

# 西门子S7-300, CPU314C-2 DP现货西门子代理

产品名称	西门子S7-300, CPU314C-2 DP现货西门子代理
公司名称	湖南西控自动化设备有限公司
价格	.00/件
规格参数	西门子:S7300 PLC:6ES73146CH040AB0 德国:数字量输入/16 数字量输出, 4模拟量
公司地址	中国（湖南）自由贸易试验区长沙片区开元东路1306号开阳智能制造产业园（一期）4#栋301
联系电话	17838383235 17838383235

## 产品详情

### 当西门子PLC遇到了python

随着工业自动化和信息化的不断发展，PLC（可编程逻辑控制器）已经成为工业自动化领域中不可或缺的关键设备。而在与 PLC 进行通讯时，Python 作为一种功能强大的编程语言，也越来越受到工程师们的青睐。因为 Python 在科技计算、数据处理、可视化等方面有着出色的表现，并且通过 Python 与 PLC 进行通讯也是一种高效、灵活的手段。本次我们将介绍如何使用 Python 与西门子 PLC 进行通讯的方法和注意事项，掌握这种技能将有助于在工业控制及信息化方面有所应用，具有广泛的应用前景。

1

#### 西门子 PLC 与其通讯协议

西门子 PLC 是工控领域广泛使用的一种控制器，其硬件结构主要由 CPU、IO 模块等组成。PLC 掌握了现代工业自动化生产中重要地位，其大量使用使得自动化生产得以高效、地进行。PLC 的操作系统是专门为自动化控制而设计的，它集成在控制器的标准芯片上，因此 PLC 具有较高的稳定性和可靠性。

PLC 通讯协议则是指人机接口（HMI）或者第三方系统与 PLC 进行通信时所采用的通讯方式，包括西门子 PLC 主站协议等。通讯协议的选择取决于应用系统的需求和具体情况。

其中，西门子 PLC 主站协议是西门子 PLC

通讯协议的一种，是相对完整的一种通讯协议。它可以用于多个平台和设备上，如 PC 端、HMI 等，实现数据的采集、监控、控制等功能。通过这种协议，可以从 PLC 读取数据、写入数据、触发控制命令等，实现 PLC 与其他系统之间高效的数据传递和应用交互。

2

## Python 语言基础

Python 语言基础主要包括以下几部分：

1. Python 基本语法：包括 Python 的关键字、语句、注释、缩进等基本语法规则，这些语法规则是编写 Python 代码的基础；

2. Python 数据类型：Python

有多种数据类型，包括数字、字符串、列表、元组、字典等，这些数据类型是 Python 编程中的基础；

3. Python 条件语句：Python 的条件语句包括 if、elif、else 语句，这些语句可以根据条件来控制程序的执行；

4. Python 循环语句：Python 的循环语句包括 for、while 语句，这些语句可以重复执行指定的代码块。

除此之外，Python 还有丰富的标准库和第三方库，这些库提供了很多功能强大、易用的函数和工具，用于处理各种场景下的数据类型和任务。在实现 Python 与 PLC 通讯时，需要使用到相关的库函数，如 socket、struct、Pycomm 等，因此，对于 Python 语言和库函数的掌握也非常重要。

3

## Python 与 PLC 通讯

Python 与 PLC 通讯可以通过多种方式，如串口通讯、以太网通讯等。在 PLC 与 Python 之间建立通讯连接后，就可以实现数据的传输和控制信号的交互。

在 Python 与 PLC 通讯中，常用的协议包括 MODBUS、OPC UA、S7Comm 等。对于西门子 PLC，通常使用 S7Comm 协议进行通讯，其使用 TCP/IP 协议进行数据传输。为了使用 S7Comm 协议，需要使用相关的库函数，如 Python-snap7、Pycomm 等。

在使用 Python 与 PLC 通讯时，通常需要进行以下步骤：

1. 建立连接

2. 完成 S7Comm 握手

3. 读取或写入数据

4. 断开连接

需要注意的是，在 Python 与 PLC

通讯过程中，需要考虑到通讯的稳定性和数据的正确性，避免出现数据传输错误或通讯中断等问题。

4

## 实际案例和应用场景

在实际应用中，使用 Python 与 PLC 通讯，可以实现工业自动化、远程监控、数据采集等功能。下面以 snap7 库为例，介绍 Python 与西门子 PLC 通讯的具体实现过程。

### 步骤 1：安装 snap7 库

在 Python 环境中，使用 pip 命令安装 snap7 库：

```
pip install snap7
```

### 步骤 2：建立连接

使用 snap7 库中的 client 库函数建立连接，指定 PLC 的 IP 地址和端口号：

```
import snap7.client as clientplc = client.Client()plc.connect('192.168.1.1', 0, 1)
```

第一个参数 192.168.1.1：表示 IP 地址

第二个参数 0：表示 Rack 号（机架号），数字非 0 即 1

第三个参数 1：表示 Slot 号（CPU 槽号），请去 plc 控制柜查看

### 步骤 3：读取数据

使用 snap7 库中的 db\_read 函数读取 PLC 中的数据，指定数据类型、数据块地址、数据地址和数据长度：

```
data = plc.db_read(1, 0, 0, 10)
```

