Siemens PLC protocol-S7COMM

tags: plc S7 socket java

Siemens PLC protocol-S7COMM

Brief description

The previous event is connected to the Siemens equipment, which is a shield machine controlled PLC (programmable logic controllers) device. The purpose is to obtain the data status running in the shield machine. It is necessary to send an early warning push through the data status. Here is a brief introduction to the S7COMM protocol and its communication method, which can be learned by combining the packets captured by wireshark in the attached page.

Get to know S7

S7 is an application layer protocol based on TCP/IP, and it cannot simply use Socket to communicate. The following is the network model:

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

OSI layer	Protocol	
1	Physical Layer	Ethernet

Connection process

1. Connect to PLC port 102

2. Connect the ISO layer

3. Establish S7 communication

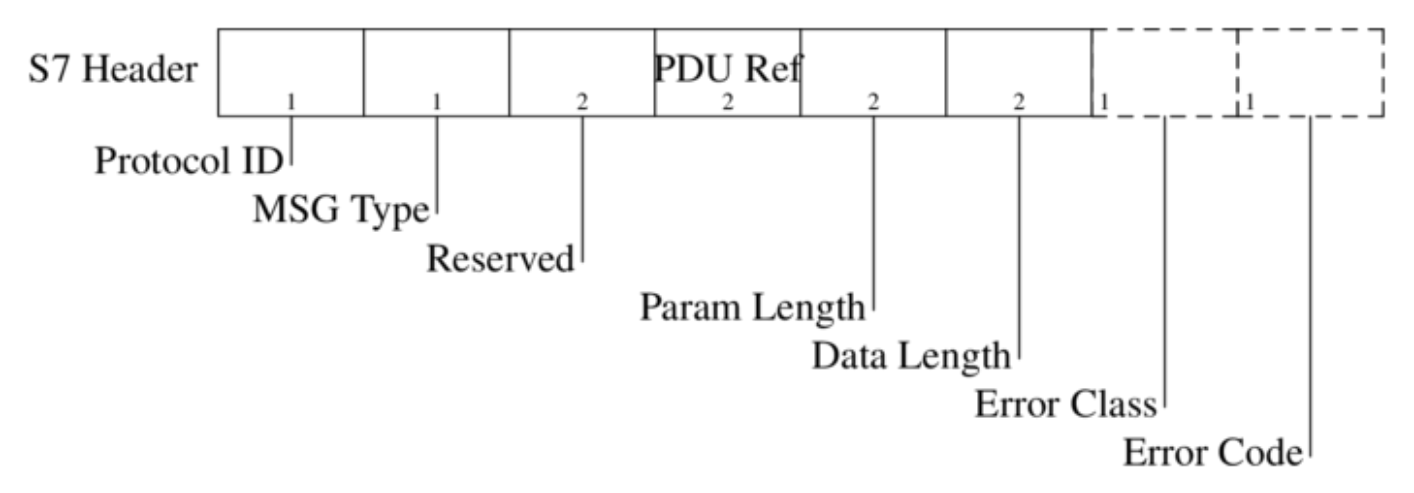
PDU

The main protocol data content is in PDU (Protocol Data Unit), S7 PDU is composed of three parts:

- Header: length information, pdu related and message type constants
- Parameters: According to the message header, content and structure are different
- Data: Data content

Message header

The message header is generally 10-12 bytes, and may also carry two additional error code bytes



- protocol ID (1byte): protocol parameter, usually 0x32
- MSG Type (1byte): Message type

    - 0x01: Job Request, mainly requested by the server, write/read memory, start/stop device, configure session
    - 0x02: Acknowledgement (Ack), which is mainly requested by the device and does not carry data
    - 0x03: response data (Ack-Data), response to the request of 0x01
    - 0x07: Userdata, extended protocol type
- Reserved (2bytes): reserved word, 0x0000
- PDU Ref (2bytes): response sequence number, which can be incremented, the device copies the reply, used to respond to the request content, little endian
- Parameter Length (2bytes): big endian (low order first), parameter length
- Data Length (2bytes): big endian, data area length
- (Error Class) (1byte): The error class will only appear after the response parameter
- (Error code) (1byte): The error code will only appear after the response parameter

Message parameter (Parameter)

Parameter data parameter content is roughly the same, the difference is the content in the request item

- Function Code (1byte): 0x04 read, 0x05 write
- Item Count (1byte): the number of request structures
- Request Item: request data item

Data content

The format of the data content changes according to the addressing mode. Different addressing modes have different structure contents. The structure is as follows:

- Function Code (1byte): 0x04 read, 0x05 write
- Item Count (1byte): the number of request structures
- Request Item: request data item

