

Siemens S7 PLC Communication

Theory of Operation

- PLC and IOC exchange blocks of PVs over TCP/IP.
- Blocks can have arbitrary length and layout.

▶ Example:

<i>offs</i>	<i>pv</i>	<i>type</i>
0	status	(16 bits)
2	voltage	(long int)
6	temperature	(float)
10	message	(string[12])
22		

- Only complete data blocks can be transmitted.
- PLC and/or IOC send periodically or when values change.
- Any number of PLCs may be connected to one IOC.

Driver setup

■ Ask programmer of PLC for

- ▶ IP address and port.
- ▶ byte order: big endian or little endian.
- ▶ block size and send period PLC → IOC (100 msec ~ seconds).
- ▶ block size and send period IOC → PLC (100 msec ~ seconds).

■ Give the PLC a name.

■ Configure driver in startup script:

```
s7plcConfigure ("PlcName", "Ipaddress", TCPport,  
               PlcToIocSize, IocToPlcSize, bigEndian,  
               PlcSendTimeout, IocSendPeriod)
```

Record setup

■ Ask programmer of PLC for

- ▶ Layout of PVs (offset, data type and size)
- ▶ Meta data (limits, bit shifts, units, etc.)

■ DTYP is "S7plc".

■ INP / OUT link

"@PlcName/offset T=type L=low H=high B=bit"

- ▶ Not all parameters required in all cases.
- ▶ L and H used in analog records for conversion.
- ▶ B used in binary records for bit number.

Input records

- SCAN should be "I/O Intr".
 - ▶ record processes whenever new data arrives.
- When PLC does not send for `PlcSendTimeout`, connection is closed and reopened (allow 2~5 times PLC send period).
 - ▶ Driver raises alarm: SEVR / STAT = INVALID / CONN
- Special DTYP "S7plc stat" for bi record.
 - ▶ Connection monitor: 1 when connected.
 - ▶ No alarm by driver.

Output records

- PINI should be "YES".
 - ▶ Initializes output block before first data is sent.
 - ▶ Compatible with auto-save-and-restore.
- Driver checks each `IocSendInterval` for new output.
 - ▶ All changes during one interval are collected.
 - ▶ If nothing has changed, nothing is sent.
 - ▶ Limited network traffic but latency.
- SCAN can be "I/O Intr" (unusual).
 - ▶ Record processes in every `IocSendInterval`.

Example

- PLC "dev-x" at address 192.168.0.10
- TCP server port 2000
- 22 byte input from PLC at least every 500 msec
- 2 byte output to PLC maximal every 100 msec
- Big endian byte order

```
s7plcConfigure ("dev-x", "192.168.0.10", 2000,  
                22, 2, 1, 500, 100)
```

Example (cont'd)

■ 16 bit status word at offset 0

```
record (mbbiDirect, "$(DEV):status") {  
    field (DTYP, "S7plc")  
    field (INP, "@dev-x/0 T=WORD")  
    field (NOBT, "16")  
    field (SCAN, "I/O Intr")  
}
```


Example (cont'd)

- 24 bit integer DAC value in 4 bytes at offset 2
- Range: $0x00000000 = -24V$ $0x00FFFFFF = +24V$

```
record (ai, "$(DEV):voltage") {  
    field (DTYP, "S7plc")  
    field (INP, "@dev-x/2 T=INT32 L=0 H=0x00FFFFFF")  
    field (EGUL, "0")  
    field (EGUF, "24")  
    field (LINR, "LINEAR")  
    field (EGU, "V")  
    field (SCAN, "I/O Intr")  
}
```

Example (cont'd)

- Single precision float temperature (in °C) at offset 6
- User wants °F. (No idea why.)

```
record (ai, "$(DEV):temperature") {  
    field (DTYP, "S7plc")  
    field (INP, "@dev-x/6 T=FLOAT")  
    field (ASLO, "1.8")  
    field (AOFF, "32")  
    field (EGU, "°F")  
    field (SCAN, "I/O Intr")  
}
```

Example (cont'd)

- 12 byte string message at offset 10

```
record (stringin, "$(DEV):message") {  
    field (DTYP, "S7plc")  
    field (INP, "@dev-x/10 L=12")  
    field (SCAN, "I/O Intr")  
}
```

Example (cont'd)

- 2 byte command output, bits 4 and 5 for switch
- 01: switch on, 10: switch off

```
record (mbbo, "$(DEV):switch") {  
    field (DTYP, "S7plc")  
    field (OUT, "@dev-x/2 T=WORD")  
    field (NOBT, "2")  
    field (SHFT, "4")  
    field (ZRVL, "2")  
    field (ZRST, "OFF")  
    field (ONVL, "1")  
    field (ONST, "ON")  
    field (PINI, "YES")  
}
```

Supported record types

■ Numeric

- ▶ ai, ao, longin, longout, calcout

■ Binary

- ▶ bi, bo, mbbi, mbbo, mbbiDirect, mbboDirect

■ Array

- ▶ stringin, stringout, waveform

Download and more info:

<http://epics.web.psi.ch/software/s7plc>

Thanks to Yu-Han Lin for the PLC configuration document.