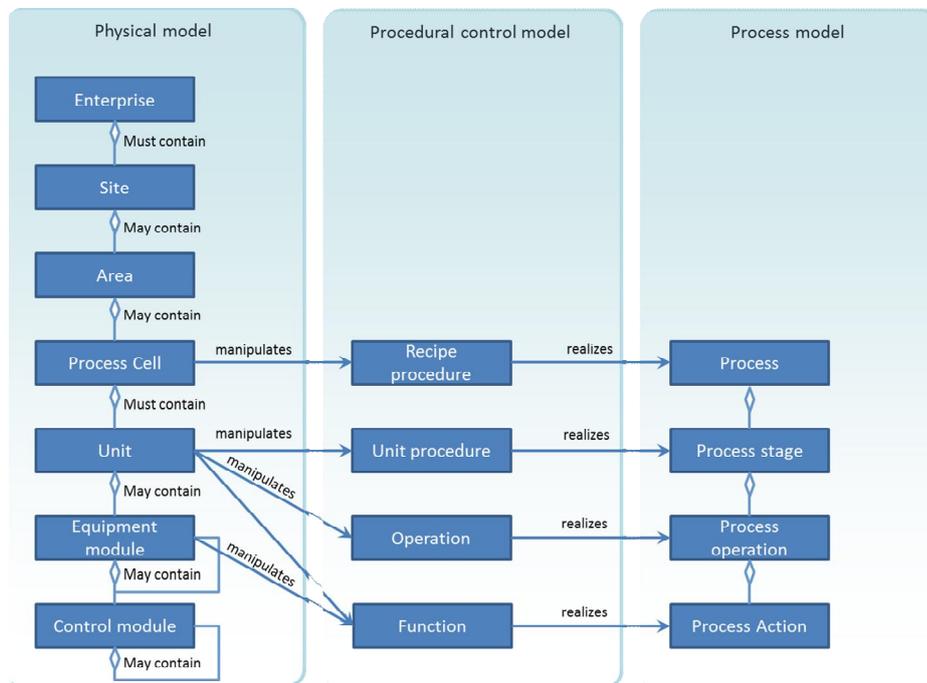
General Discrete Control:

The hardware and software model for distributed logic for general discrete control is shown below. In this case the supervisory role will be played by a controller. PROFINET or PROFIBUS network can be used to interconnect the controllers. Peripheral data exchange or controller to controller communication (C2C) can be used to exchange data within the system.

Process Control:

The diagram below illustrates the hardware and software structure for an S88 process control application. The PC will be running SIMATIC Batch software, which is a software package for running production batches by means of recipes. SIMATIC Batch software resides in a PC and communicates with each controller via any network (PROFIBUS or PROFINET).

Figure 2-2 – Hardware and software structure for a S88 process control application



2.7 Connecting SIEMENS and Rockwell Automation Devices

From the perspective of the engineer or customer who wants to migrate to a modern SIEMENS system, it is possible to describe two major scenarios

- a complete retrofit, where the single machine is a new design using SIEMENS equipment
- a partial retrofit, where parts of the Allen- Bradley system are maintained

Each of these scenarios has other challenges to master. The following section covers them and describes solutions for them.

Considerations for incorporation into existing installations

A complete retrofit is most likely a new system/ machine/ plant which do not contain any preexisting installations. This scenario is typically found at OEMs building machines with control systems from a single vendor. However, there is never a real “complete retrofit” since the engineer or customer has done already some machines with another vendors control system, thus having already solutions in his mind. These may be a hardware layout and selection related or concerning the software design circling around the available functionality in Allen- Bradley controllers.

Latest when delivering the new system/ machine to his customer the OEM faces the situation to interconnect to an existing system on a higher level.

Options for incorporation into existing systems

SIMATIC controllers offer a wide range of protocols used over Ethernet. These protocols are either

- TCP/IP based
- UDP/IP based
- ISO-on-TCP based, aka COTP (connection oriented transport protocol using port 102)
- ISO Transport

In some cases it may be necessary to use PROFIBUS with one of its protocols to interconnect. These supported protocols are:

- PROFIBUS DP (both as Master or as Slave)
- PROFIBUS FMS (rarely used in the real world)
- PROFIBUS FDL (common used protocol for C2C)

The above mentioned protocols can be used either through the integrated interfaces or through InChassis communication modules (CP cards).

NOTE

The InChassis communication processors are known as bridges to the PLC-5 programmer. They may add further communication protocols, additional functionality or provide additional interfaces.

Partial retrofit

In a partial retrofit scenario the engineer or customer decides to maintain parts of the preexisting control system or extend them with new parts. These parts could be

- Controller
- Periphery, which includes VSD (VFD) and other network attached devices
- Sensors and Actuators
- HMI/ Operator Panels

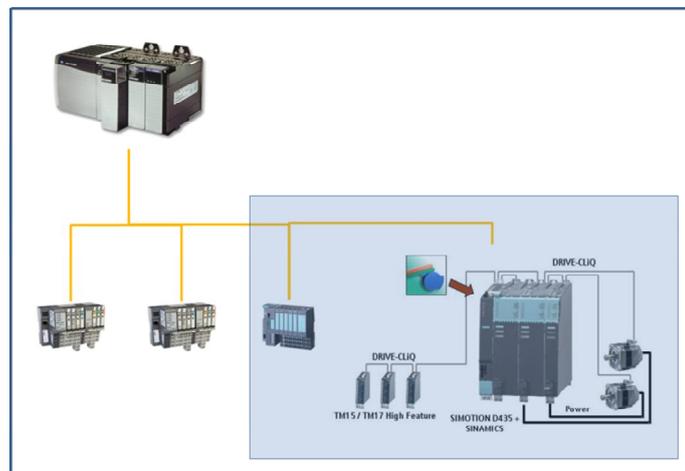
The following section addresses these scenarios in detail

Maintaining the Controller

In this scenario the controller will be kept. The only parts exchanged or being added are the periphery systems and possibly sensors and actuators. Here one of three solutions can be used.

- Using ET200S Interface module with EtherNet/IP or DeviceNet protocol to provide connectivity to the PLC-5 controller. This requires the existence of an EtherNet/IP interface card or an Ethernet PLC-5 controller. SIEMENS also offers VFD (VSD) systems with the above mentioned protocols built in.

Figure 2-3 – Partial Retrofit



- Using PROFIBUS DP and SIEMENS periphery systems on this fieldbus, by adding a PLC-5 PROFIBUS DP Scanner (Master module) from Rockwell's Encompass partner Molex such as SST-PB3-REM-R (http://www.literature.molex.com/SQLImages/kelmscott/Molex/PDF_Image_s/987650-5731.pdf)
- Using an EtherNet/IP to PROFINET gateway (<http://www.anybus.com/products/products.asp?PID=372&ProdType=Anybus%20X-gateway>) from Rockwell's Encompass partner HMS or ProSoft.

This scenario is an unlikely one, since most the time an engineer wants to use the features and functions a more modern controller offers.

Maintaining the Periphery

With this situation the customer keeps the periphery systems and replaces the controller only. This way one gains the increased performance in program execution. This only has a limited effect on throughput of a specific machine, since most the time the periphery and/ or the mechanical parts are limiting the production speed. However it may shorten response times, which increase safety aspects.

Two possible methods can be used in order to keep the Allen- Bradley periphery systems.

- a. Replacing the DeviceNet, ControlNet or EtherNet/IP adapter with a PROFIBUS slave adapter
- b. Inserting a gateway to PROFIBUS or PROFINET into the existing installation

This scenario is the most common one. It allows for taking advantage of the modern software architecture a SIMATIC controller offers. A step-by-step approach can be taken to replace peripheral systems, whenever possible, one at a time or all at once.

Sensors and Actuators

When keeping the sensors and actuators the only thing that needs to be taken care of is the physical interface and the logical protocols being used by them. An appropriate interface module needs to be found, able to handle the preexisting interface.

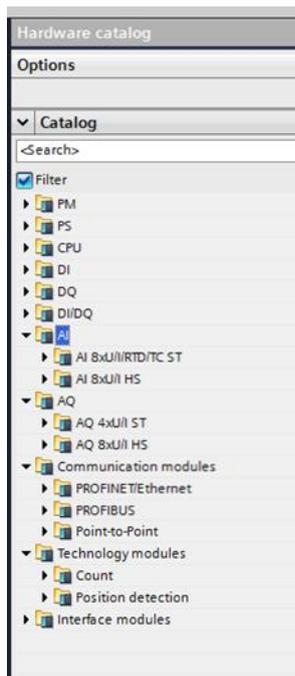
Due to the wide range of periphery modules SIEMENS offers, such a module can most likely be found.

For more information, refer to the ET 200MP/ ET 200SP manuals.

[\(ET 200SP Manual Collection – Link ID: 84133942\)](#)

[\(SIMATIC S7-1500 /ET 200MP Manual Collection – Link ID: 86140384\)](#)

Figure 2-4 – TIA Portal Hardware catalogue



HMI/ Elektronik Operator Panels

In addition to peripheral I/O systems, HMI/EOPs can also be replaced separately as peripheral devices. The available protocols of such panels need to be considered.

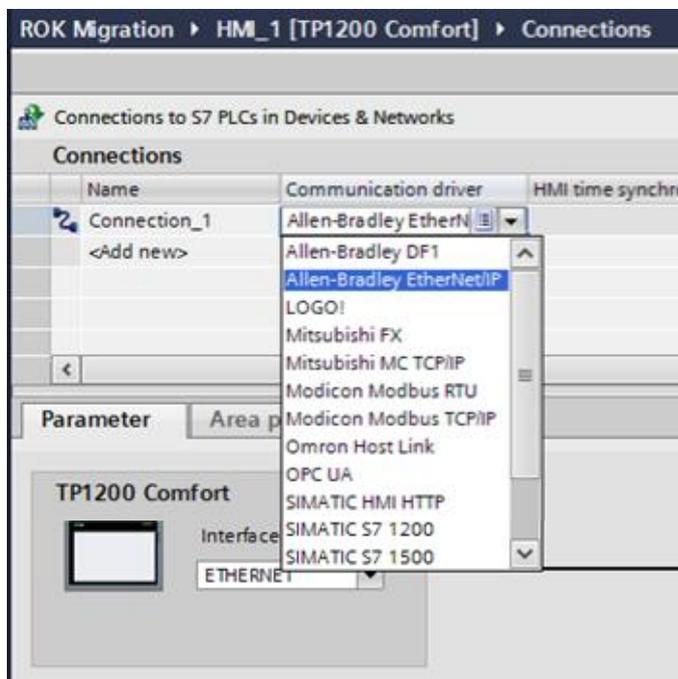
Although it is priority number one to maintain functionality as well as “Look& Feel”, new features are included in the panels, which allow improved functionality.

Typical protocols used by PLC-5 controllers are

- DF1 (+)
- DH485
- ControlNet
- DeviceNet
- EtherNet/IP (+)
- Remote I/O (RIO)
- RS 232

The SIEMENS Comfort Panels have some of the Allen- Bradley protocol drivers already included. These are marked with a plus (+) in the above list.

Figure 2-5 – TIA Portal Panel connection types



3 Rockwell features and their SIMATIC S7 equivalents

3.1 Introduction

This chapter describes the SIMATIC S7 features that may not be familiar to Rockwell users as well as the commonalities.

There are common features to both systems such as the peripheral structure and the ability to program fully symbolic. Other features are easier to use, such as networking multiple PLCs or HMIs, or simply no longer necessary, such as tag mapping to or alarm configuration in the HMIs.

This chapter contrasts those features that differ and compares those features that have underlying similarities.

The objective is to:

- Provide information on TIA Portal features that will make design process faster and easier for converting engineer.
- Provide an overview of out of the box functionality of SIMATIC S7 controllers, to avoid the unnecessary conversion of functionality that needed to be programmed in PLC-5 systems.

3.2 Architectural differences between PLC-5 and SIMATIC S7

Since the SIMATIC S7 controllers represent an advanced generation of controllers over the PLC5 systems, there are inherent differences in their architectures and capabilities. This section lists the most significant differences.

CPU

The PLC-5 processor is based on 16-bit operations. The SIMATIC S7 processors also provides 32-bit and 64-bit operations, allowing you to maintain existing code, but offers the possibility of increased precision and speed for certain operations. i.e. Floating-point numbers and masking tasks.

Operating System

In PLC-5 processors, program execution is determined primarily through the use of main control programs (MCPs). In addition, selectable timed interrupts (STIs) or input interrupts (DIIs/PIIs) are available. In a SIMATIC S7 controller, this functionality is realized through the use of Organizational Blocks (OBs) with continuous execution, fixed/adjustable timed execution or triggered execution through peripheral interrupts.

Input and Output

The PLC-5 processor maps I/O memory into I and O data table files. The I/O data is updated synchronously to the program scan so you know you have current values each time the processor begins a scan. In SIMATIC S7 controllers, peripheral I/O is mapped synchronously with the logic into Process-Image-Partitions (PIP). In this sense, the peripheral mapping of the two systems is consistent, with the exception, that in a SIMATIC S7 system, one can bypass the PIP and directly access the peripheral points by using a “:P” suffix.

Data

The PLC-5 only knows global data, which are stored in global data tables. Each PLC-5 data table can store several words of related data.

The SIMATIC S7 knows global and local data. Global data are stored in data files (called data blocks), which are accessible from any place in the program. Local data is stored locally inside a program and is freed after the block has been left. Static data are storable locally. Related data can be organized in structures or arrays independent of data type. User defined data types allow for declaration of structures as data types.

Times

Time variables in the PLC-5 processors are stored in Timer files, which are 16-bit wide and have selectable time bases of 10ms and 1s.

SIMATIC times are 32-bit or 64-bit wide and support 1ms (32-bit) or 1ns (64-bit) time bases (not selectable). As such, scaling of preset values will be required to adjust for time base differences.

Communications

The PLC-5 processor supported block transfer read and write (BTR and BTW) instructions are specific to Allen- Bradley controllers. The SIMATIC S7 uses system function calls to realize this functionality. Reading data from peripheral modules is typically achieved using either "DP_RDDAT". Writing a block of data is realized using the system function "DP_WRDAT". However under certain circumstances, such as reading diagnostic data or writing configuration data the system functions "RDREC" and "WRREC" are being used.

Communications to other automation systems is realized with a "MSG" instruction and preconfigured messages to be sent or received.

The SIMATIC S7 system uses a more generic approach to communicate with other automation systems. It uses the more intuitive instructions of "T_SEND" and "T_RECV".

4 Converting the Program Structure

The SIMATIC S7 controller uses a different execution model than the PLC-5 processor. The SIMATIC S7 controller operating system supports:

- Organization Blocks
- Function Blocks
- Functions

This chapter provides a short description of the SIMATIC S7 controller to help you better understand the translation process.

4.1 Organization Blocks

The initial starting point for user code execution is an Organization Block in the SIMATIC S7. The Organization Blocks have a specific function assigned to it. They contain user code, which is able to control peripheral signals. It may also branch into further code blocks realizing the program logic. The Organization Blocks may be programmed in any of the available programming languages, with the exception of SFC.

