The demo is used as follows:
Esp8266 and stm32 to connect through the spi, then esp8266 io0 connected to the stm32 interrupt pin. After the start of esp8266 and stm32, the stm32 can send AT commands to the serial port, stm32 received data, will be transmitted through the spi to esp8266, esp8266 will process the received command, and the results of the implementation of the command sent to stm32 spi, stm32 from Esp8266 read, print to the serial port.

Esp8266 and mcu of the connection:
Esp8266 and mcu interaction, the use of spi and interrupt pin way, that is, CS, MOSI, MISO, SCK and intr_pin (demo
In esp8266 for io0). Which CS, MOSI, MISO, SCK spi communication used by the line, specifically to see the spi protocol.
Intr_pin: When esp8266 data to take the initiative to send to mcu, by pulling down the line to achieve the purpose of notice mcu. So the mcu pin connected to this line need to monitor the way the interrupt mcu.

work process:
When Mcu is actively sending:
Mcu side:
1. mcu first address to send a byte of esp8266 0x01 data length, the format is: 0x01 data_len
2. mcu then send data to the esp8266, the format is: 0x02 0x00 data1 data2 ... ... data32, where the second byte 0x00 not currently used
Esp8266 side:
1. When mcu finished the first step above, esp8266 will generate an interrupt SPI_SLV_WR_STA_DONE, and then read out the length of data_len and save.
2. When the mcu finished the second step above, esp8266 will generate an interrupt SPI_SLV_WR_BUF_DONE, then esp8266 read out 32 bytes, and data_len bytes passed to the AT module, which data_len bytes for the effective word Section

ESP8266 active transmission:
Esp8266 side:
1. The Esp8266 first populates the spi data register with the data, writes the data length to the SPI_RD_STATUS register, and lowers the intr_pin pin to inform the mcu.
2. After reading the data, esp8266 generates an SPI_SLV_RD_BUF_DONE interrupt and clears the SPI_RD_STATUS register to wait for the next transmission.

Mcu side:
1. When the mcu side intr_pin interrupt received, send 0x04 command, query esp8266 data length data_len, format 0x04 0xFF, mcu send a second in itself, will also receive the return of esp8266 data length
2. Then mcu send 0x03 command to read data, in which the former data_len data for the valid data, the format 0x03 0x00 0xFF 0xFF ... ..., when mcu transmission of the first 0xFF, began to receive esp8266 return data.