A NEW EPICS DEVICE SUPPORT FOR S7 PLCS

S. Marsching*, aquenos GmbH, Baden-Baden, Germany

Abstract

S7 series programmable logic controllers (PLCs) are commonly used in accelerator environments. A new EPICS device support for S7 PLCs that is based on libnodave has been developed. This device support allows for a simple integration of S7 PLCs into EPICS environments. Developers can simply create an EPICS record referring to a memory address in the PLC and the device support takes care of automatically connecting to the PLC and transferring the value. This contribution presents the concept behind the s7nodave device support and shows how simple it is to create an EPICS IOC that communicates with an S7 PLC.

INTRODUCTION

Siemens S7 programmable logic controllers (PLCs) [1] are prevalent in industrial automation applications because they offer a modular architecture and a wide range of products for different applications. The platform is so popular that other companies offer supposedly compatible PLCs.

As they are widely used in experimental physics facilities, integration into the control system should be made as simple as possible. Therefore, aquenos developed the s7nodave device support [2] for EPICS [3]. It allows read and write access to the PLC's memory without having to do any control-system specific programming on the PLC. The addressing scheme used is the same that is used by the STEP 7 [4] development environment. Thus, using s7nodave, it is very easy for PLC developers to integrate their devices into an EPICS-based control system.

S7 PLC PLATFORM

The family of S7 PLCs offers a wide range of devices. A CPU module is at the center of each device and can be extended by adding input / output (I/O) modules and communication processors. Communication processors connect the PLC to the outside world using various field-bus interfaces. Some of the CPUs have some I/O interfaces and a communication processor already built in.

In order to use s7nodave, a communication processor supporting PROFINET or a CPU with PROFINET support is needed. PROFINET effectively offers an Ethernet-compatible TCP/IP interface, that can be used for communication between the PLC and a PC running the EPICS software.

The S7-1200 [5] sub-family is of particular interest because all of its CPUs have the PROFINET interface and some I/O interfaces already built-in while being offered at very competitive prices. Therefore, they are interesting for applications where traditionally I/O modules based on a field-bus would have been used.

Memory Model

From the programmer's point of view, the memory of an S7 PLC is divided into different areas (see Table 1). Typically, the areas of interest for access from an EPICS driver are the input, the output, the marker, and the data block area.

Table 1: S7 PLC Memory Areas

Name	Description
I	Input
Q	Output
M	Marker
DBn	Data Block
C	Counter
T	Timer

The input and output areas represent the memory image of the input and output modules. At the start of each cycle, the state of the input and output modules is copied to these areas. At the end of each cycle, the state stored in the output area is copied back to the output modules, applying the changes made in memory to the outputs. The data block and marker areas are used to store the internal state of the program running on the PLC.

The STEP 7 notation of memory addresses consists of the area code, the width of the data (if it is not a single bit) and the address offset within the area. For example, the address I2.3 refers to the fourth bit in the third byte of the input area and the address QW4 refers to the fifth and sixth byte (two bytes form a word) in the output area.

Remote Access

For remote access over the Ethernet interface, there are two feasible approaches: First, code that reads data from and writes data to a TCP socket can be added to the PLC program. A remote program, like an EPICS Input / Output Controller (IOC), can then use this TCP socket to communicate with the PLC. This is the approach taken by the S7 PLC device support from PSI [6]. The disadvantage of this approach is that every piece of information that is supposed to be used outside the PLC has to be added to the memory block sent and received by the program running on the PLC. If another piece of information (e.g. an additional input) shall be read by the EPICS IOC, the PLC program has to be changed.

* sebastian<dot>marsching<at>aquenos.com

ISBN 978-3-95450-139-7

The other approach uses the same protocol that is used by the STEP 7 development environment and the supervisory control and data acquisition (SCADA) solutions offered by Siemens. The support for this protocol is built into the PLC's communication processor. Thus, it allows direct access to the PLC's memory without having to add code to the program running on the PLC. Siemens offers the commercial PRODAVE library [7] as a PC-side implementation of this protocol. Unfortunately, there is no official specification of this protocol. However, various competing libraries have been developed through reverse-engineering. The most interesting one is the Libnodave library [8], which is the only implementation released under an open-source license. The s7nodave EPICS device support discussed in this paper uses this library for communication with the PLC.

S7NODAVE

The s7nodave device support for EPICS offers an easy way to integrate S7 PLCs into EPICS-based control systems. It has been tested with PLCs of the S7-300 and S7-1200 families, however PLCs of the S7-400 and S7-1500 families are also likely to be compatible. The user can connect to one or multiple PLCs in the IOC start-up file (see Fig. 1) and then refer to their memory from EPICS records without having to know any details about the communication process. For example, the device support transparently takes care of re-establishing the connection after a network interruption. Therefore, the application developer can fully concentrate on the application logic rather than having to deal with low-level details.

st.cmd

```
[...]
# Configure PLC connection.
s7nodaveConfigureIsoTcpPort("myPLC",
  "s7plc.example.com", 0)
# Set polling interval for "default" poll
# group to 1 second.
s7nodaveConfigurePollGroup("myPLC",
  "default", 1.0, 0)
# Create a second poll group for records
# that shall be polled every 100 ms.
s7nodaveConfigurePollGroup("myPLC,
  "fast", 0.1, 0)
[...]
```

Figure 1: Example of an IOC start-up file.