4.2 Function Blocks

The program logic for a complex system can easily become large, making it difficult to maintain and debug. Splitting the code into smaller pieces is a good idea to keep it maintainable.

Function Blocks are such smaller code blocks with the capabilities to accept input parameters and provide output parameters. This allows for reusable code fragments including own instance specific persistent storage.

4.3 Functions

These code blocks have the same features as a function block with the difference of not having instance specific persistent storage. The functions are more like a mathematical function accepting an input vector and calculating an output vector.

4.4 Dividing the Logic

The PLC-5 controllers store their program logic in program files programmed in either SFC or LAD. Program files can branch into other program files. They may be assigned to certain events thus being executed only in case of such an event. The PLC-5 controller differentiate between continuous, periodic and event triggered execution

- Continuous execution is being realized by programming the LAD2 file in the PLC-5 controller or assigning additional/ different file numbers to controller registers. The program logic is being executed till the end and restarted from its beginning after the scan has finished.
- Code execution is considered event driven when an external event occurs and triggers the execution of code dedicated to this event.
- Periodic execution of program logic starts the execution on a given event which reoccurs after a predefined time period. In between the execution of the periodic program portion any program logic with lower priority will be executed.

4.5 Creating a continuously executed program

The SIMATIC S7 supports at least one continuous task. It is an Organization Block, which is restarted by the controller's operating system after it has been executed. This Organization Block operates at the lowest priority, which means any event interrupts the continuous Organization Block. The default continuous Organization Block traditionally gets the name "Main" and is realized as OB1. On SIMATIC S7-1200 and S7-1500 systems multiple continuous Organization Blocks can be created. Here the numbering, such as OB1, is no longer relevant.

4.6 Creating an event triggered program

Event programs are called whenever a certain events occur. Best mapping of a DII or PII instruction from the PLC-5 would be the use of a "process interrupt" The SIMATIC S7 distinguishes between different interrupt sources and therefore different interrupt types. Each interrupt source can have its own program assigned to, thus making programming necessary to distinguish between the different sources based on a bit field, obsolete.

Event tasks responding to events triggered by a signal change (e.g. rising edge) are called "process interrupts" and traditionally triggers Organization Block OB40 or one of the sequential OBs to OB47. In the SIMATIC S7 1200 and S7 1500 systems the numbering is no longer relevant, as a multitude of such interrupts are available, and the number is freely assignable above OB123. Such events are triggered by the operating system of the SIMATIC processor whenever the defined event occurs.

4.7 Creating a periodically executed program

A periodic interrupt, like the one generated by the STI timed interrupt in a PLC-5 system, are replaced by cyclically-interrupting OBs in SIMATIC S7 systems. They receive a default name "CYC_INTx", where x is a number and can take on numbers 30 and higher. Multiple such cyclic OBs are possible; the exact number is dependent on the specific S7 controller being used. Programmatic enabling and disabling of such interrupts is also supported on SIMATIC S7 systems.

4.8 Creating a Status File

While the PLC-5 maintained a status file to provide diagnostic information, the modern SIMATIC S7 keeps such information internally. The human readable format for diagnostic information is provided as "Diagnostic Buffer". It provides a history of the last 100 entries including timestamp.

Such diagnostic entries are considered events in the SIMATIC world. They trigger diagnostic interrupt routines which are generally found in OBs 80-123.

For a complete overview of the available pre-defined organizational blocks and their properties and configuration, see the manual of the corresponding controller.

[\(SIMATIC S7-1200 controller manual- Link ID: 91696622\)](#)

[\(SIMATIC S7-1500 controller manual – Link ID: 71704272\)](#)

5 Converting the Data

PLC-5 controller typically store data in several globally accessible “Data files”. Each of the data files contains a collection of variables of a single data type. The best examples of this are the I/O data table files, which contain an array of words.

In the more modern world of SIMATIC the user is not limited by such an artificial separation. Data Blocks, which are the modern equivalents to data files, can store all kinds of data types. The variables or tags can either be stored as a single variable or as part of a structure or an array together with other tags of different types.

Access is possible through either the absolute address, if available, or more efficiently by its symbolic representation from anywhere in the code. This applies to global data blocks. In S7-1200 and S7-1500 systems, pure symbolic access is suggested in order to optimize memory usage.

Since the PLC-5 processor supports only up to 16-bit datatypes, while the SIMATIC S7 systems can support up to 64-bit datatypes, a conversion to the wider data types is recommended, but not required.

Table 5-1 – PLC-5 data types and the SIMATIC equivalents

PLC-5 data type	SIMATIC data type
O	BOOL
I	BOOL
S	n.a.
B	BOOL/BINARY
T	TIME
C	COUNTER
R	n.a.
Serial Port Control	n.a.
N	Integer
F	Floating Point
A	ASCII
D	BCD
BT	n.a.
MG	n.a.
M0	n.a.
M1	n.a.
PD	PID Controller
SC	GRAPH
ST	String
CT	n.a.
Constants	Constants
Indirect Addressing	n.a.
Indexed Addressing	n.a.

5.1 Converting Input (I) and Output (O) data

A conversion of the PLC-5 Input and Output data table files is necessary, due to the different addressing schemas. The PLC-5 processor uses octal addressing, while the SIMATIC S7 uses decimal addressing.

Table 5-2 – PLC-5 addressing to SIMATIC addressing

PLC-5 original address	Comment	SIMATIC converted address
I:007	Addresses the 8 th word of the Input data table file	IW7
O:010	Addresses the 9 th word of the Output data table file	QW8
I:021/05	Addresses the 6 th bit in the 18 th word of the Input data table file	I36.5
O:035/15	Addresses the 14 th bit in the 30 th word of the Output data table file	Q61.5

It is important to understand the difference in addressing. In the PLC-5 the input and output data table files have 16-bit wide elements. The SIMATIC allows for 8-bit wide elements. Conversion of the addresses follows a simple algorithm:

1. Convert PLC-5 offset address from an octal to a decimal number
2. Double the offset to get the SIMATIC S7 byte offset
3. Convert the PLC-5 channel displacement from an octal number to a decimal number
4. Calculate a Modulo 8 to get the channel displacement for the SIMATIC S7

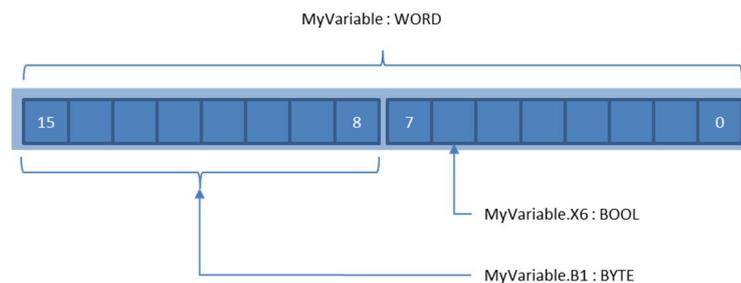
NOTE

Depending on the real layout of the signal modules inside the rack, a different address layout may exist. It may become necessary to reorganize the address layout in the S7-1500 completely.

The addressing schema of the PLC-5 controllers described above is more complex than it seems from the above description. In fact the addressing schema contains rack and slot information, where the rack information depends on the addressing schema selected in the PLC-5. The PLC-5 controller allows for ½ slot, 1 slot and 2 slot addressing creating logical racks influencing the allocation of memory in the “I” and “O” files.

Slice addressing is possible with SIMATIC controllers, allowing you offset addressing to “slices” or subsegments of larger data structures. This allows bit addressing within bytes/ words/ double words/ long words, byte addressing in words/ double words/ long words and word access to double words/ long words. Slice addressing is as simple as attaching a suffix to the larger data type denoting the desired offset.

Figure 5-1 – TIA Portal Slice access



5.2 Converting the PLC-5 Status file (S)

The status file of a PLC-5 controller is used to provide both, status/diagnostic information about the controller, as well as configure program and interrupt execution. SIMATIC S7 controllers offer the same functionality and more, however the configuration of program execution and use of diagnostics is handled differently

Table 5-3 – PLC-5 Status file data integration into a SIMATIC program

PLC-5 Status data	SIMATIC realization
Arithmetic flags (S:0)	This status word contains information about the status of the last arithmetic operation, such as the carry, overflow, zero and sign flags. The SIMATIC S7 maintains a status word, which is implicitly evaluated by certain instructions. An access to these bits, as with PLC-5 processors, is possible but not necessary.
Processor status (S:1)	The processor's status is represented by flags in this status word, indicating the current situation the processor is in. In the SIMATIC S7 access to this information is typically not necessary, since there are other methods available to retrieve this information.
Switch setting information (S:2)	In here the PLC-5 processors store information, set using switches on the outside of the processor module. Since the S7-1200 and S7-1500 doesn't have any switches other than the operating mode selector switch, these information are configured using software (TIA Portal). The equivalent information is available to the user program by means of calling system functions. Others cannot be applied to a SIMATIC S7 and should be ignored and access to them removed from the user program.
Active Node table (S:3-S:6)	This table is filled with information about the active nodes on a DH+ network connected the PLC-5 processor's channel 1A. An equivalent life list is built in into the SIMATIC controllers, which is accessible through a call to a system function "DeviceStates"
Global status bits (S:7)	The status bits contain a fault flag for each of the 8 connectable extension racks. The SIMATIC processors don't need these flags, since they are configured purely centralized with no extension racks. Decentralized rack faults are read out using the above mentioned system function "DeviceState".
Program scan times (S:8; S:9)	The PLC-5 processors store the last (S:8) and the maximum (S:9) program scan times in the respective status register word. In the SIMATIC controllers these information can be read on demand using the "RD_SINFO" system function.
Minor/ Major faults (S:10; S:11; S:17) and Fault codes (S:12)	During user code execution different faults can occur. The PLC-5 processors store information about them in a field of bits. A set bit in one the two bit field means a specific error, either minor (S:10; S:17) or major (S:11). In case of a major fault a fault code is stored in the S:12 status word. A diagnostic buffer contains the last entries (up to 100) in a retentive ring buffer inside the SIMATIC S7 controllers
Fault location (S:13; S:14)	The fault location for software related errors is stored as program file number (S:13) and rung number (S:14). The SIMATIC controllers store this kind of information as part of the diagnostic buffer, which can be read out using a system function.
Controller Wall Clock (S:18 – S:23)	The PLC-5 controller's real time clock is being stored as part of the status file. The words 18 (year), 19 (month), 20 (day) contain the date information, while the following words 21 (hour), 22 (minute) and 23 (seconds) contain the time of the day.
Indexed addressing offset (S:24)	Status word 24 contains the current address index used for indexed addressing in the PLC-5 processors. SIMATIC processors can use any globally accessible variable or constant as index for indexed addressing.

5 Converting the Data

5.2 Converting the PLC-5 Status file (S)

PLC-5 Status data	SIMATIC realization
	Thus any access to S:24 in the program to convert should be replaced appropriately.
Rack control bits (S:27)	Setting any of the bits 0-7 inhibits the represented rack 0-7 of the PLC-5 controller system. Setting the bits 8-15 acknowledges a fault in the respective rack 0-7. The SIMATIC S7 controller does realize the same functionality using a system function for decentralized racks only. In a centralized configuration, this functionality is not necessary and therefore not available.
Program watchdog setpoint (S:28)	In the PLC-5 processors the watchdog setpoint can be adjusted in the status file word 28. The SIMATIC doesn't allow a permanent modification of this watchdog during runtime. However retriggering the watchdog is possible as a result of an overlap (watchdog time exceeded error), due to prolonged program execution.
Fault routine file (S:29)	In this status word the PLC-5 processors keep the number of a program file, which will be executed whenever a fault occurs. Inside that routine an engineer can program logic to identify the fault. The SIMATIC processors provide multiple sources for interrupts allowing an engineer to reduce the logic inside the different fault routines. Different fault sources cause the controller to execute different OB's. Traditional these OBs have the number 80 and sequential numbers till 87.
STI status data (S:30; S:31)	In the status word 30 the PLC-5 processor keeps the interrupt time interval in which the program file identified by the content status word 31 is being executed. A similar behavior can be emulated using a "Delayed Interrupt" triggering an Organization Block (OB20) in the SIMATIC S7. However the more straight forward conversion would be the use of cyclic interrupt OBs, such as OB35. The time is adjustable using the OB properties within the S7-1500.
Global status and rack fault bits (S:32 – S:35)	Each of the extension racks has an assigned bit in alternating status words. The even status word (S:32;S:34) represents its status. The odd numbered status words (S:33; S:35) have the same functions in S:27
DII/ PII status data (S:46)	With a number placed in the status file word 46 an engineer indicated the execution of that program file, whenever an input interrupt occurred. In the SIMATIC world you may convert this behavior using Process Interrupt event Organization Blocks (OB40). Another method is to synchronize the update of periphery data with the program using them. For this to use, you configure and program an OB60 organization block and set up isochronous Fieldbus operation for PROFIBUS or PROFINET.
User Program checksum (S:57)	The PLC-5 processors store the user programs checksum in this status file word. This allows an engineer to check for program modifications comparing that checksum to a provided value. The same functionality can be realized for the whole program in the SIMATIC S7 controllers, however modification prevention is typically realized assigning passwords to the controller preventing unauthorized modifications.
MCP status data (S:80 – S:127)	The PLC-5 processor supports up to 16 main control programs (MCP). Each MCP uses 3 words of status data. The Status words 80-127 contain information about each MCP, such as used program file number, scan time and maximum scan time. Since the SIMATIC S7 does not maintain a status file globally, this information is provided in a locally accessible block of temporary variables. An engineer may transfer this information into a globally accessible data block for compatibility purposes or retrieve this information using a system function ("RD_SINFO") when necessary.

5.3 Converting the Binary (B) file type

A Binary (B) file is best translated by converting its 16-bit values into an array of 16-bit values. Operations within the SIMATIC S7 controllers take into consideration operand length, and as such, all operations used in the PLC-5 will work just as expected, including Shift and Rotate operations.

5.4 Converting the Timer (T) file type

Timers in the PLC-5 processors consist of a 16-bit preset value, a 16-bit accumulator value, and a time base of 1s or 10ms. The SIMATIC S7 uses IEC Timers, which have a 1ms time base and use a 32-bit preset and accumulator value.

The conversion requires proper scaling of the variable values in relation to the new time base.

Figure 5-2 – PLC-5 Timer with T4 data file

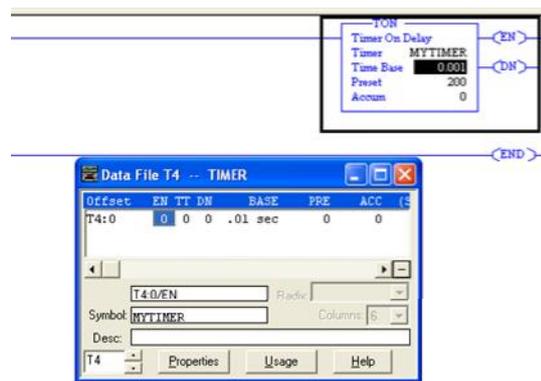
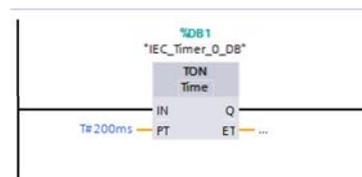


Figure 5-3 – TIA Portal Single Instance Timer



The most suitable conversion of the Timer file type is most likely the conversion into an array of Timers contained in a data block (DB). The available Mnemonics are different from those used in the PLC-5.

