



# S7comm

## S7 Communication (S7comm)

---

S7comm (S7 Communication) is a Siemens proprietary protocol that runs between programmable logic controllers (PLCs) of the Siemens S7-300/400 family.

It is used for PLC programming, exchanging data between PLCs, accessing PLC data from SCADA (supervisory control and data acquisition) systems and diagnostic purposes.

The S7comm data comes as payload of COTP data packets. The first byte is always 0x32 as protocol identifier. Special communication processors for the S7-400 series (CP 443) may use this protocol without the TCP/IP layers.

	OSI layer	Protocol
7	Application Layer	S7 communication
6	Presentation Layer	S7 communication
5	Session Layer	S7 communication
4	Transport Layer	ISO-on-TCP (RFC 1006)
3	Network Layer	IP
2	Data Link Layer	Ethernet
1	Physical Layer	Ethernet

To establish a connection to a S7 PLC there are 3 steps:

1. Connect to PLC on TCP port 102
2. Connect on ISO layer (COTP Connect Request)
3. Connect on S7comm layer (s7comm.param.func = 0xf0, Setup communication)

Step 1) uses the IP address of the PLC/CP.

Step 2) uses as a destination TSAP of two bytes length. The first byte of the destination TSAP codes the communication type (1=PG, 2=OP). The second byte of the destination TSAP codes the rack and slot number:

This is the position of the PLC CPU. The slot number is coded in Bits 0-4, the rack number is coded in Bits 5-7.

Step 3) is for negotiation of S7comm specific details (like the PDU size).

## History

---

The protocol is used by Siemens since the Simatic S7 product series was launched in 1994. The protocol is also used on top of other physical/network layers, like RS-485 with MPI (Multi-Point-Interface) or Profibus.

## Protocol dependencies

---

S7 communication consists of (at least) the following protocols:

- **COTP**: ISO 8073 COTP Connection-Oriented Transport Protocol (spec. available as [RFC905](#))
- **TPKT**: [RFC1006](#) "ISO transport services on top of the TCP: Version 3", updated by RFC2126
- **TCP**: Typically, TPKT uses [TCP](#) as its transport protocol. The well known TCP port for TPKT traffic is 102.

## Example traffic

---

Filter: s7comm

No.	Time	Source	Destination	Protocol	Info
9	2.016210	192.168.1.10	192.168.1.40	S7COMM	ROSCTR:[Job ] Function:[Setup communication]
10	2.020010	192.168.1.40	192.168.1.10	S7COMM	ROSCTR:[Ack_Data] Function:[Setup communication]
11	2.023543	192.168.1.10	192.168.1.40	S7COMM	ROSCTR:[Job ] Function:[Read var]
12	2.026132	192.168.1.40	192.168.1.10	S7COMM	ROSCTR:[Ack_Data] Function:[Read var]
13	2.036359	192.168.1.10	192.168.1.40	S7COMM	ROSCTR:[Job ] Function:[Read var]
14	2.040122	192.168.1.40	192.168.1.10	S7COMM	ROSCTR:[Ack_Data] Function:[Read var]
15	2.051235	192.168.1.10	192.168.1.40	S7COMM	ROSCTR:[Job ] Function:[Write Var]
16	2.055125	192.168.1.40	192.168.1.10	S7COMM	ROSCTR:[Ack_Data] Function:[Write Var]
17	2.055236	192.168.1.10	192.168.1.40	S7COMM	ROSCTR:[Job ] Function:[Write Var]
18	2.059092	192.168.1.40	192.168.1.10	S7COMM	ROSCTR:[Ack_Data] Function:[Write Var]
19	2.059150	192.168.1.10	192.168.1.40	S7COMM	ROSCTR:[Job ] Function:[Write Var]
20	2.063096	192.168.1.40	192.168.1.10	S7COMM	ROSCTR:[Ack_Data] Function:[Write var]
21	2.063144	192.168.1.10	192.168.1.40	S7COMM	ROSCTR:[Job ] Function:[Write Var]
22	2.067129	192.168.1.40	192.168.1.10	S7COMM	ROSCTR:[Ack_Data] Function:[Write Var]
23	2.067263	192.168.1.10	192.168.1.40	S7COMM	ROSCTR:[Job ] Function:[Read Var]
24	2.071100	192.168.1.40	192.168.1.10	S7COMM	ROSCTR:[Ack_Data] Function:[Read var]

ISO 8073/X.224 COTP Connection-Oriented Transport Protocol

S7 Communication

Header: (Job)

Protocol Id: 0x32

ROSCTR: Job (1)

Redundancy Identification (Reserved): 0x0000

Protocol Data Unit Reference: 65535

Parameter length: 8

Data length: 0

Parameter: (Setup communication)

Function: Setup communication (0xf0)

Reserved: 0x00

Max AmQ (parallel jobs with ack) calling: 1

Max AmQ (parallel jobs with ack) called: 1

```

0000 00 1b 1b 23 eb 3b 90 e6 ba 84 5e 41 08 00 45 00 ...#.;... ^A..E.
0010 00 41 2e 8b 40 00 80 06 00 00 c0 a8 01 0a c0 a8 .A..@... ..
0020 01 28 10 a2 00 66 1d 34 c8 d7 00 02 fc 83 50 18 .(...f.4 .....P.
0030 fa da 83 b6 00 00 03 00 00 19 02 f0 80 32 01 00 .....2..
0040 00 ff ff 00 08 00 00 f0 00 00 01 00 01 07 80 .....

```

Protocol Identification, 0x32 for S7 (s7comm.header.protid), 1 byte

## Wireshark

The S7comm dissector is partially functional.

## Preference Settings

(XXX add links to preference settings affecting how PROTO is dissected).

## Example capture file

- [SampleCaptures/s7comm\\_downloading\\_block\\_db1.pcap](#) s7comm: connecting and downloading program block DB1 into PLC

- [SampleCaptures/s7comm\\_program\\_blocklist\\_onlineview.pcap](#) s7comm: connecting and getting a list of all available block in the PLC
- [SampleCaptures/s7comm\\_reading\\_plc\\_status.pcap](#) s7comm: connecting and viewing the PLC status
- [SampleCaptures/s7comm\\_reading\\_setting\\_plc\\_time.pcap](#) s7comm: connecting, reading and setting the time of the PLC
- [SampleCaptures/s7comm\\_varservice\\_libnodavedemo.pcap](#) s7comm: running libnodave demo with S7-300 PLC, using variable-services with several areas
- [SampleCaptures/s7comm\\_varservice\\_libnodavedemo\\_bench.pcap](#) s7comm: running libnodave demo benchmark with S7-300 PLC using variable-services to check the communication capabilities

## Display Filter

---

A complete list of PROTO display filter fields can be found in the [display filter reference](#)

Show only the S7comm based traffic:

s7comm

## Capture Filter

---

You cannot directly filter S7comm protocols while capturing.

S7comm uses port 102, so it is possible to capture S7comm data by using the capture filter

tcp port 102

## External links

---

- [RFC1006 ISO Transport Service on top of the TCP Version: 3](#), based on ISO 8073
- [RFC905 ISO Transport Protocol Specification ISO DP 8073](#)
- [Siemens - Information about the properties of the S7 protocol](#) *What properties, advantages and special features does the S7 protocol offer* - Siemens Industry Online Support

## Discussion

---

Imported from <https://wiki.wireshark.org/S7comm> on 2020-08-11 23:24:17 UTC