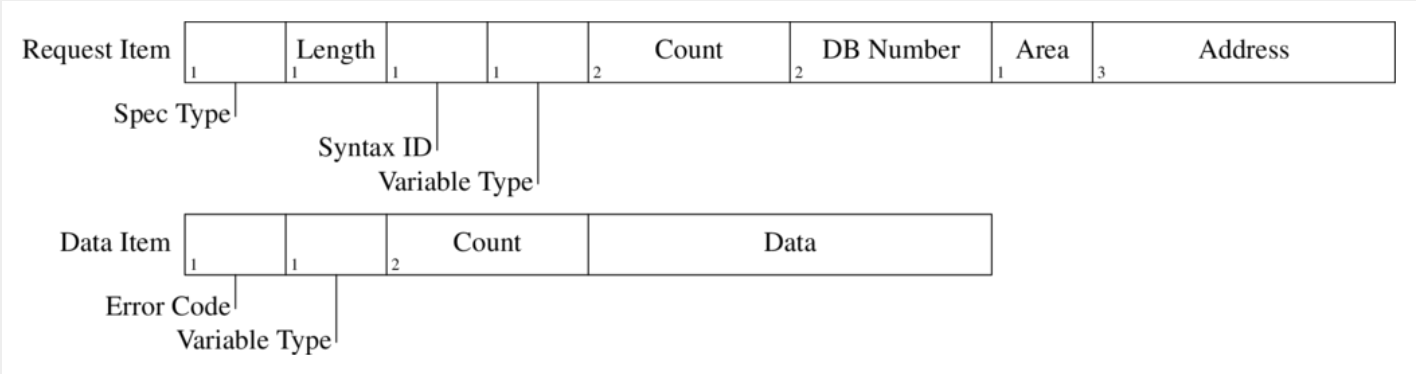
There are three ways of addressing:

- any-type: This is the default addressing mode, used to query any variable. All three parameters (area, address, type) are specified for each addressing variable.
- db-type: Special DB area value
- symbolic-addressing: Special symbol address

Here is a brief introduction `any-type` the way

any-type

The following are the request and data items (**Data Item**)



Request Item

- Specification Type（1byte）：0x12
- Length (1byte): the length of the remaining items
- Syntax ID(1byte): Determine the addressing mode, 0x10 means any-type
- Variable Type(1byte): Data type (REAL, BIT, BYTE, WORD, DWORD, COUNTER, Timer)
- Count (2bytes): You can use a single item structure to select the entire array of similar variables. These variables must have the same type and must be continuous in memory, and the count field determines the size of this array. For reading or writing of a single variable, it is set to 1.
- DB Number（2bytes）：

- Area (1byte): The memory area constant, which can be in**Appendix**Query in
- Address (3bytes): address, such as DBX40.3 calculation 40 * 8 + 3 = 323 hexadecimal: 0x000143
-

Response Data Item data items:

- Error Code (1byte): error code
- Variable Type and Count (1byte 2byte): Variable type and calculation, the same as Request Item
- Data： `len(variable) * count`

> I have used Netty to directly initiate an S7 protocol request, but it failed. If there is something that can be achieved, you can comment below and discuss it together.

Protocol docking

My connection here is to directly use an open source project of a big guy.

s7connector

https://github.com/s7connector/s7connector

Simple usage is also explained in gay

```
1 s7Connector.read(DaveArea.DB, areaNumber, bytes, offset);
```

Parameter Description:

- Parameter 1: Reading method, generally the default is DB area block
- Parameter 2: Area block number
- Parameter 3: Area data type size, int 2 bytes, real 4 bytes
- Parameter 4: Area offset

for example:

```
# DB40_DBD72.0
s7Connector.read(DaveArea.DB, 40, 4, 72);
```

annex

Establish connection example

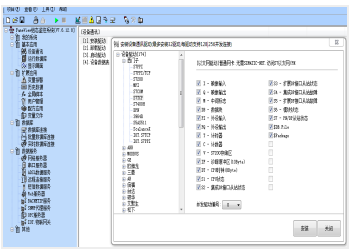
Read example

Qt interprocess communication (1) --------
QProcess

Intelligent Recommendation

Python communicates with Siemens PLC

Install Python-snap7 Open "Run" with win+R, enter cmd, after confirming, enter the DOS command line terminal, enter the following command: pip install python-snap7 areas = ADict({ ‘PE&...



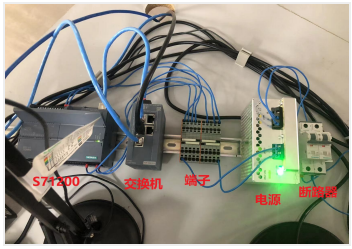
Jiekong configuration Siemens PLC

Jiekong configuration Siemens PLC experience summary 1. Install the driver To create a new project, select Device Communication -

> Install Driver, and select Siemens S7TCP. 2. Add the switch data b...