Table 5-4 – PLC-5 Timer file type conversion to SIMATIC

PLC-5 Mnemonic	Description	SIMATIC S7 Mnemonic
EN	Enable: When set to true the Timer starts running accumulating the time it runs	I N
TT	Timing: True when the timer is timing.	Doesn't exist, but can be created as: I N AND NOT Q
DN	Done: True when the accumulated timer value is equal to the preset value	Q
FS	First Scan: True when the timer executes for the first scan, providing a start pulse.	Doesn't exist, but can be created as: PositiveEdge(I N)

5 Converting the Data

5.4 Converting the Timer (T) file type

PLC-5 Mnemonic	Description	SIMATIC S7 Mnemonic
LS	Last Scan: True when the timer executes for the last scan, providing a stop pulse.	Doesn't exist, but can be created as: PositiveEdge(0)
OV	Overflow: True when the accumulated value becomes negative due to an overflow.	n.a.
ER	Error: True when either the accumulated or preset value are negative.	n.a.
PRE	Preset: Value: contains the time value the timer should accumulate.	PT
ACC	Accumulator Value: contains the accumulated time	ET

The closest one-to-one conversion of a PLC-5 Timer file, would be to create a Data Block containing an array of IEC Timer instances, however it would be preferred and more flexible to generate single instances DB for each of the timers.

5.5 Converting the Counter (C) file type

Similar to the Timer (T) file type the Counter (C) file type does not exist in the SIMATIC world, however SIMATIC controllers support IEC counters. The same applies here, as to the Timers, where the straight forward conversion would be an array with counter status values. The best conversion is probably the use of a single instance DB whenever a counter is needed. In this way a fully symbolic programming can be utilized.

The mnemonics of the IEC counter differ from the mnemonics of the PLC-5 counter.

Figure 5-4 – PLC-5 mnemonic conversion to SIMATIC

PLC-5 Mnemonic	Description	SIMATIC S7 Mnemonic
CU	Count Up: When set to true the counter increments its accumulator value	CU
CD	Count Down: When set to true the counter decrements its accumulator value	CD
DN	Done: True when the accumulator value is equal to the preset value	Doesn't exist, but can be created as: PV = CV
OV	Overflow: True when an overflow in the accumulator value occurred.	QU
UN	Underflow: True when an underflow in the accumulator value occurred.	QD
UA	Update accumulator :	Doesn't exist
PRE	Preset value: Contains the preset value the accumulator value should reach.	PV
ACC	Accumulator value: Contains the currently accumulated counter value	CV

The counting range is the same as for the SIMATIC S7, when using either DINT or DUINT as counter type.

5 Converting the Data

5.6 Converting the Control (R) file type

Figure 5-4 – PLC-5 Counter file type

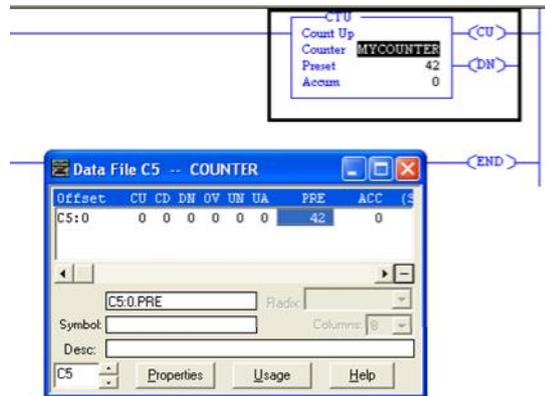
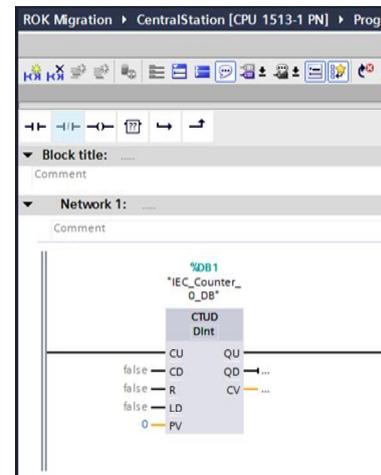


Figure 5-5 – TIA Portal IEC Counter



The access to a Counter in the PLC-5 uses the syntax C5:4, meaning the 5th (zero based counting) counter object in data file 5 of type Counter. In the SIMATIC S7 world the IEC counters are addressed by their symbolic representation, such as "MyCounter1". The individual mnemonics can be accessed using the "dot" (.) operator, such as "MyCounter1.Done" or similar. This applies to input parameters as well as to output parameters.

5.6 Converting the Control (R) file type

In the PLC-5 processors the each object in a Control (R) file consists of three 32-bit DINT words. The first couple bits represent status information, while the last two 32-bit DINT words contain length and current position information. This kind of information is typically used with instructions, which are in need for internal data to be kept persistent.

The SIMATIC S7 has the ability to store information locally also in a persistent manner, thus not needing globally stored control file objects. The proper conversion of such data structure would be the move of them into the local storage of the appropriate SIMATIC S7 instructions.

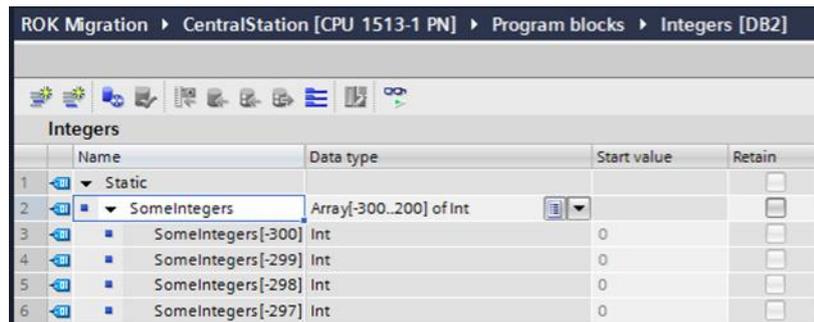
5.7 Converting the Control (R) type for Serial Port Control

When a Control file object is used in a serial port instruction, the proper conversion to SIMATIC S7 is automatically done, when replacing the serial port control instructions with the appropriate SIMATIC S7 instructions. These instructions handle the control internally.

5.8 Converting the Integer (N) file type

The most appropriate conversion of an Integer data file type would be the creation of a data block containing a one dimensional array of integers with as many elements as the original file contained. However the SIMATIC S7 can mix and match any data type into a single data block, as long as the maximum size of a data block is not exceeded.

Figure 5-6 – TIA Portal Array of Integer



The screenshot shows the TIA Portal interface for a data block named 'Integers [DB2]'. The table below represents the data block structure:

	Name	Data type	Start value	Retain
1	Static			<input type="checkbox"/>
2	SomeIntegers	Array[-300..200] of Int		<input type="checkbox"/>
3	SomeIntegers[-300]	Int	0	<input type="checkbox"/>
4	SomeIntegers[-299]	Int	0	<input type="checkbox"/>
5	SomeIntegers[-298]	Int	0	<input type="checkbox"/>
6	SomeIntegers[-297]	Int	0	<input type="checkbox"/>

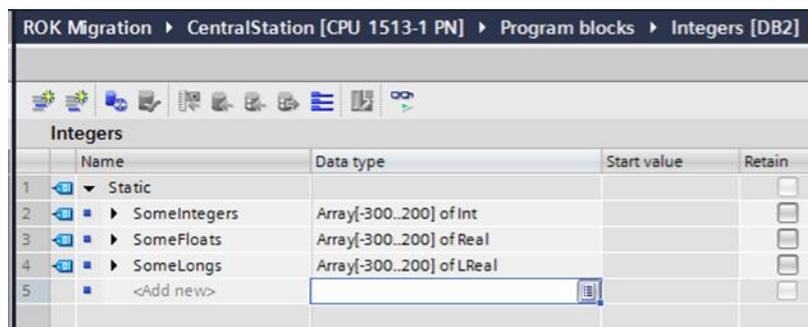
5.9 Converting the Floating Point (F) file type

What applied to the Integer type file applies to the Floating Point file type. A one dimensional array of REAL variables in a data block would best approximate the PLC-5 architecture.

The SIMATIC S7 systems support the single precision (32-bit) floating point format according to

ANSI/IEEE Standard 754-1985; IEC-60559:1989 - International version and additionally to that the double precision (64-bit) floating point numbers on SIMATIC S7-1200 and S7-1500 series processors.

Figure 5-7 – TIA Portal Array of Real and LReal



The screenshot shows the TIA Portal interface for a data block named 'Integers [DB2]'. The table below represents the data block structure:

	Name	Data type	Start value	Retain
1	Static			<input type="checkbox"/>
2	SomeIntegers	Array[-300..200] of Int		<input type="checkbox"/>
3	SomeFloats	Array[-300..200] of Real		<input type="checkbox"/>
4	SomeLongs	Array[-300..200] of LReal		<input type="checkbox"/>
5	<Add new>			<input type="checkbox"/>

5.10 Converting the ASCII (A) file type

On the PLC-5 processors each character stored in an ASCII file occupies 8-bit (BYTE) or ½ of a word. However the minimum memory size allocated is a word, thus allowing the storage of two characters. The straight forward conversion would be an array of word in the SIMATIC S7, providing the same addressing schema as the PLC-5 controllers have. The SIMATIC S7 also supports strings adding length and maximum length information to it, but it can also handle an array of Char.

5.11 Converting the Decimal (D) file type

On the PLC-5 systems a decimal type file contains so called binary coded decimals (BCD) numbers. A straight forward conversion would be an array of words with a maximum value of '9999'. Since the SIMATIC S7 doesn't know the data type "Decimal" the conversion into a 16-bit word or a 32-bit double word seems useful. The conversion from a BCD number into an integer and back is supported by the means of instructions operating on those kinds of values.

5.12 Converting the Block Transfer (BT) file type

The BT file type only appears in 6200 Legacy files using the *.PC5 extension and is therefore irrelevant for the converted program.

The conversion itself needs to address two cases, where the legacy instructions have been used. The first scenario is the transfer of larger chunks of data in a consistent manner to and from peripheral modules. In the second scenario configuration and/ or diagnostic data have been transferred between the controller and its periphery.

The SIMATIC S7 use for the different scenarios different system functions. In the first scenario, where the exchange of peripheral data is the focus, the SIMATIC S7 uses either a fully symbolic access and let the controller's operating system handle the consistencies or system functions named "DP_RDDAT" and "DP_WRDAT".

In the second scenario focusing on diagnostic- and configuration data, the SIMATIC S7 uses the system functions "RD_REC" and "WR_REC".

5.13 Converting the M0 and M1 file types

These files are being used in conjunction with a transfer of bulk data to and from an intelligent module to the processor. The use of the M0x or M1x files is dramatically obsoleted and has been replaced with the previously mentioned BTR and BTW instructions of the PLC-5 processors.

The conversion of such files into the SIMATIC S7 will most likely lead to the use of the BTR/ BTW equivalents "DPRDDAT" and "DPWRDAT".

5.14 Converting the Message (MG) file type

An MG file may only be found in 6200 Legacy files using the *.PC5 extension.

It contains information about messages being sent or received to or from other communication partners. For each of the possible communication partners a message type has been defined.

To convert this kind of communication, you should take notes on the configuration settings being used, such as IP address and port number of the remote partner.

The SIMATIC S7 doesn't need such specification of message types. Messages can be sent easily using straight forward Send/ Recv mechanisms or even simpler Put and Get instructions. The configuration is done fully graphical. Additional information or information unknown to the project can be entered easily into the configuration masks provided by TIA Portal.

5.15 Converting the PID (PD) file type

In the PLC-5 processors each PID closed loop uses one structure within this data file. With the SIMATIC family this conversion would easiest translate into the same number of PID instruction calls as there have been elements in the original PD data file. The data a no longer stored globally, they are stored locally with the PID control loop instance.

Although the PID instruction of the PLC-5 processors are convertible into the SIMATIC S7 system, it is recommended to thoroughly look over and correct, when necessary, the parameter assignments.

5.16 Converting the SFC status (SC) file type

The SFC status file, is a collection of four-word structures that store control and status information for sequential function charts within the PLC 5 program. In the SIMATIC S7 world, this control and status information is stored within an instance DB for each sequencer.

The programming language SFC uses a four 16-bit word structure to store control and status information about the sequential function chart. In the SIMATIC S7 world a SFC like language called GRAPH exists, which is also certified according to IEC 61131-3 SFC. Its graphical programming is very similar to the language elements used by the PLC-5 processors. A conversion is a manual, yet easy process. The largest difference between the two systems in regards to SFCs is the storage of the sequencer control and status information in a globally accessible data block (Simatic), instead of local storage within the sequencer in PLC5 programs.

5.17 Converting the ASCII String (ST) file type

As already mentioned with the ASCII file type the SIMATIC S7 also supports strings. However the maximum string length in a SIMATIC S7 is limited 255 characters, not 82 as with the PLC-5 processors.

5.18 Converting the ControlNet (CT) file type

This data file type only appears in 6200 Legacy files using the *.PC5 extension and is used for holding ControlNet communication information. ControlNet is not supported by Simatic S7 controllers and the conversion of these data files is irrelevant. The integration of peripheral devices needs to take place with the use of a gateway to ControlNet or the peripherals need to be upgraded/replaced with those utilizing a supported network.

5.19 Converting the Constant Values

The PLC-5 uses different format strings for constants than the SIMATIC S7 and thus it is important to convert these constant values. The differences in the mnemonics and the procedure converting them are provided in the following table.

Table 5-5 – PLC-5 constant conversions to SIMATIC

PLC-5 Mnemonic	Difference	SIMATIC Mnemonic
&N49 -49	Integer numbers are being converted by removing the “&N”, if present and copying the remainder of the constant	49 -49
&B00110001	Binary numbers are converted by replacing the “&B” by “2#” and copy the remainder of the constant	2#00110001
&A1 &Amx	H ASCII Codes are being converted by removing the “&A” and put the remainder in quotation marks	'1' 'mx'
&H0041 0x0042 0X0043	Hexadecimal numbers are converted by replacing the “&H”, “0x” or the “0X” with a “16#”. You may also add the data type width as “B#16#” for Byte (8-bit) “W#16#” for Word (16-bit) “DW#16#” for Double Word (32-bit) width. Leading zeros can be omitted.	16#0041 W#16#42 DW#16#43
&D0049	Binary Coded Decimal numbers are not directly supported and need to be stored as Hexadecimal values. The conversion is accomplished by simply replace the “&D” with a “16#”. It is then necessary for the programmer to remember that these are BCD values and any operations on them need to be reviewed. Alternatively, the values can be converted to hexadecimal numbers for simplified use in operations.	16#0049
&O61	Octal numbers are converted by replacing the “&O” with “8#”. The optional specification of the width is also possible.	8#61 B#8#61
-12.34E-12 3.14	Floating Point numbers will be converted normalized and represented as such.	-1.234E-11 3.14E+0

5.20 Converting Indirect Addresses

Indirect addressing is when a part of an address is replaced with a reference to another address. The final address is unknown at compile time, meaning the final access address is being calculated during runtime. The PLC-5 processors can use an address reference to define these address parts

- File number
- Word or element number
- Bit number (only for B type addresses)

The conversion into a SIMATIC S7 program needs to translate the functionality into the different system paradigm. A system, as the SIMATIC S7, supporting full symbolic programming is not intended to support indirect addressing.

