

---

# Remote interfaces

The SMD4 offers 3 communication interfaces, USB, Ethernet and Serial. The serial interface is dual mode, and can be configured for RS232 or RS485. Multiple interfaces can be connected and in use at the same time. All interfaces use a text-based communication protocol described in section [Communications protocol](#).

This section may only be relevant if you are using terminal software or writing your own application. AML Device control software interacts with the SMD4 in the same way and requires no specialist knowledge to use. A C# API is also provided, allowing you to easily integrate communication with the SMD4 into your own C# .NET application.

## USB

A reversible USB-C connector is provided.

The SMD4 appears as a virtual COM port on your PC, and will be assigned a designation such as "COM1". Each unique SMD4 that is connected to the PC will be assigned a new designation. Upon reconnecting a previously known SMD4, it will typically assume the same designation as when last used, however, this behaviour cannot be guaranteed.

The virtual COM port does not require any particular configuration to function correctly. Typical settings such as baud rate and parity are not applicable.

Most programming languages include easy built-in methods to read and write data to and from a COM port, for example, the `SerialPort` class in the `System.IO.Ports` namespace in C#, or `pySerial` in Python. Alternatively, use a terminal program such as Tera Term to manually type commands and see responses. Remember to set line termination which is a carriage return followed by line feed (in hexadecimal, 0x0D, 0x0A)

## Serial

A pair of RJ45 connectors are provided allowing multiple devices to be bussed together if required. This is discussed in section [RS232/RS485](#). Do not confuse these connectors with the Ethernet connector.

The serial interface can be configured for RS232 or RS485.

If connecting to a PC, a serial port is required. Modern PCs seldom include RS232/485 ports, so typically a USB adapter or add-in card is required. This appears on the PC as one or more COM ports.

Unlike the virtual COM port provided via the USB interface, the COM port must be configured correctly:

- Baud to match that set on the SMD4
- 1 stop bit

- No parity
- No flow control

Thereafter, the same notes as for USB apply.



Configure the mode of operation (RS232 or RS485) before plugging a connector in. Do not make changes to the mode without first disconnecting both connectors. Undefined behaviour will result if the SMD4 is connected to an RS485 network when in RS232 mode or vice-versa.

## Ethernet

A standard 10/100M ethernet port is provided. It features Auto MDI-X so straight or crossover cable may be used.

The factory default setup enables DHCP, so under normal circumstances, this will assign the network configuration to the SMD4. Alternatively, use another interface, such as USB, to configure the ethernet port.

As with COM ports, most programming languages provide easy means to open a socket and communicate with network devices. Alternatively, use a terminal program such as PuTTY to manually type commands and see responses.

The SMD4 uses port 11312 for its text-based communication protocol. Only one connection at a time is supported, additional connections will be refused.

## SSDP

The SMD4 is discoverable on the network using SSDP (Simple Service Discovery Protocol). This makes it possible to plug the SMD4 into a network, then discover its IP address and so make a connection to it. Alternatively, use another interface, for example, USB to discover the network configuration.

### Supported search targets (queries)

All – Not SMD4 specific, all devices on the network supporting this command will respond including SMD4

- ["ssdp:all"](#)

JPNP Root Device, as above

- ["upnp:rootdevice"](#)

Device – Specify domain name, device type and version to match the SMD4 and only SMD4s will respond:

- ["urn:schemas-arunmicro-com:device:StepperMotorDrive:1"](#)

- "urn:" SSDPR\_SCHEMA\_DOMAIN\_NAME ":device:" SSDPR\_SCHEMA\_DEVICE\_TYPE ":" SSDPR\_SCHEMA\_VER"

UUID – Specify a UUID and only one device having that UUID should respond. Use **this command** to get the UUID of your SMD4.

- "uuid:" UUID

Where:

- SSDPR\_SCHEMA\_DOMAIN\_NAME "schemas-arunmicro-com"
- SSDPR\_SCHEMA\_DEVICE\_TYPE "StepperMotorDrive"
- SSDPR\_SCHEMA\_VER "1"

## Response

The SMD4 always responds in this way:

```
HTTP/1.1 200 OK\r\n
CACHE-CONTROL:max-age=120\r\n
DATE:\r\n
EXT:\r\n
LOCATION:http://10.0.97.6:80/desc.xml\r\n
SERVER:OS/version product/version\r\n
ST:uuid:28254095-7194-11ee-a857-44b7d0c76aa3\r\n
USN:uuid:28254095-7194-11ee-a857-44b7d0c76aa3::urn:schemas-arunmicro-com:device:StepperMotorDrive:1\r\n
\r\n
```

Note:

- Response always starts with http ok
  - CACHE\_CONTROL, DATE and EXT are not used. CACHE-CONTROL has a default value but no meaning
  - LOCATION is a URL to an xml file containing information as required by UPnP. The SMD4 does not implement a webserver to allow this file to be retrieved, however the root of the URL gives the IP address of the device which may be used to address future communications.
  - SERVER:OS not implemented
  - ST; this echos the search target sent in the query
  - USN: gives the uuid and device type info
-