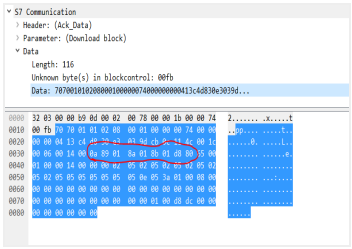
C # Communication Siemens PLC

First, the environment c#： .net core、s7netplus Siemens PLC: S7200Smart Second, the code...



c# connect to Siemens plc

1. Basic configuration: 2. Set the local IPv4 code segment to be the same. 3.c# interface design: (the IP format has been defined and can be seen in the program) 4. Reference xktcomm in the c# manager...



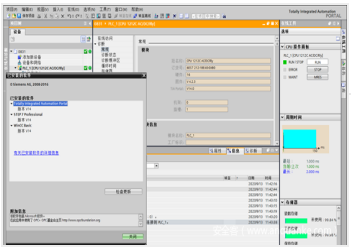
S7comm protocol analysis of the data fields (1)

Experimental environment: Siemens S7-300, CUP 315-2DP, step7 5.6, wireshark Objective: fetch packet reduction codes PLC Using wireshark fetch PC and PLC data transmission, which contains code for the ...

More Recommendation

S7Comm Plus protocol research dynamic debugging two

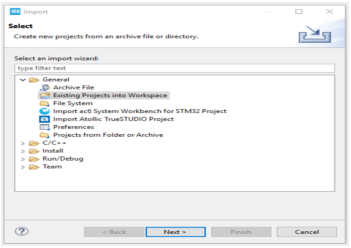
1 Overview Previous articleDescribes the dynamic debugging of OMSp_core_managed.dll to understand the specific communication handshake and encryption authentication



process. Through calculation, the v...

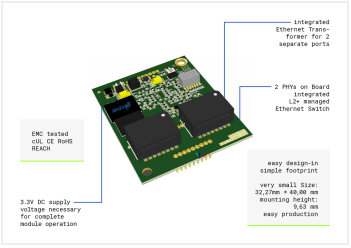
Dynamic debugging of S7C omm Plus protocol research

1 Overview Previous articlePreliminary research on the S7comm-Plus protocol is considered as a theoretical research. This article takes the core communication DLL (OMSp_core_managed.dll) as the target...



SoM IoT multi-protocol module and Siemens PLC S7-1200 communication test guide (bel ow)

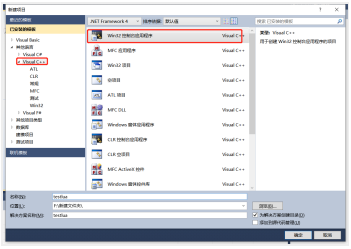
2.2 Import project Use the import dialog of STM Cube IDE to import the projects in the previously unzipped folder into the IDE. When prompted to enter the import type, select "Existing projects i...



SoM IoT multi-protocol module and Siemens PLC S7-1200 communication test guide (on)

1.SoM Introduction to IoT multi-protocol module The SoM IoT multi-protocol solution is a real-time, pre-certified (including sample applications) dual-port Ethernet module solution.

Currently, th...



Lua language with Siemens PLC interacts wi th S7 communication (II, homemade Lua pro tocol library, LUA and C language interactio n)

Connect the upper article, this article is divided into several contents: 1. Use VS to create a dynamic link library. 2. Transplant the Lua Agreement Library. 3. Write the ADD (addition function) in t...

Related Posts

- Siemens PLC protocol-S7COMM-extended
- Use QT to give Siemens PLC to Siemens PLC through the Modbusrtu protocol
- Siemens PLC various communication protocol parsing, analysis
- S7Comm Plus protocol research
- (2) Analysis of S7COMM protocol
- Analysis of data fields in the S7comm protocol
- Each brand PLC protocol reads and writes (Siemens AB Mitsubishi)
- [Industrial control old horse] Detailed explanation of Siemens PLC TCP protocol
- Analysis of Siemens S7comm-plus communication process and replay attack
- Siemens Soft PLC

Popular Posts

- Keywords: packet
- idea integrates tomcat and solves the console garbled problem
- [SOJ 639] trees
- Xiaobaixue front-end ----------CSS basic grammar
- MYSQL date and time type format (detailed introduction)
- Network stream (template transfer)
- Layui jq finds the elements of the first element
- Python Algorithm Learning: Competitive Code Programming-Lanqiao Cup School Trial (Preliminary) Replay
- Freeswitch startup service script
- RISCV's cache

Recommended Posts

- Data Structures and Algorithms basis
- Front End Knowledge Sharing - SHEETJS Usage Experience
- Solve Endnote's reference to the problem without GBT7714 format
- Day04 (array)
- Mac installation git
- Image and large array types
- Convert grayscale image to pseudo-color image-pseudo-color processing
- GHOST blog build
- Android application component - Service
- Krpano tutorial - view tag Chinese description

Related Tags

plc

S7

java

Internet of Things

The internet

Qt

automated industry

Industrial Control Protocol Series

Safety

Industrial control