However with the SIMATIC S7-1200 and S7-1500 controllers an engineer has the ability to realize such flexible programs as have been realized using indirect addressing.

Figure 5-8 – RSLogix5 indirectly addressing a data file

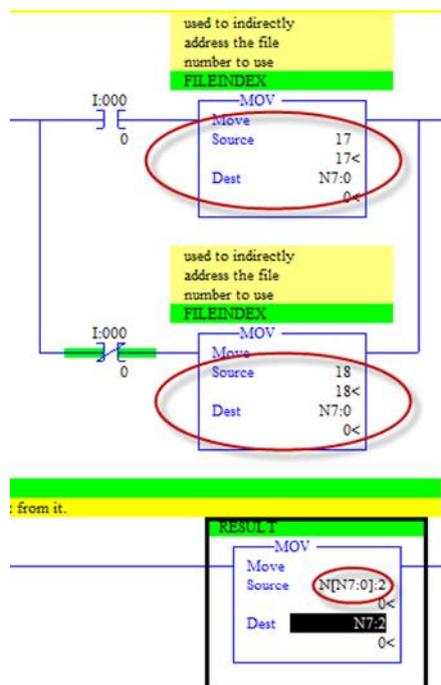
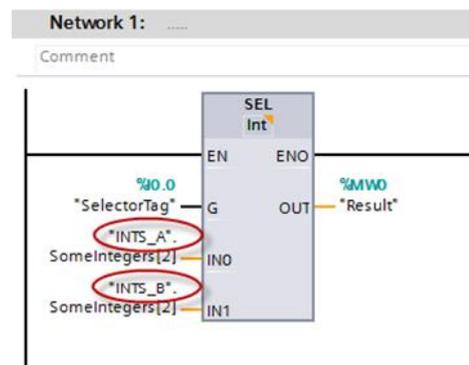


Figure 5-9 – TIA Portal selecting the data file



Indirect addressing on a PLC-5 controller can be translated, depending on the realized function, into SIMATIC S7 controllers using different methods. One is shown above another has been mentioned already and is called "Slicing".

5.21 Converting Index Addresses

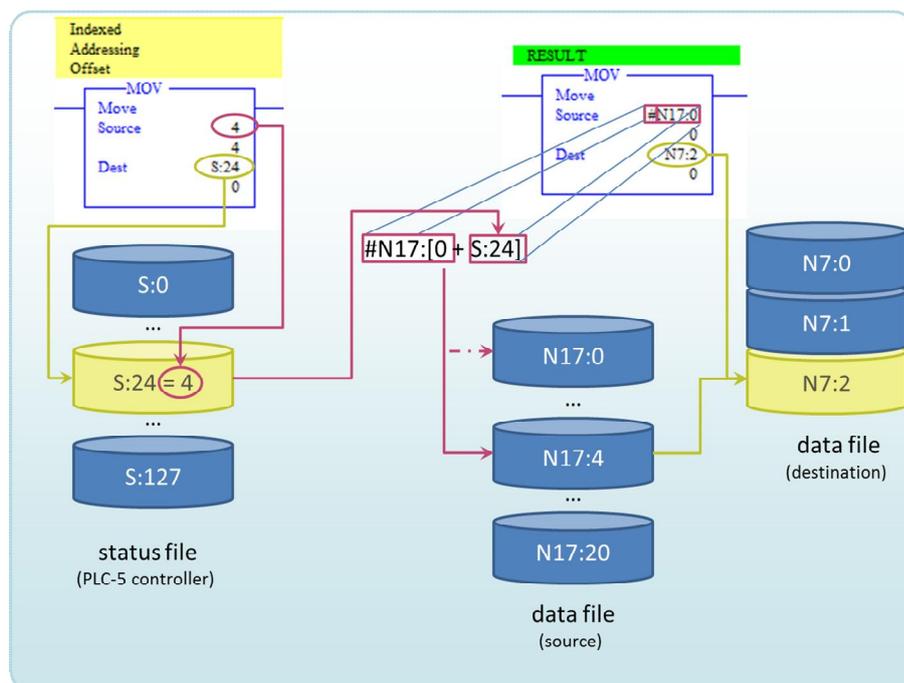
Indexed addresses in the PLC-5 processor are indicated by a preceding # character. The PLC-5 processors differentiate two methods:

- Processor status word S:24 controlled
- Control structure controlled addresses

Processor status word S:24 controlled

In this case the processor status word S:24 contains the current index value to add to an address reference to be used for reading or writing. This behavior is triggered by putting the '#' in front of the base address.

Figure 5-10 – PLC-5 indexed addressing using S:24



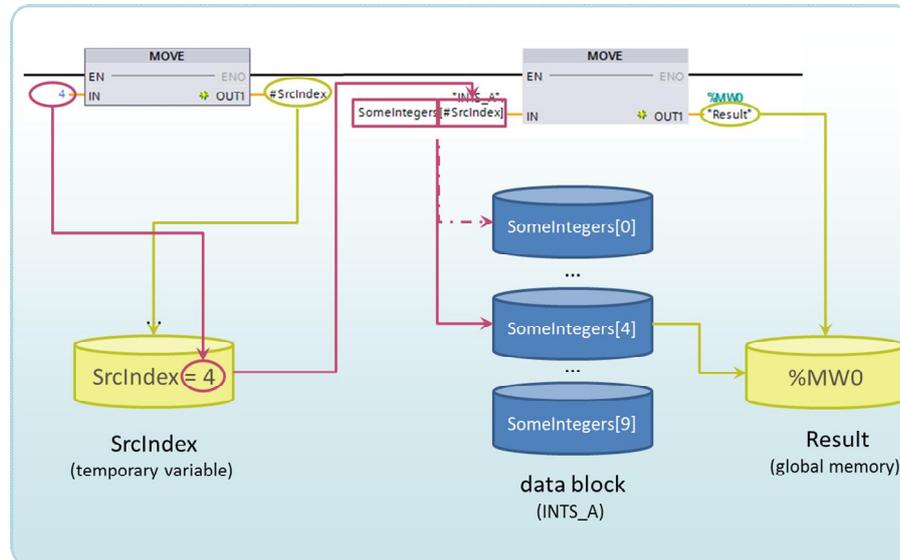
This kind of addressing is transferable under certain circumstances to a SIMATIC S7 controller. As with indirect addressing the method being used to transfer this to the SIMATIC depends on the functionality.

5 Converting the Data

5.21 Converting Index Addresses

Best translation into the SIMATIC S7 controller is the iteration through the source data, which should be organized in an array of the specific type.

Figure 5-11 – SIMATIC S7 controller indexing an array



Control structure controlled addresses

Indexed addresses are also used with the file instructions to operate on files of data. These instructions use a CONTROL structure to determine the index value; the current position.

The CONTROL structure is no longer necessary, since the SIMATIC S7 stores such values internally to the instruction instance working on the data in the data blocks. Users may also recreate those control structures manually, however this is not recommended.

6 Conversion of System Software and Standard Functions

6.1 Introduction

This chapter lists the more commonly used PLC-5 instructions, explains how the equivalent is done in SIMATIC S7 and provides several specific examples

Table 6-1 – Topics of chapter “Conversion of System Software and Standard Functions”

Topic	Page
System Functions	49
Copy Instructions	50
Reading and Setting System Date and Time	53
Interrupt Handling	55
Error and Fault Handling	56
Diagnostics	56
Timers and Counters	57
Conversion Routines	57
Closed Loop Control	59
Examples of System Function Calls	60

The purpose of this chapter is to make you aware of the dedicated instructions and system functions available in the SIMATIC S7, so you do not waste time developing solutions that already exist.

6.2 System Functions

Compared to the limited instructions set of the PLC-5 processors, the SIMATIC S7 has a wide variety of system functions built-in to the firmware in addition to the regular instructions set. This way access to sensitive system data can be provided in a controlled manner. Many functions, which may have to be implemented manually on the PLC-5 systems, are readily available on the SIMATIC S7 system.

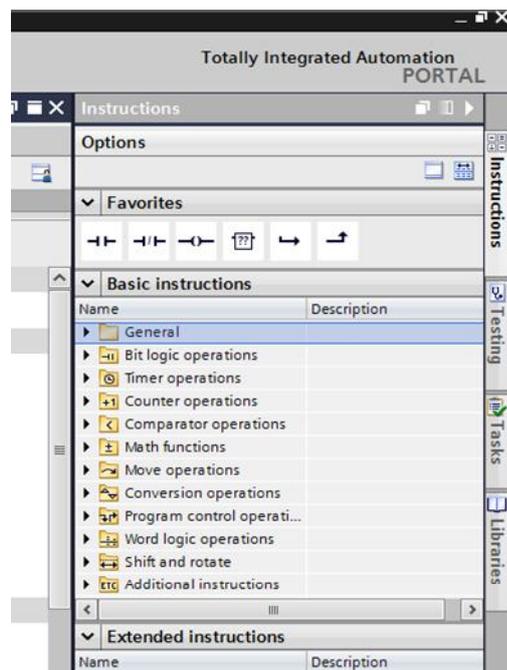
Compared to the limited instructions set of the PLC-5 processors, the SIMATIC S7 has a wide variety of system functions built-in to the firmware in addition to the regular instructions set. This way access to sensitive system data can be provided in a controlled manner. Many functions, which may have to be implemented In TIA Portal, the SIMATIC engineering framework, these system functions are contained in the function library and can be used in the engineer's code. The library is split up into a few folders:

- Basic Instructions
- Extended Instructions
- Technology Instructions
- Communication

The folders itself provide access to the instructions and a grouped based on their functionality.

With the SIMATIC S7 there is no longer a difference between user defined functions and system provided functions. However the performance and memory usage is optimized on the system provided functions.

Figure 6-1 – TIA Portal Global Library Basic instructions



6.3 Copy Instructions

The PLC-5 has a universal Copy instruction called “COP”, which does not use any status bits. It copies the source data into the destination area. The length always indicates the number of elements to copy. If the source elements are Integer numbers, it will copy as many 16-bit words as identified by the Length parameter. It is typically used with complex data structures not with simple values, tags or immediate values. For simple values, the PLC-5 “MOV” instruction is more commonly used.

The copy instructions are an essential part of a program. In the SIMATIC S7 world they belong to the Basic Instructions and can be found there.

The most straight forward conversion for the PLC-5 copy instruction is likely the “MOVE_BLK” instruction. However it may not be always the best translation. Depending on the data type and the data length optimized instructions may be available.

Indexed addressing and indirect addressing is possible with “MOVE_BLK” instruction. However the data types of source and destination have to match. There is no conversion. The number of elements copied with “MOVE_BLK” must be provided at the input parameter “COUNT”. This parameter always counts the number of elements to be copied, independent of the size of each element. In this regard the PLC-5 “COP” instruction is identical to the “MOVE_BLK” instruction of the SIMATIC S7.

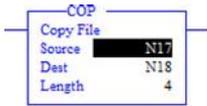
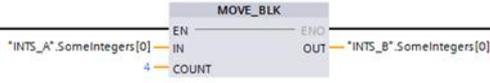
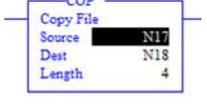
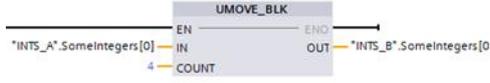
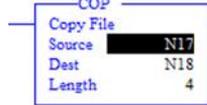
Figure 6-2 – TIA Portal Global Library - Move instructions

Name	Description	Version
Basic instructions		
General		
Bit logic operations		
Timer operations		
Counter operations		
Comparator operations		
Math functions		
Move operations		
MOVE	Move value	
FieldRead	Read field	
FieldWrite	Write field	
MOVE_BLK	Move block	
MOVE_BLK_VARIANT	Move block	
UMOVE_BLK	Move block uninterrupt...	
FILL_BLK	Fill block	
UFILL_BLK	Fill block uninterruptible	
SWAP	Swap	
ARRAY DB instructions		
ReadFromArrayDB	Read from the ARRAY d...	
WriteToArrayDB	Write to the ARRAY dat...	
ReadFromArrayDBL	Read from the ARRAY d...	
WriteToArrayDBL	Write to ARRAY data bl...	
Others		
Conversion operations		

6.3 Copy Instructions

In general a conversion of PLC-5's "COP" instruction should be realized as shown in the table below.

Table 6-2 – Conversion of PLC-5 copy instructions

PLC-5	S7 Instruction	Usage
		<p>When copying complex data structures</p> <ul style="list-style-type: none"> Length = Count <p>*) Source and destination data types need to match</p>
		<p>When copying complex data structures, and interrupt routines could modify transferred data.</p> <ul style="list-style-type: none"> Length = Count
<p>RESULT</p> 		<p>When copying simple values, tags or immediate values</p> <ul style="list-style-type: none"> Length = Count <p>*) Automatic data type conversion from source to destination, without loss where possible</p>
		

In the SIMATIC S7 controllers a data type conversion takes place, without loss, when it is possible. The PLC-5 mnemonic "Length" of the "COP" instruction is identical to the SIMATIC S7's mnemonic "Count" of both the "UMOVE_BLK" and the "MOVE_BLK".

6.4 Reading and Setting System Date and Time

The SIMATIC S7 has a Real Time Clock which keeps the current system date and time internally. In order to set or read the RTC you need to use system functions (SFC).

The engineer will find these instructions in the global library under the Extended Instructions.