对于 plc.db\_read(1, 0, 0, 10) 这句代码，其中每个数字的含义如下：

第一个参数 1：表示需要读取的数据块的数据块号，也可以使用区域标识符来指定，如 plc.db\_read("DB1", 0, 0, 10)。

## 第二个参数

0：表示需要读取的数据块内的起始地址，即需要读取数据块的哪个地址开始读取。此处设为 0，表示从数据块中的个地址（也就是起始地址）开始读取数据。

第三个参数 0：表示读取的数据类型。0

表示位（boolean）类型。可以根据读取的数据类型对应不同的数字，如 1 代表 byte（8 位）类型，2 代表 word（16 位）类型，3 表示双字节（double word）类型，4 表示浮点型（float）类型，5 表示双精度浮点型（double）类型。

第四个参数 10：表示需要读取多少个数据，即读取的数据长度。此处设为 10，表示需要读取 10 个位（boolean）类型的数据，读取的范围是从数据块的地址 0 开始，读取 10 个地址的数据。

## 步骤 4：写入数据

使用 snap7 库中的 db\_write 函数向 PLC

中写入数据，指定数据类型、数据块地址、数据地址、数据长度和数据内容：

```
plc.db_write(1, 0, 0, b'\x01\x02\x03\x04\x05\x06\x07\x08\x09\x0A')
```

个参数 1：表示要写入的数据块在 PLC 中的编号，即 DB1。

第二个参数 0：表示写入开始的偏移量（即在 DB1 中的起始地址，从 0 开始）。

第三个参数 0：表示要写入的数据在数据块中的偏移量（个数据的起始地址为 0）。

第四个参数 b'\x01\x02\x03\x04\x05\x06\x07\x08\x09\x0A'：表示要写入的数据。在此例中，写入的是一个 10 字节的二进制数据，其中每个字节均为 \x01 至 \x0A。

## 步骤 5：关闭连接

通讯完成后，使用 snap7 库中的 disconnect 函数关闭连接：

```
plc.disconnect()
```

通过以上步骤，可以实现 Python 与西门子 PLC 通讯，并实现数据的传输和控制信号的交互。在实际应用中，可以根据具体情况选择合适的通讯协议和通讯方式，并进行相应的参数配置和数据解析。

5

## 拓展应用

下面介绍应用场景，以及如何在这些场景中使用 Python 与西门子 PLC 通讯来实现特定的功能。

1.生产数据采集与监控：在工业生产过程中，需要对设备状态、生产数据等信息进行实时采集与监控，以实现生产过程的优化和效率提升。可以使用 Python 与西门子 PLC 通讯来读取各种传感器数据、设备状态等信息，并进行实时处理和分析。

2.自动化控制：对于某些生产过程中需要进行自动化控制的设备，比如自动化包装、装配设备等，可以通过 Python 与西门子 PLC 通讯来实现对设备的远程控制，包括开/关机、设备速度控制、工作状态监测等。

3.工业互联网应用：随着工业互联网的快速发展，工业设备的联网需求也越来越高。可以使用 Python 与西门子 PLC 通讯来实现工业设备与互联网的连接，包括设备数据的上传、云端数据的下载、实时监控等功能。

4.机器学习应用：在某些场景下，需要使用机器学习算法对生产数据进行分析 and 处理，以实现生产过程的智能化管理。可以使用 Python 与西门子 PLC 通讯来读取生产数据，并对其机器学习算法的训练和优化，以实现更好的生产效率和品质管理。

在拓展应用时，需要注意使用 Python 与西门子 PLC 通讯时安全性和数据准确性的保证，同时根据特定的应用场景和需求进行具体的开发和实现。

6

## 总结

在与 PLC 进行通讯时，需要考虑以下几点：

1.通讯协议和数据类型：不同厂商的 PLC 通常采用不同的通讯协议和数据类型，需要选用适合的通讯方式和数据类型，并对其进行了解和熟悉。

2.连接的建立与关闭：在与 PLC 进行通讯时，需要先建立连接，进行数据读写操作后，再关闭连接，保证连接的稳定性。

3.数据的读取和写入：对于数据的读取，可以通过指定 PLC 中数据的类型和地址，从而读取到指定地址上的数据。对于数据的写入，需要按指定的格式将数据传入 PLC。

4.错误处理和异常处理：在进行数据读写的过程中，可能会遇到通讯错误、超时、设备离线等异常情况，需要进行异常处理，避免程序崩溃或者数据错误。

总之，与 PLC 进行通讯需要注意通讯协议和数据类型选择、连接的建立和关闭、数据的读写方式、异常处理等问题。在实际应用中可以根据具体需求和使用的 PLC 型号选择适合的通讯方式和库，并进行适当的调试和优化，保证通讯的稳定性和数据的准确性。