

ESP-IDF BT-SPP-ACCEPTOR demo

This is the demo for user to use ESP_APIs to create a **Serial Port Protocol (SPP)** acceptor and we aggregate **Secure Simple Pair (SSP)** into this demo to show how to use SPP when creating your own APPs.

How to use example

Hardware Required

This example is designed to run on commonly available ESP32 development board, e.g. ESP32-DevKitC.

To operate it should be connected to an SPP Initiator running on a smartphone or on another ESP32 development board.

Configure the project

```
idf.py menuconfig
```

Build and Flash

Build the project and flash it to the board, then run monitor tool to view serial output:

```
idf.py -p PORT flash monitor
```

(Replace PORT with the name of the serial port to use.)

(To exit the serial monitor, type `Ctrl-]`.)

See the [Getting Started Guide](#) for full steps to configure and use ESP-IDF to build projects.

Example Description

After the program started, `bt_spp_initiator` will connect it and send data.

Trouble shooting

- Acceptor is expected to start before `bt_spp_initiator` start.
- To see the information of data, users shall set `SPP_SHOW_MODE` as `SPP_SHOW_DATA` or `SPP_SHOW_SPEED` in code(should be same with `bt_spp_initiator`).
- We also show the Security Simple Pair in this SPP demo. Users can set the IO Capability and Security Mode for their device (security level is fixed level 4). The default security mode of this demo is `ESP_SPP_SEC_AUTHENTICATE` which support MITM (Man In The Middle) protection. For more information about Security Simple Pair on ESP32, please refer to [ESP32 SSP](#).

FAQ

Q: How many SPP servers does ESP32 support? A: For now, the maximum number of SPP servers is 6, which is limited by the maximum number of SDP records. When the SPP server is successfully started, the unique SCN (Server Channel Number) will be mapped to the SPP server.