Here are functions available for reading both:

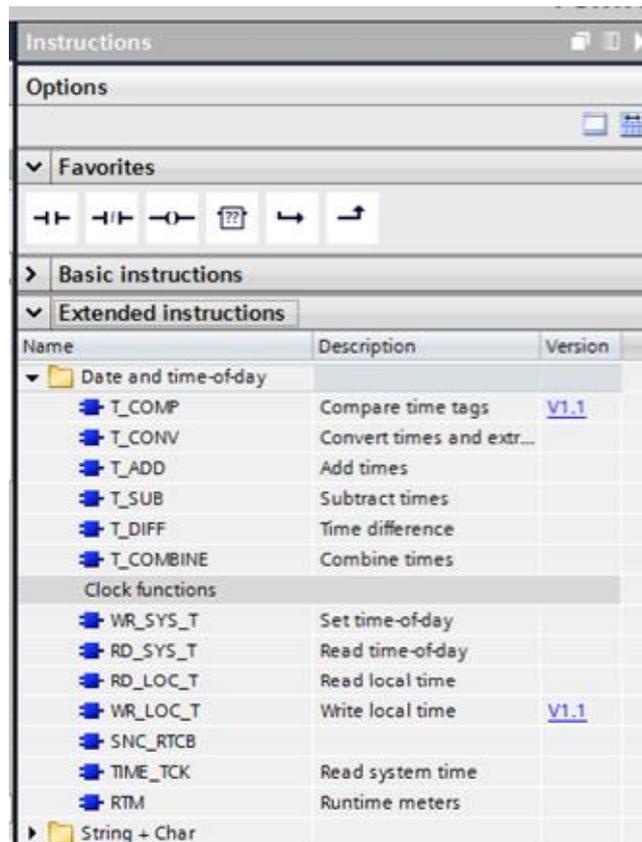
- The local time (RD_LOC_T)
- The time of day (RD_SYS_T)

There are also functions for setting these values:

- The local time (WR_LOC_T)
- The time of day (WR_SYS_T)

Setting times and reading them is possible with different source and destination data types.

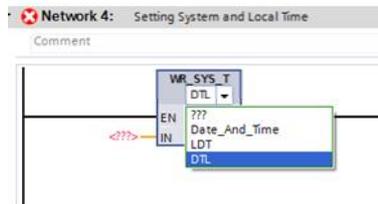
Figure 6-3 – TIA Portal - Global Library Extended instructions



6.4 Reading and Setting System Date and Time

The SIMATIC S7-1200 and S7-1500 support long versions of the DateAndTime data types, which can be selected easily using the DropDown list.

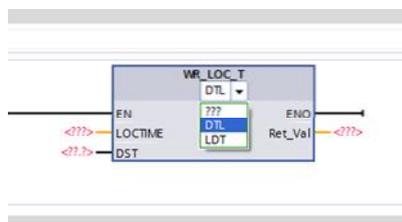
Figure 6-4 – TIA Portal - writing system time



The SIMATIC S7 supports the regular Date_And_Time data type as input as well as the Long_Date_And_Time (LDT) and Date_and_Time_Long (DTL). The difference in the Long versions is explained as follows. While the LDT data type counts the nanoseconds since midnight of January 1st 1970 the DTL data type stores the date and time in a structured way including the nanoseconds since midnight that day.

Writing the local time allows you to use the long versions only and adds the possibility to define the provided time as Daylight Savings Time (DST).

Figure 6-5 – TIA Portal - writing local time



The nanoseconds can be retrieved using the provided count in the DTL data type of the reading instruction.

Figure 6-6 – TIA Portal - reading system time in ns (S7-1200 and S7-1500 only)



With this little example you are able to access every member of the DTL structure, such as Day, Month and Year as well as Hour, Minute, Second and Nanoseconds. In the PLC-5 processors you access the processor's status file words S:18 to S:23 to get or set the date and time of the processor. Setting the time is limited to seconds in accuracy.

6.5 Interrupt Handling

In the PLC-5 processors it is possible to enable or disable interrupts of the current and higher priorities. This way it is possible to create an almost atomic operation ensuring consistency.

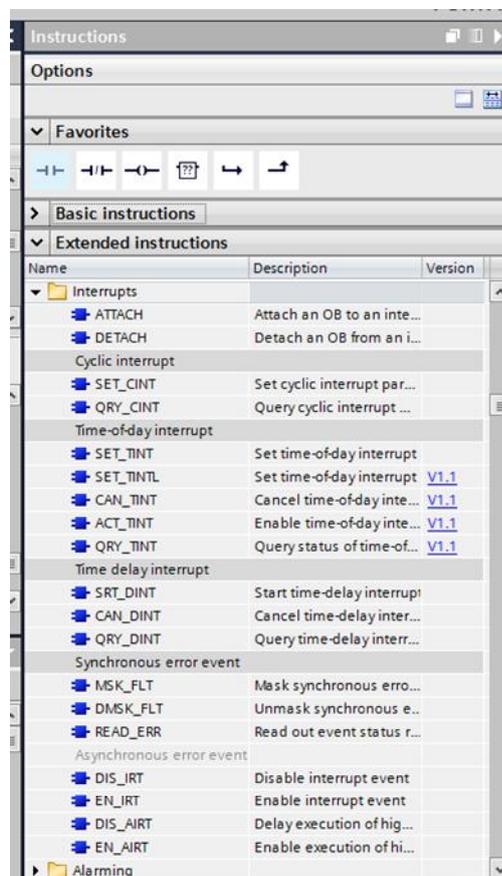
The instructions of the SIMATIC related to Interrupt handling are located in the Extended instructions section of the global library of the TIA Portal.

The instructions ATTACH and DETACH allow you to dynamically change the assignment of an OB to a certain interrupt source. In the PLC-5 processors the access of the processor status word S:46 writing the program file number responsible for PII is a similar behavior. However the SIMATIC S7 supports multiple such event handlers, which will be executed one after the other.

With SET_CINT the SIMATIC S7 allows, similar to the PLC-5 status word S:53, to set the time interval of a specific cyclic interrupt.

The reader should also be aware, that SIMATIC S7 systems provide Time-of-Day interrupts, which are not found in PLC-5 Systems. This interrupt source allows a scheduled execution at a specific time of any day, such as shutting down for maintenance every last day in a month or similar.

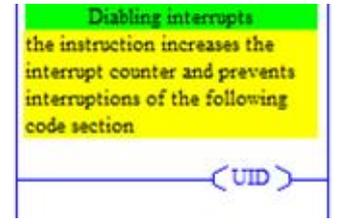
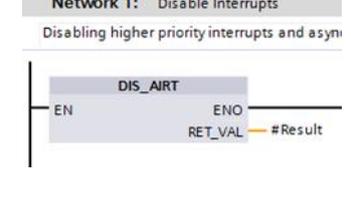
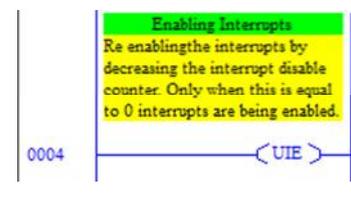
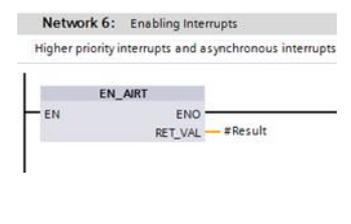
Figure 6-7 – TIA Portal Global Library Interrupt instructions



6.6 Error- and Fault Handling

The UID and UIE instructions of the PLC-5 processors are best converted into DIS_AIRT and EN_AIRT instructions on the SIMATIC S7.

Table 6-3 – Conversion of PLC-5's UID and UIE instruction

PLC-5	S7 Instruction	Usage
		Disabling interrupts prevents interruptions of sections of code. The purpose is mainly to maintain data consistency.
		Re- Enabling interrupts allow future events to interrupt code execution until disabled again.

6.6 Error- and Fault Handling

The PLC-5 processors store an error codes in the processors status file at words S:11 for major faults and S:12 and S:17 for minor faults.

SIEMENS provides a result output on several instructions, which indicates success or other execution results. Besides that the instructions GET_ERROR and GET_ERR_ID provide detailed error codes on the current status of the processor. Each error will be stored in a buffer internal to the processor. The behavior is that of a ring buffer, meaning overwriting the oldest entry. The size can be parameterized with a default value of 100 entries.

6.7 Diagnostics

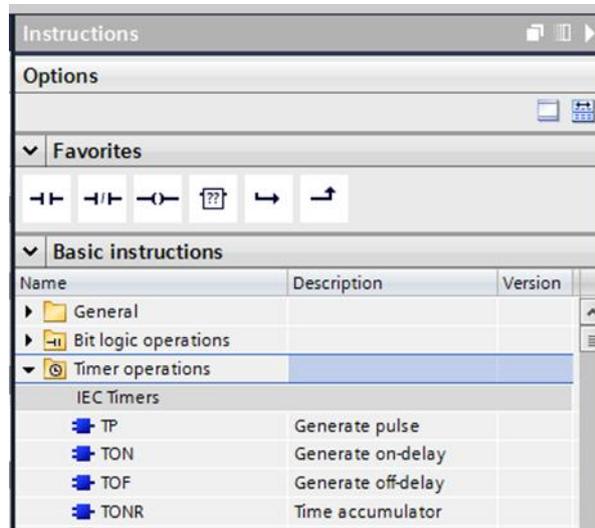
The PLC-5 provides more generic information about the health status of the system as part of the processors status file. The conversion of code reading such diagnostic information is typically not possible, since most information is not available in a SIMATIC S7.

Rather than reading the diagnostic information continuously, as the status file in the PLC-5, the SIMATIC S7 systems provide the occurrence of a diagnostic event as source for an interrupt. This allows the engineer to respond in case of such an event, removing continuous load from the processor.

6.8 Timers and Counters

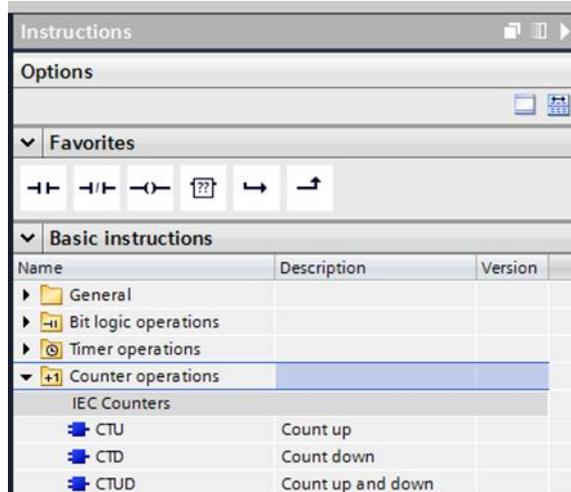
Timers on the PLC-5 have a different and adjustable time base than the timers on the SIMATIC S7 processors. The time base of the SIMATIC is standardized to the IEC 61131 to be fixed at 1ms.

Figure 6-8 – TIA Portal Global Library Timer instructions



You'll find the available timer instructions complete and easy to use. The parameters are mostly self-explanatory. For more a complete mapping of the PLC-5 mnemonics to SIMATIC, see chapter "[Converting the Data](#)".

Figure 6-9 – TIA Portal Global Library Counter instructions



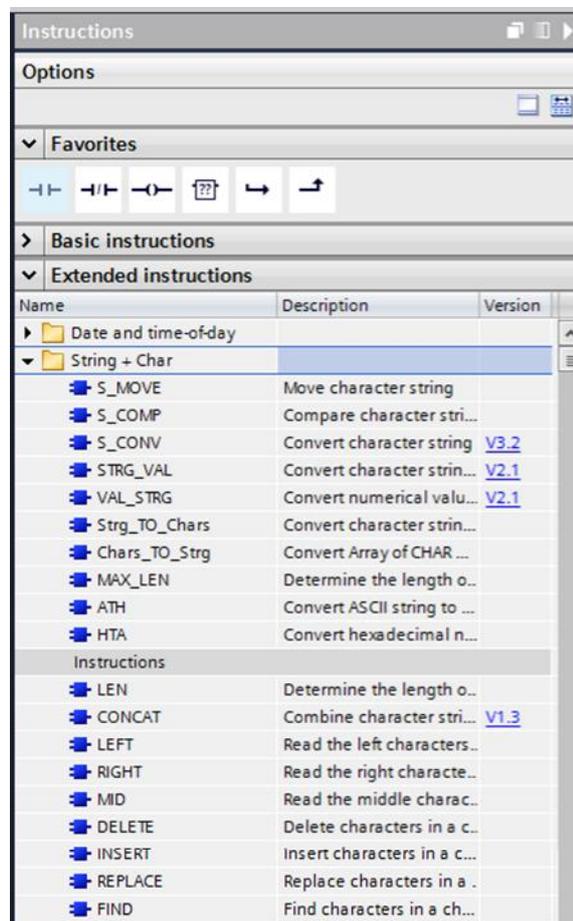
Again the instructions are complete and very similar to the PLC-5 counters. The behavior is identical. A complete parameter mapping to the SIMATIC mnemonic can be found in chapter "[Converting the Data](#)".

6.9 Conversion Routines

In contrast to the PLC-5 the SIMATIC S7 engineering framework TIA Portal maintains several conversion functions as library elements. This way the execution of such functions can be either a part of the compiler or the runtime system. This adds flexibility to the code and therefore lets the programmer keep focused on what he has to do not worrying about conversions.

However String to Number and vice versa conversions are conversions, which have to be done explicitly as some others where data loss may occur. The String conversion instructions can be found in the extended instructions section of the global library provided with TIA Portal.

Figure 6-10 – TIA Portal Global Library String Conversion instructions

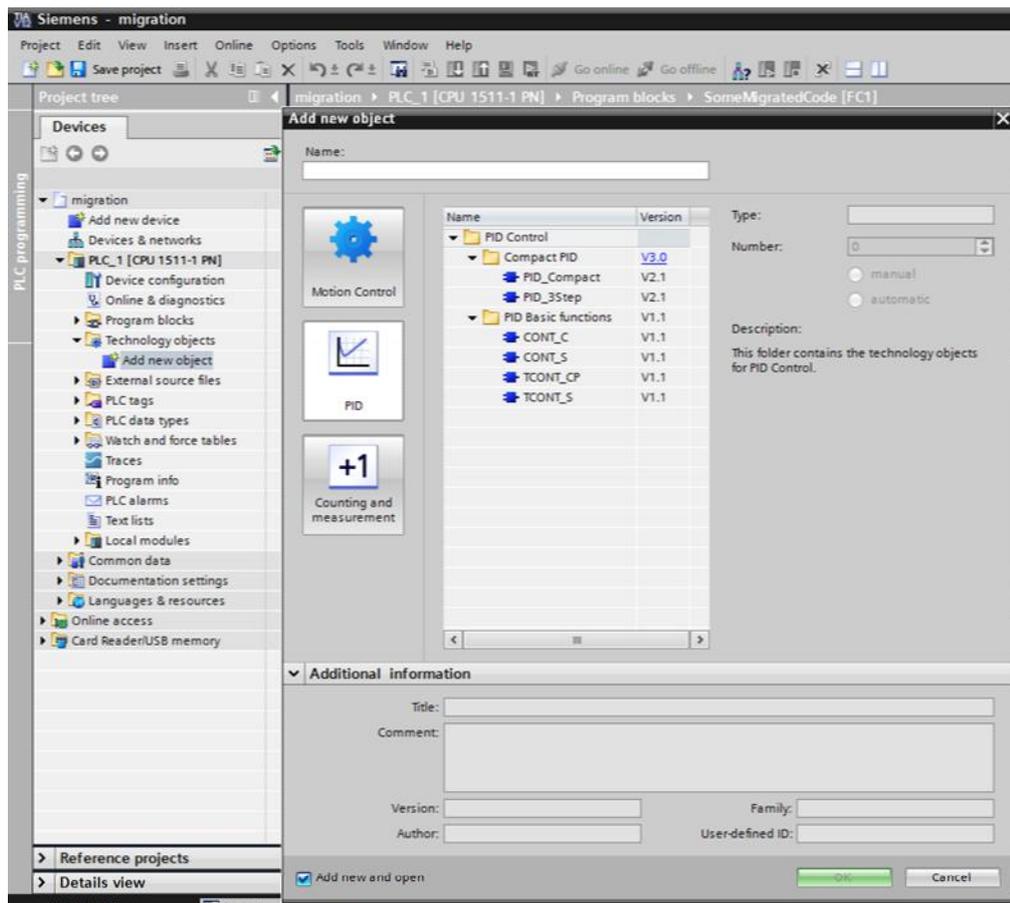


The engineer, who wants to convert his application to SIMATIC S7 may find the available instructions complete not only for conversion but also for string manipulation.