Addressing

Figure 2 shows two example EPICS record definitions. The device address in an EPICS records consists of four parts:

record(ai, "AnalogInput")

```
{
  field(SCAN, "I/O Intr")
  field(DTYP, "s7nodave")
  field(INP, "@myPLC(DLV=0,DHV=27648) IW66")
  field(LINR, "LINEAR")
  field(EGUL, "0")
  field(EGUF, "10")
  field(EGU, "V")
}
```

record(bo, "BinaryOutput")

```
{
  field(DTYP, "s7nodave")
  field(OUT, "@myPLC Q0.3")
}
```

Figure 2: Example of an input and output record definition.

The first part is the name of the PLC (in this case "myPLC"), which has to match the name used in the IOC start-up script.

The second part (specified in parentheses) is optional. It is used to specify additional options like the range of the value used by the PLC. This information can be used to automatically convert integer values coming from analog-digital converters (ADCs) into floating-point numbers, that represent the actually measured quantity. In the first example, the ADC returns a number between 0 and 27648, which is equivalent to a range from 0 to 10 V.

The third part of the device address is the memory address in the PLC. This memory address uses the same notation that is also used by STEP 7. Therefore, addresses can easily be copied from the PLC program code.

Finally, there is a fourth optional part of the device address, which is not shown in this example: This part is the PLC data type that has to be specified when the memory address specified could represent different data-types: For example, a double-word could represent a 32-bit signed or unsigned integer or a single-precision floating-point number. By knowing the data-type, s7nodave can automatically take care of converting the number between its representation in the PLC's memory and its representation on the host running the EPICS IOC. This includes handling differences in endianness.

Poll Groups

Updating many different input records by scanning them individually can be very inefficient because each update requires a full network round-trip. Therefore, input records can be aggregated in so-called poll groups. A poll group is a group of input records that should all be updated at the same rate. A record is made part of a poll-group by setting

ISBN 978-3-95450-139-7

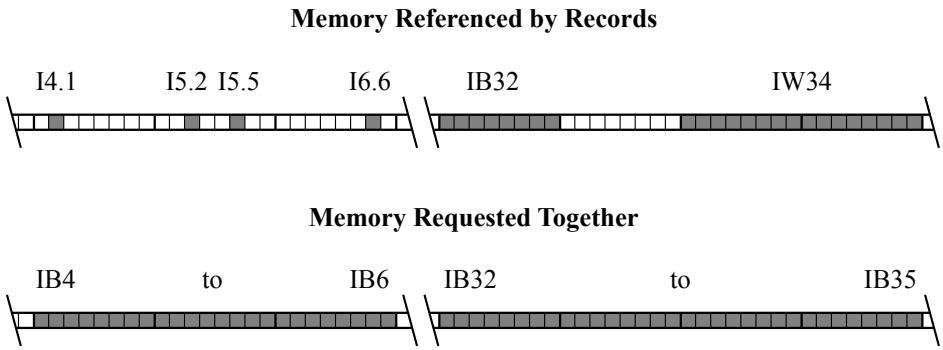

Figure 3: Records referring to close-by memory sections are grouped together when using poll groups.

its "SCAN" field to "I/O Intr" and optionally specifying a poll-group name. If no name is specified, the "default" poll group is used.

When a poll group is processed, the memory sections needed by the different records are joined to as few requests as possible, reducing the number of round-trips needed significantly. As each memory section requested incurs a small overhead due to headers, the poll-group algorithm intelligently joins adjacent and close-by memory sections to a single section, increasing the number of sections that can be put into a single request. Figure 3 illustrates this concept.

SUMMARY

The s7nodave device support for EPICS offers a new, simpler way for integrating S7 PLCs into EPICS environments. Now PLC developers can access the PLC's memory from EPICS in a way that they are used to from the STEP 7 development environment and commercial SCADA solutions. By using poll groups, the number of network round-trips is optimized, bringing the performance close to a solution using custom PLC code.

REFERENCES

[1] Siemens, "Modular PLC controllers SIMATIC S7", http://www.automation.siemens.com/mcms/programmable-logic-controller/en/simatic-s7-controller/

[2] aquenos GmbH, "s7nodave Device Support for EPICS", January 2013, http://oss.aquenos.com/epics/s7nodave/.

[3] "EPICS – Experimental Physics and Industrial Control System", http://www.aps.anl.gov/epics/

[4] Siemens, "SIMATIC STEP 7 Engineering Software", http://www.automation.siemens.com/mcms/simatic-controller-software/en/step7/

[5] Siemens, "Compact automation solutions – SIMATIC S7-1200", http://www.automation.siemens.com/mcms/programmable-logic-controller/en/simatic-s7-controller/s7-1200/

[6] D. Zimoch, "EPICS S7 PLC Device Support", http://epics.web.psi.ch/software/s7plc/

[7] Siemens, "PRODAVE", http://eb.automation.siemens.com/mall/de/WW/Catalog/Products/5000150

[8] T. Hergenhahn, "libnodave", February 2011, http://libnodave.sourceforge.net/

ISBN 978-3-95450-139-7