6.10 Closed Loop Control

Closed Loop control is handled completely different than the PLC-5 does it. An automatic conversion of a PID loop is not possible, and it needs to be re-engineered. That said, as PID Control is a task that is not always straightforward for the typical user, the TIA Portal realizes a wizard guiding you through the setup process. This speeds up commissioning and reduces or eliminates the risk for errors.

Figure 6-11 – TIA Portal TO PID Control



The whole PID controller is being realized as a “Technology Object” short TO. This allows the engineer to keep focused on the technological task and not on the surrounding program. Most of the PID parameters can be found in the setup masks of the TIA Portal.

6.11 Examples of System Function Calls

As system function calls are implicit part of the system architecture, they vary between the PLC-5 and S7 systems. Below you will find several examples intended primarily to illustrate the use of the S7 system functions compared to the PLC-5 equivalent.

Setting and Reading the Clock

Setting and Reading the Clock is one of the first things you do, when you start using a new system. For this reason it is listed here as the first example.

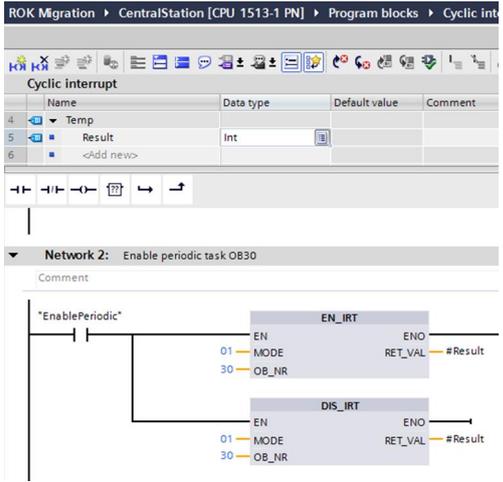
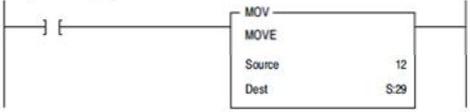
Table 6-4 – PLC-5 date and time structure & SIMATIC date and time structure

Rockwell RSLogix		SIEMENS TIA Portal	
The date and time values are stored in six 16-bit words in the PLC-5 processors status file.		The date and time values are stored in a eight byte sized data structure named "DATE_AND_TIME" or short "DT".	
The date time values are stored as shown in the following table.		The datatype is structured similar to the PLC-5 status file as shown below.	
Offset	Meaning	Offset	Meaning
S:18	year []	+ 0	Year [..]
S:19	month [1 .. 12]	+ 1	Month [1 .. 12]
S:20	day [1 .. 31]	+ 2	Day [1 .. 31]
S:21	hour [0 .. 59]	+ 3	Hour [0 ..23]
S:22	Minute [0 .. 59]	+ 4	Minute [0 .. 59]
S:23	seconds [0 .. 59]	+ 5	Seconds [0 .. 59]
		+ 6	Milliseconds [0 .. 99] (most significant 2 digits)
		+ 7	Milliseconds [0 .. 9] (least significant digit) Day of the week [0 = Sunday .. 6 = Saturday]
Read/ Write access Direct access is required to modify the current date and time. This may include invalid entries for date and time values, leading to miscalculations in user code.		Read/ Write access Each access to the system values is channeled through an instruction, which implicitly evaluates the passed date and time information. This ensures valid entries and avoids miscalculation in user code.	

Disabling and Enabling Interrupts

Under rare circumstances it is required to disable interrupts from various sources. This mostly applies to situations with prolonged data handling, where data consistency is essential.

Table 6-5 – Disabling and Enabling Interrupts

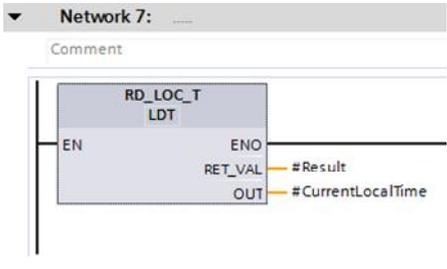
Rockwell RSLogix	SIEMENS TIA Portal
<p>The PLC-5 processors status file contains a 16-bit word S:29, which represents the program file to execute when an interrupt occurs. Interrupts are enabled by writing a program file's number into status file word S:29.</p>	<p>Figure 6-12 – TIA Portal enabling and disabling interrupts</p> 
	
<p>Disabling Interrupts isn't possible, since entering 0 or an invalid program file number, will cause the processor to enter fault mode on any interrupt. However the instruction "UID" disables user interrupts.</p>	

© Siemens AG 2015 All rights reserved

Reading System time

Reading the system time can be used to calculate execution times, giving an indication of processor loading or simply for displaying purposes.

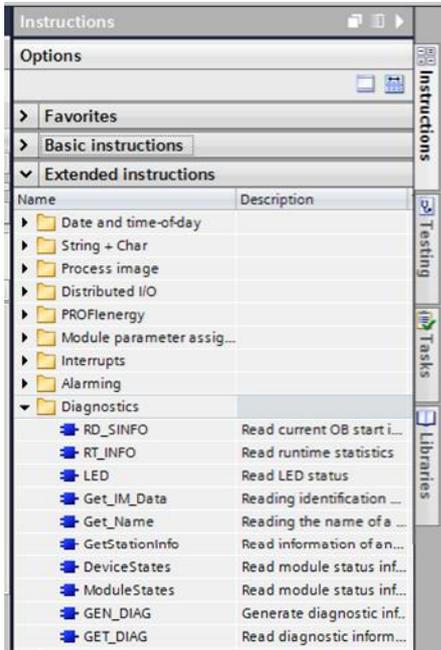
Table 6-6 – Reading System time

Rockwell RSLogix	SIEMENS TIA Portal
<p>PLC-5 processors provide no runtime or system time other than the system clock, which is held in status words S:18 - S:23 and provides at best a resolution in seconds..</p>	<p>Figure 6-13 – TIA Portal "RD_LOC_T"</p>  <p>The system function "RD_LOC_T" reads out the local time (incl. time zone adjustments and daylight savings) with resolution down to the nanosecond.</p>

Get Faults

No system or program runs error-free for its entire life. As such, it is important to be able to detect and identify faults as quickly as possible in order to prevent damage to man and machine. Fault and diagnostic handling in the controller is the first step to providing this information.

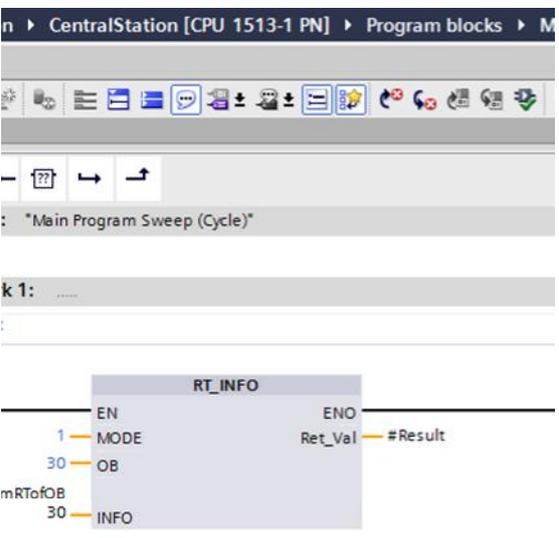
Table 6-7 – Get Faults

Rockwell RSLogix	SIEMENS TIA Portal
<p>In the PLC-5 processors the processor status file contains major fault bits in S:11, which indicate a specific type of major fault. The processor will interrupt the current running logic program to run the fault routine, when configured in S:29. The processors status file S:12 stores the fault code of the occurred fault.</p>	<p>The TIA Portal has a range of specific instructions available. They are geared towards a specific set of information related to a specific part of the system. This leads to optimized runtime and lower loading on the system.</p> <p>Figure 6-14 – TIA Portal Diagnostic functions</p>  <p>For a large number of use-cases, the handling of diagnostics information is no longer programmatically necessary, because they can be displayed directly on a compatible SIMATIC HMI or SCADA system, or directly from the PLC webserver</p>

Get Scan Time

The processors scan time affects the best possible response time. For this reason an engineer may be interested in knowing what the current or last scan time was. The scan time is defined as the time the processor requires to calculate the whole user program plus the system code. Under controlled circumstances the processor would have a stable scan time, since the executed code would be the same. In practice however input signals change their values, often cause branching in the program to occur, which varies the execution time. This leads to a varying scan time. With the SIMATIC S7 1500 there is a method to get more detailed runtime information out of the system.

Table 6-8 – Get Scan Time

Rockwell RSLogix	SIEMENS TIA Portal
<p>A Watchdog timer is being used to monitor the scan time to ensure it remains below a set threshold. The PLC-5 processors use the processors status file word S:28 for the watchdogs maximum value. Information about the maximum and the previous (last) program scan duration is stored in the processors status file at word S:8 (last program scan duration in ms) and S:9 (maximum program scan duration in ms)</p> <p>The last and maximum scan time of an STI or PII can be read out from the processors status file words S:53; S:54 (for STI) and S:54; S:55 (for PII). The MCPs carry their scan times in the status file words S:81; S:82 for MCP A.</p>	<p>Figure 6-15 – TIA Portal Reading RT statistics OB30</p>  <p>With the system function "RT_INFO" an engineer can read out may kinds of informative values, such as the processor loading caused by higher prior OB's. It allows also to retrieve the previous, maximum and minimum scan time for an OB, including the cyclic OB1.</p>

© Siemens AG 2015 All rights reserved

7 Common misunderstandings when converting to SIMATIC S7

7.1 Introduction

The objective of this section is to point out some of the design and programming misunderstandings Rockwell users typically have.

These misunderstandings come from misinformation and obsolete knowledge. The misunderstandings can be grouped into two categories:

- Software related
- Hardware related

These misunderstandings may lead to reduced control system performance, expandability and reusability.

The following section will address both categories and try to clarify these things.

7.2 Not Selecting Appropriate Hardware

This is a common problem during migration, as each system comes with particular caveats to their IO portfolio. As a result, the selection of components is often sub-optimal. In order to make the selection easier, you can refer to appendix A. Essential for a proper hardware selection is the understanding of the application. Specific functions may be realized using specialized modules in the PLC-5 system, while the SIMATIC S7 controller may be able to use standard modules instead.

7.3 The Impact of Task Scheduling

Whereas task scheduling was a significant consideration in PLC5 systems, as interrupts and the order of program execution had direct impact on cycle time and data consistency, with Simatic S7, these issues are largely transparent. Important to consider however are the increased sources of interrupt, both hardware, timed and diagnostic that are available in modern S7 Systems and their priority classes.

Please refer to the SIMATIC S7 controller manual for more details. **Not Selecting Appropriate Hardware**

This is a common problem during migration, as each system comes with particular caveats to their IO portfolio. As a result, the selection of components is often sub-optimal. In order to make the selection easier, you can refer to appendix A. Essential for a proper hardware selection is the understanding of the application. Specific functions may be realized using specialized modules in the PLC-5 system, while the SIMATIC S7 controller may be able to use standard modules instead.

[\(S7-1500 controller manual collection – Link ID: 86140384\)](#)

7.4 Drawbacks of straight code translation vs. conversion

While direct, line-by-line translation of PLC5 code into STEP7 can save time and reduce initial errors, one loses out on optimization potential that is available by re-programming the application with consideration of the architecture of target system.

By converting rather than translating your PLC-5 programs, you will make better use of the capabilities of your SIMATIC S7 system.

7.5 Not using the most appropriate Programming Languages

Especially for programs translated from PLC-5 the most appropriate language isn't always used. The PLC-5 processors only support Ladder Logic (LAD) and Sequential Function Chart (SFC). While these two languages can be used to solve nearly every automation task, more up-to-date languages such as SCL could be more efficient. When converting, the programmer should also consider other programming languages.

The SIMATIC S7 supports multiple programming languages, which are:

- LAD, most commonly used for binary logic
- Function Block Diagram (FBD), usable for a mix of binary logic and more complex function calls
- Statement List (IL), commonly used in STEP7 for advanced jobs
- GRAPH (SFC), commonly used for sequences of instructions
- Structured Control Language (ST), perfect for all kinds of jobs.

7.6 Implementation of Incorrect Data Types – INT vs. DINT

On the PLC-5 system, the data types are based on 16-bit words leading to a typically faster execution of code operating on 16-bit words rather than 32-bit double words.

With SIMATIC S7, the situation is somewhat different it is a 32-bit based system. In this case, 16-bit instructions are slower, since additional masking is required.

7.7 User Code Emulating Existing Instructions

Many programmers often “reinvent the wheel” by programming helper routines to do a certain job, although there are instructions available providing the functionality already. Most of the instructions provided by the libraries of TIA Portal are native instructions. This means that they represent internal code, native to the processor that yields faster execution and optimized memory use than user code.

With the possibility to encapsulate the instructions, a programmer is able to adopt the parameter assignment to their needs. This allows the engineer to provide functionality not available as built-in instruction or library element.

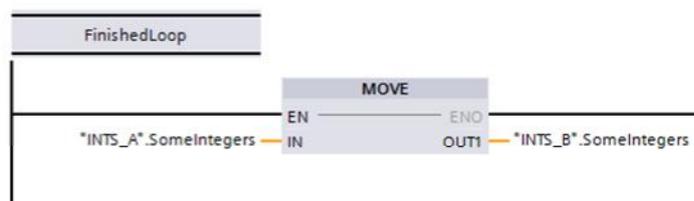
The following example copies 10 integer values from an array into another array.

Figure 7-1 – TIA Portal copy 10 elements in a loop



This can be implemented much easier with a “MOVE_BLK” instruction.

Figure 7-2 – TIA Portal copy 10 elements

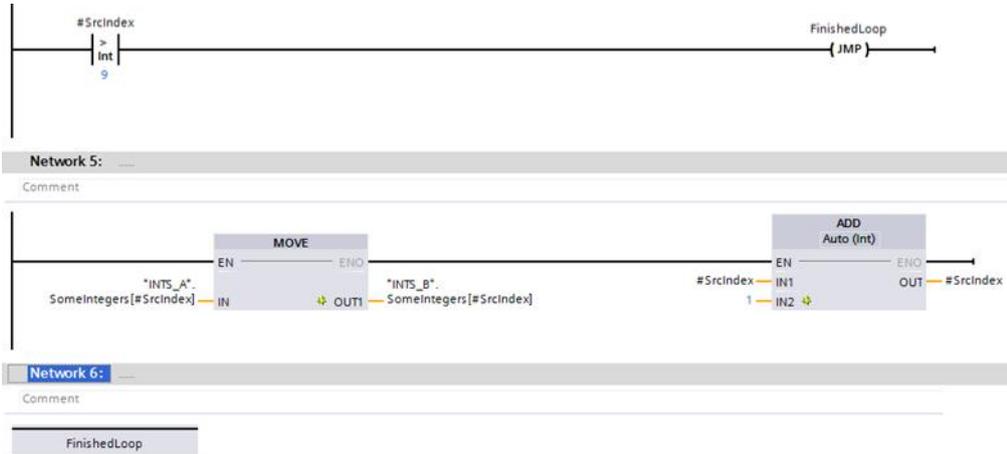


Before starting an own function development, an engineer should always check the provided instructions in the library. The functionality may already exist, helping shortening the development time and reducing runtime and memory usage.

7.8 Extensive Usage of Jumps

Jump instructions continue the execution at a certain marked address in the code. That mark is being applied by a label. Jumping inside the code is necessary, but should be used sparingly. It makes code difficult to read and maintain.

Figure 7-3 – TIA Portal copy 10 elements in a loop (Jump and Label)



The Jump is a very basic instruction, which is being used by compilers such as ST/SCL compilers to form conditional execution of certain code parts. The IEC 61131 programming language standard defines a GOTO instruction in ST/SCL, which resembles the Jump instruction.

7.9 Not using Symbolic Addressing

Symbolic addressing is being used, whenever a variable is referenced by its descriptor (or its name). The TIA Portal compilers assign a symbolic name, an alias, to every variable declared somewhere. Depending on preference settings these tags may not have an absolute address. The compiler is able to optimize data storage and access for increased runtime performance and lower memory consumption, when pure symbolic addressing is being used.

Mixing both absolute and symbolic addressing has basically two effects:

- a) The code becomes harder to read, to debug and to maintain
- b) The code will be executed much slower and uses much more memory

As such, any programmer migrating to a SIMATIC S7 would want to convert to symbolic programming, completely utilizing the full performance of the controller.

Mixing absolute and symbolic addressing confuses the engineer, because in some code parts one would see only absolute addresses not knowing the symbolic name for it. In other parts the same variable is being addressed properly using its symbolic name. The link between absolute address in memory and the assigned symbolic name can easily be lost.

7.10 Using indexing instead of indirect addressing

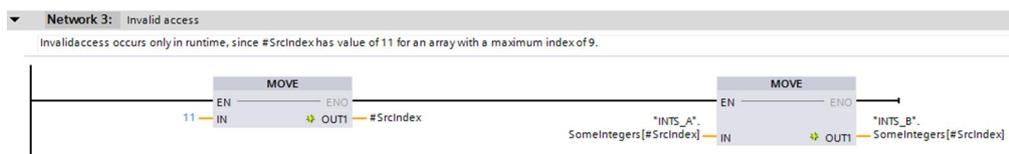
Programming purely absolute additionally prevents the compiler and the runtime system from optimizing for speed and memory usage. The variable is bound to a specific address. It is impossible to relocate it to another memory location, which may provide better accessibility.

7.10 Using indexing instead of indirect addressing

The use of indirect addressing is not possible in a purely symbolically addressed system. There is no means of which to calculate the physical address. The use of indirect addressing is therefore only possible when absolute addressing is being used and therefore degrades system performance.

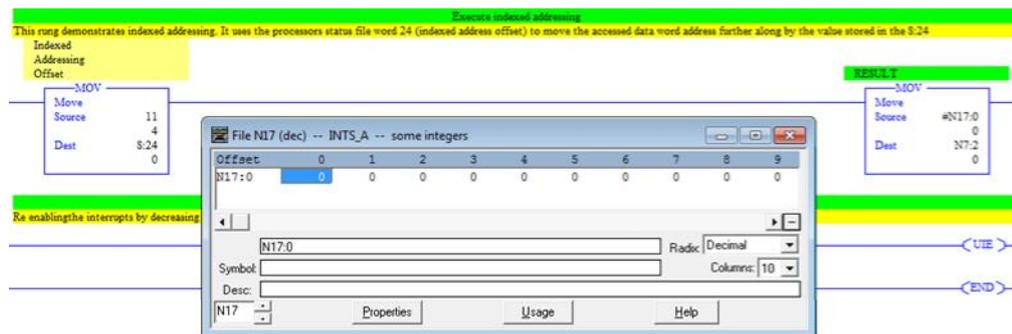
To improve performance indexing can be used in which an index variable is used to access data within an array. This index allows the systems to optimize memory usage and allows for consistent accessibility. It is also more stable, in that one can then use error checking, to ensure that they remain within the bounds of the data structure. (i.e. An error can be identified when an index of 11 is given for an array of length 9)

Figure 7-4 – TIA Portal causing invalid indexed access



A similar instruction on the PLC-5 causing the same invalid access on a PLC-5 controller would look like this.

Figure 7-5 – PLC-5 causing invalid indexed access



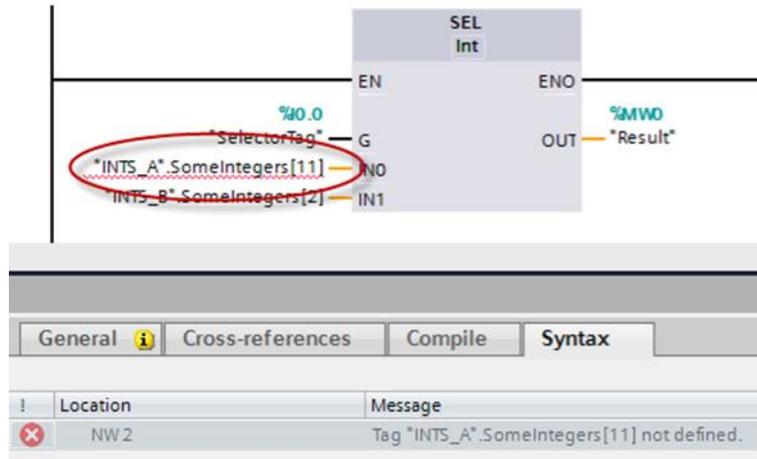
The above mentioned problem with indirect- and indexed addressing is the fact that the address calculation has been put into the engineers hands. Fully symbolic addressing avoids the mistakes possible by removing the information of the variables physical location. Therefore symbolic addressing should be the preferred method on the SIMATIC S7 controllers.

7 Common misunderstandings when converting to SIMATIC S7

7.10 Using indexing instead of indirect addressing

The below example demonstrates indexed addressing on a modern SIMATIC S7-1500 controller using fully symbolic access to an array. The compiler already detects the index 11 being out of bounds.

Figure 7-6 – TIA Portal indexing out of bounds



8 PLC-5 to S7 Glossary

8.1 Introduction

This chapter provides a glossary of terms used with Rockwell Automation equipment and the respective SIMATIC equivalent.

8.2 Hardware Terminology

Table 8-1 – glossary Hardware terminology

Rockwell Automation term	Definition	SIMATIC term	Definition
Chassis	The metal cage providing the backplane connectivity from the processor to the I/O modules	Rack	A metal rail providing only mechanical stability as well as an electrical grounding point
I/O scanner	Field bus connector to read and write I/O modules remotely	Master/ Controller	Defining a functionality to remotely collect sensor signals and control actuator signals
I/O adapter	Field bus connector to provide I/O modules to a controller	Slave/ Device	Defining a functionality to receive actuator signals and provide sensor signals to a remote Master/Controller
ControlNet		PROFIBUS/ PROFINET	
DeviceNet		ASi	
EtherNet/IP		PROFINET	
GuardLogix		Failsafe Controller	
PLC-5	Low-end to Mid-range controller	S7 1200 or S7 1500	
Controller or PAC		PLC or CPU	The processor unit/ module
Serial	Interface description utilizing a serial data transmission for either DF1 or DH485 protocols	Serial Interface	Interface card providing serial communications based on RS232 or RS485/422 physics. Some specific protocols are available
Bridge	Provides gateway functionality to forward CIP for programming, diagnostics and communications	Interface or CP/CM	Provides an interface for programming, diagnostics and communications.
Logix/ PLC-5/ SLC	Brand name for various generations of Rockwell Automation products	SIMATIC S7	The latest generation of Siemens automation systems

8.3 Software Terminology

Table 8-2 – glossary Software terminology

Rockwell Automation term	Definition	SIMATIC term	Definition
BTW/ BTR	Block Transfer Writing/ Block Transfer Reading instructions used to transfer I/O data from and to periphery modules	WR_REC/ RD_REC DP_RD_DAT/ DP_WR_DAT	Writing or Reading Data Records. Reading or Writing consecutive data in a consistent manner.
COP	Instruction for copying complex data types.	MOVE_BLK	
SINT	Short integer	SINT	Defines an 8-bit wide data type
FBD	Function Block diagram	FBD or CFC	
Continuous Task	Continuously executed user code entry point	Continuous OB	Cyclic program (typically OB1 "Main")
Program file	Encapsulated functionality	FC or FB	
Sequential Function Chart (SFC)	Sequentially executed program using a graphical programming language	GRAPH	IEC 61131 compliant version of SFC
I/O Configuration	Part of the RSLogix program to define the available I/O modules	Hardware Configuration	Part of the TIA Portal to define the hardware layout of the automation system.
Periodic Task	Periodically executing task	Cyclic interrupt OB	Periodically/ cyclic triggered interrupt program
LD (LAD)	Ladder Logic	LAD	Ladder Logic
Tag	A variable defines the data type and reserves the memory required to store the assigned value	Variable	A variable defines the data type and reserves the memory required to store the assigned value.
MCP	Master Control program	OB1	Organization Block which is being executed cyclic continuous.
RSLogix 5	Programming environment to program the PLC-5 controller	TIA Portal	Programming environment to program Simatic S7 controllers

9 Appendix A – PLC-5 parts and their S7 Equivalents

9.1 Digital input modules

Table 9-1 – PLC-5 digital input modules

Rockwell catalogue number	Range	Points/channels	SIEMENS catalogue number	Points/channels
1771-IN	24V AC/ DC	8	6ES7521-1BH00-0AB0	16
1771-IND	24V AC/ DC	16	---	---
1771-ID	120V AC/ DC	6	6ES7521-1FH00-0AA0	16
1771-IA	120V AC/ DC	8	6ES7521-1FH00-0AA0	16
1771-IAD	120V AC/ DC	16	6ES7521-1FH00-0AA0	16
1771-IAN	120V AC/ DC	32	6ES7521-1FH00-0AA0	16
1771-ID16	120V AC/ DC	16	6ES7521-1FH00-0AA0	16
1771-ID01	220V AC/ DC	6	6ES7521-1FH00-0AA0	16
1771-IM	220V AC/ DC	8	6ES7521-1FH00-0AA0	16
1771-SC-IMI16	220V AC/ DC	16	6ES7521-1FH00-0AA0	16
1771-IGD	5V DC/ TTL	16	---	---
1771-IB	24V DC	8	6ES7521-1BH00-0AB0	16
1771-IC	48V DC	8	---	---
1771-IH	48V DC	8	---	---
1771-IT	24V DC	8	6ES7521-1BH00-0AB0	16
1771-IBD	24V DC	16	6ES7521-1BH00-0AB0	16
1771-ICD	48V DC	16	---	---
1771-IBN	24V DC	32	6ES7521-1BL00-0AB0	32
1771-IV	24V DC	8	6ES7521-1BH00-0AB0	16
1771-IVN	24V DC	32	6ES7521-1BL00-0AB0	32
1771-IQ	24V DC	8	6ES7521-1BH00-0AB0	16
1771-IQ16	24V DC	16	6ES7521-1BH00-0AB0	16

9.2 Digital output modules

Table 9-2 – PLC-5 digital output modules

Rockwell catalogue number	Range	Points/channels	SIEMENS catalogue number	Points/channels
1771-ON	24V AC	8	6ES7522-5HF00-0AB0	8
1771-OP	120V AC	4	6ES7522-5FF00-0AB0	8
1771-OD	120V AC	6	6ES7522-5FF00-0AB0	8
1771-ODD	120V AC	16	6ES7522-5FF00-0AB0	8
1771-OD16	120V AC	16	6ES7522-5FF00-0AB0	8
1771-ODZ	120V AC	8	6ES7522-5FF00-0AB0	8
1771-OA	120V AC	8	6ES7522-5FF00-0AB0	8
1771-OAD	120V AC	16	6ES7522-5FF00-0AB0	8
1771-OR	220V AC	6	6ES7522-5FF00-0AB0	8
1771-OM	220V AC	8	6ES7522-5FF00-0AB0	8
1771-OMD	220V AC	8	6ES7522-5FF00-0AB0	8
1771-OAN	120/ 220V AC	16	6ES7522-5FF00-0AB0	8
1771-SC-OMI16	120/ 220V AC	32	6ES7522-5FF00-0AB0	8
1771-OG	5V DC/ TTL	8	---	---
1771-OGD	5V DC/ TTL	16	---	---
1771-OB	60V DC	16	6ES7522-5HF00-0AB0	8
1771-OBDS	40V DC	16	6ES7522-5HF00-0AB0	8
1771-OBN	30V DC	32	6ES7522-5HF00-0AB0	8
1771-OQ	24V DC	8	6ES7522-1BF00-0AB0	8
1771-OQ16	32V DC	16	6ES7522-5HF00-0AB0	8
1771-OVN	30V DC	32	6ES7522-5HF00-0AB0	8
1771-OW	120V DC Relay	8	6ES7522-5HF00-0AB0	8
1771-OW16	40V DC Relay	16	6ES7522-5HF00-0AB0	8
1771-OWNA	120V DC Relay	32	6ES7522-5HF00-0AB0	8
1771-OYL	Relay	8	6ES7522-5HF00-0AB0	8
1771-OZL	Relay	8	6ES7522-5HF00-0AB0	8

9.3 Analogue input modules

Table 9-3 – PLC-5 analogue input modules

Rockwell catalogue number	Range	Points/channels	SIEMENS catalogue number	Points/channels
1771-IFE	+/- 5V, +/- 10V, +/- 20mA	8	6ES7531-7NF10-0AB0	8
1771-IFF	+/- 5V, +/- 10V, +/- 20mA	8	6ES7531-7NF10-0AB0	8
1771-IFMS	0..50mV	8	6ES7531-7KF00-0AB0	8
1771-IL	+/- 5V, +/- 10V, +/- 20mA	8	6ES7531-7NF10-0AB0	8
1771-IXE	Type E, J, K, R, S, T, +/- 100mV	8	6ES7531-7KF00-0AB0	8
1771-IXHR	Type B, E, J, K, R, S, T, +/- 100mV	8	6ES7531-7KF00-0AB0	8
1771-IR	PT100, CU10	6	6ES7531-7KF00-0AB0	8
1771-NIS	4..20mA	8	6ES7531-7NF10-0AB0	8
1771-NIV	+/- 5V, 4..20mA	8	6ES7531-7NF10-0AB0	8
1771-NIV1	+/- 5V, 4..20mA	8	6ES7531-7NF10-0AB0	8
1771-NIVR	+/- 5V, 4..20mA, RTD	4, 4	6ES7531-7KF00-0AB0	8
1771-NIVT	+/- 5V, 4..20mA, TC	4, 4	6ES7531-7KF00-0AB0	8
1771-NR	RTD	8	6ES7531-7KF00-0AB0	8
1771-NT1	TC	8	6ES7531-7KF00-0AB0	8
1771-NT2	TC	8	6ES7531-7KF00-0AB0	8

9.4 Analogue output modules

Table 9-4 – PLC-5 analogue output modules

Rockwell catalogue number	Range	Points/channels	SIEMENS catalogue number	Points/channels
1771-OFE1	+/- 10V	4	6ES7532-5HD00-0AB0	4
1771-OFE2	4..20mA	4	6ES7532-5HD00-0AB0	4
1771-OFE3	0..50mA	4	6ES7532-5HD00-0AB0	4
1771-NOC	0..25mA	8	6ES7532-5HF00-0AB0	8
1771-NOV	+/- 10V	8	6ES7532-5HF00-0AB0	8

9.5 Analogue mixed modules

Table 9-5 – PLC-5 mixed analogue modules

Rockwell catalogue number	Range	Points/channels	SIEMENS catalogue number	Points/channels
1771-NB4S	4..20mA/ 0..25mA	2 In/ 2 Out	6ES7531-7NF10-0AB0/ 6ES7532-5HD00-0AB0	8/ 8
1771-NB4T	+/-100mV, Type B, R, S, E, J, K, T/ 0..25mA	2 In/ 2 Out	6ES7531-7KF00-0AB0/ 6ES7532-5HD00-0AB0	8/ 8
1771-NBRC	PT100, NI120, CU10/ 0..25mA	6 In/ 2 Out	6ES7531-7KF00-0AB0/ 6ES7532-5HD00-0AB0	8/ 8
1771-NBSC	4..20mA/ 0..25mA	6 In/ 2 Out	6ES7531-7NF10-0AB0/ 6ES7532-5HD00-0AB0	8/ 8
1771-NBTC	+/-100mV, Type B, R, S, E, J, K, T/ 0..25mA	6 In/ 2 Out	6ES7531-7KF00-0AB0/ 6ES7532-5HD00-0AB0	8/ 8
1771-NBV1	+/-10V, +/-20mA/ +/- 10V	6 In/ 2 Out	6ES7531-7NF10-0AB0/ 6ES7532-5HD00-0AB0	8/ 8
1771-NBVC	+/-5V, +/-20mA/ 0..25mA	6 In/ 2 Out	6ES7531-7NF10-0AB0/ 6ES7532-5HD00-0AB0	8/ 8

NOTE

The above mentioned combination of 2 SIMATIC modules is a recommendation, which needs to be evaluated. It may be possible to combine the output signals of multiple PLC-5 modules (1771) into a single SIMATIC S7 analogue output module.

Analogue values are provided by the PLC-5 modules in two different formats.

- As a digital representation of the converted value in a resolution dependent range, e.g. 0..4095 (12 Bit resolution) plus sign bit
- As a binary coded digital (4 digit BCD) representation of the converted value in a resolution dependent range, e.g. 0..4095
- As engineering units for some specific PLC-5 analogue modules (1771 product family).

In the SIMATIC S7 automation system the analogue periphery modules are controlled identical, regardless of their resolution. The input values, read back from the modules, are always in the same range regardless of the ADC resolution. This makes scaling to the engineering units easier and can be realized with a function "Scale". It follows the simple formula for a linear correlation.

$$\text{measured engineering units} = \text{digits} * \frac{\text{Full Scale (engineering units)}}{27648}$$

This formula can be used for the calculation of engineering units for analogue input values. The calculation for the output values follows the same correlation and uses this formula:

$$\text{digits} = \text{control engineering units} * \frac{27648}{\text{Full Scale (engineering units)}}$$

Analogue value representation

The following table lists some common resolutions available with analogue modules.

Table 9-6 – analogue value representation

ADC resolution	Number of digits per full scale (FS)	Physical unit per digit
10 Bit	1024 digits	0..10V → 9.76 mV/ digit -10 .. 10V → 19.53 mV/ digit 0..20mA → 19.53 µA/ digit -20 .. 20mA → 39.06 µA/ digit
11 Bit	2048 digits	0..10V → 4.88 mV/ digit -10 .. 10V → 9.76 mV/ digit 0..20mA → 9.76 µA/ digit -20 .. 20mA → 19.53 µA/ digit
12 Bit	4096 digits	0..10V → 2.44 mV/ digit -10 .. 10V → 4.88 mV/ digit 0..20mA → 4.88 µA/ digit -20 .. 20mA → 9.76 µA/ digit
14 Bit	16.384 digits	0..10V → 0.61 mV/ digit -10 .. 10V → 1.22 mV/ digit 0..20mA → 1.22 µA/ digit -20 .. 20mA → 2.44 µA/ digit
15 Bit	32.768 digits	0..10V → 0.31 mV/ digit -10 .. 10V → 0.61 mV/ digit 0..20mA → 0.61 µA/ digit -20 .. 20mA → 1.22 µA/ digit

NOTE

10 bit resolution is not commonly used, since its accuracy is not good enough for many applications. In a 10 bit Analog to Digital Converter (ADC) the always existing quantization error has a greater impact on the process value.

In the SIMATIC S7 analogue modules the resolution is always 16 bit (incl. sign bit). However the full scale value is represented by 27648 digits yielding a resolution of 0.72 mV/ digit assuming a physical unit range of -10V .. 10V. This allows the safe measurement of a values slightly beyond the nominal range, such as -11.8V .. 11.8V.

9.6 Specialty modules

The PLC-5 controllers are able to use “specialty” modules designed for a specific purpose, such as counting of impulses in certain voltage ranges. Another typical “specialty” module is a stepper motor controller. Most of these “specialty” modules are provided by third party companies and can be replaced by an appropriate Siemens part.

Table 9-7 – PLC-5 specialty modules

Rockwell catalogue number	Range	Points/channels	SIEMENS catalogue number	Points/channels
1771-TCM	TC temperature PID loop (heating/cooling)	8 channel	Available in different form factor (ET200M)	
1771-TCMR	RTD temperature PID control loop (heating/cooling)	8 channel	Available in different form factor (ET200M)	
1771-DL	Gray Code Encoder input (1kHz)	4 channel	6ES7551-1AB00-0AB0	2 channel
1771-QD	Injection molding control module	3 AI/ 2AQ	--- *)	
1771-QDC	Plastic molding control module	4 AI/ 4 AQ	--- *)	
1771-QI	Co-Injection module	4 AI/ 4AQ	--- *)	
1771-VHSC	High Speed Counter module (1MHz)	3 channel	6ES7550-1AA00-0AB0	
1771-CFM	Configurable flow meter (50mV ... 200V AC)	4 DI/ 4 DQ	6ES7531-7NF10-0AB0	8 analogue channel
1771-PM	Clutch/ Brake control system		--- *)	
1771-SIM	8 channel simulator module	8 DI/ 8 DQ	Will be realized via simulator board	16 channel

*) There is no specific module available in the SIMATIC S7 module spectrum, however for specific applications other modules and module combinations may be suitable for the application.

NOTE

The suitability for a specific application needs to be evaluated on a case-by-case.

9.7 Conversion of remote Point I/O modules

Specifically for the translation of Rockwell's Point I/O Siemens provides an online tool to realize this. This tool can be accessed using the following link:

<https://www.automation.siemens.com/ET200Migrator/member/ET200Migrator.aspx?lang=en>

9.8 PLC-5 CPU modules

Enhanced PLC-5 CPU modules

Table 9-8 – Enhanced PLC-5 controllers

Rockwell catalogue number	User Memory (Words)	Total I/O	Number of I/O chassis	Siemens catalogue number	User Memory (Bytes)	Total I/O	Number of I/O chassis
1785-L11B	8000	512 any mix	5	6ES7 511-1AK00-0AB0	150k	1k	256
1785-L20B 1785-L26B	16,000	512 any mix	13	6ES7 511-1AK00-0AB0	150k	1k	256
1785-L30B	32,000	1024 any mix	29	6ES7 511-1AK00-0AB0	150k	1k	256
1785-L40B 1785-L46B	48,000	2048 any mix	61	6ES7 513-1AL00-0AB0	300k	2k	256
1785-L60B	64,000	3072 any mix	93	6ES7 513-1AL00-0AB0	300k	2k	256
1785-L80B 1785-L86B	100,000	3072 any mix	93	6ES7 515-2AM00-0AB0	500k	8k	256

Ethernet PLC-5 CPU modules

Table 9-9 – Ethernet PLC-5 controllers

Rockwell catalogue number	User Memory (Words)	Total I/O	Number of I/O chassis	Siemens catalogue number	User Memory (Bytes)	Total I/O	Number of I/O chassis
1785-L20E	16,000	512 any mix	13	6ES7 511-1AK00-0AB0	150k	1k	256
1785-L40E	48,000	2048 any mix	61	6ES7 513-1AL00-0AB0	300k	2k	256
1785-L80E	100,000	3072 any mix	65	6ES7 515-2AM00-0AB0	500k	8k	256

ControlNet PLC-5 CPU modules

Table 9-10 – ControlNet PLC-5 controllers

Rockwell catalogue number	User Memory (Words)	Total I/O	Number of I/O chassis	Siemens catalogue number	User Memory (Bytes)	Total I/O	Number of I/O chassis
1785-L20C15	16,000	512 any mix	77	6ES7 511-1AK00-0AB0	150k	1k	256
1785-L40C15	48,000	2048 any mix	125	6ES7 513-1AL00-0AB0	300k	2k	256
1785-L46C15	48,000	2048 any mix	125	6ES7 513-1AL00-0AB0	300k	2k	256
1785-L80C15	100,000	3072 any mix	125	6ES7 515-2AM00-0AB0	500k	8k	256

10 Appendix B – Panel Builder HMI Cross Reference Table

Since there is a large table available translating a Rockwell PanelView part number into a Siemens part number, which would exceed the page's width, it has been embedded in an EXCEL spreadsheet document, „104206213_HMI CrossReference.xlsx”.

This EXCEL spreadsheet is separately embedded in the website, where you downloaded this document.

11 Appendix C – Figures

Figure 1-1 – SIEMENS Industry Online Support Web Page.....	7
Figure 1-2 – SIEMENS Service & Support Web Page Heading	7
Figure 2-1 – TIA Portal S7 central 1500 station.....	14
Figure 2-2 – Hardware and software structure for a S88 process control application	20
Figure 2-3 – Partial Retrofit.....	22
Figure 2-4 – TIA Portal Hardware catalogue.....	23
Figure 2-5 – TIA Portal Panel connection types	24
Figure 5-1 – TIA Portal Slice access.....	30
Figure 5-2 – PLC-5 Timer with T4 data file	33
Figure 5-3 – TIA Portal Single Instance Timer.....	33
Figure 5-4 – PLC-5 Counter file type	36
Figure 5-5 – TIA Portal IEC Counter.....	36
Figure 5-6 – TIA Portal Array of Integer.....	37
Figure 5-7 – TIA Portal Array of Real and LReal.....	37
Figure 5-8 – RSLogix5 indirectly addressing a data file	42
Figure 5-9 – TIA Portal selecting the data file	42
Figure 5-10 – PLC-5 indexed addressing using S:24	43
Figure 5-11 – SIMATIC S7 controller indexing an array.....	44
Figure 6-1 – TIA Portal Global Library Basic instructions.....	46
Figure 6-2 – TIA Portal Global Library - Move instructions.....	47
Figure 6-3 – TIA Portal - Global Library Extended instructions.....	49
Figure 6-4 – TIA Portal - writing system time	50
Figure 6-5 – TIA Portal - writing local time	50
Figure 6-6 – TIA Portal - reading system time in ns (S7-1200 and S7-1500 only).....	50
Figure 6-7 – TIA Portal Global Library Interrupt instructions	51
Figure 6-8 – TIA Portal Global Library Timer instructions	53
Figure 6-9 – TIA Portal Global Library Counter instructions.....	53
Figure 6-10 – TIA Portal Global Library String Conversion instructions	54
Figure 6-11 – TIA Portal TO PID Control	55
Figure 6-12 – TIA Portal enabling and disabling interrupts.....	57
Figure 7-1 – TIA Portal copy 10 elements in a loop.....	62
Figure 7-2 – TIA Portal copy 10 elements.....	62
Figure 7-3 – TIA Portal copy 10 elements in a loop (Jump and Label)	63
Figure 7-4 – TIA Portal causing invalid indexed access.....	64
Figure 7-5 – PLC-5 causing invalid indexed access	64
Figure 7-6 – TIA Portal indexing out of bounds	65

12 Appendix D – Tables

Table 1-1 – Terminology	6
Table 1-2 – Referenced documents	8
Table 2-1 – Topics of chapter “Introduction”	11
Table 2-2 – PLC-5 Controller Replacement	12
Table 2-3 – Periphery network comparison	16
Table 2-4 – Allen- Bradley periphery systems and SIEMENS equivalents	16
Table 5-1 – PLC-5 data types and the SIMATIC equivalents	29
Table 5-2 – PLC-5 addressing to SIMATIC addressing	30
Table 5-3 – PLC-5 Status file data integration into a SIMATIC program	31
Table 5-4 – PLC-5 Timer file type conversion to SIMATIC	33
Table 5-5 – PLC-5 constant conversions to SIMATIC	41
Table 6-1 – Topics of chapter “Conversion of System Software and Standard Functions” ..	45
Table 6-2 – Conversion of PLC-5 copy instructions	48
Table 6-3 – Conversion of PLC-5's UID and UIE instruction	52
Table 6-4 – PLC-5 date and time structure & SIMATIC date and time structure	56
Table 6-5 – Disabling and Enabling Interrupts	57
Table 6-6 – Reading System time	57
Table 6-7 – Get Faults	58
Table 6-8 – Get Scan Time	59
Table 8-1 – glossary Hardware terminology	66
Table 9-1 – PLC-5 digital input modules	68
Table 9-2 – PLC-5 digital output modules	69
Table 9-3 – PLC-5 analogue input modules	70
Table 9-4 – PLC-5 analogue output modules	70
Table 9-5 – PLC-5 mixed analogue modules	71
Table 9-6 – analogue value representation	72
Table 9-7 – PLC-5 specialty modules	73
Table 9-8 – Enhanced PLC-5 controllers	74
Table 9-9 – Ethernet PLC-5 controllers	75
Table 9-10 – ControlNet PLC-5 controllers	75
Table 13-1 – document revision history	78

13 History

Table 13-1 – document revision history

Revision	Author	Date	Changes
0.9	Scheel, Olaf	04/23/2014	Initial version
1.0	Scheel, Olaf	10/23/2014	<ul style="list-style-type: none"> • Edited for wording • Added content • Added tables for hardware conversion