ALGORITHM

Master:

1. Initialize the ESP8266 as a master and set up the necessary parameters for communication. Such as the mode, Wi-Fi credentials, and IP address.

2. Establish a connection with the slave ESP8266 device by creating a TCP socket by sending "AT+CIPSTART" command through the serial interface. Specify the slave ESP8266's IP address and port number in the command. Wait for a response from the ESP8266 module to confirm that the connection has been established.

3. Prepare the data to be sent in the desired format (e.g., a string or a binary buffer). Send the data to the slave ESP8266 by writing it to the socket using the "AT+CIPSEND" command. Wait for a response from the ESP8266 module to confirm that the data has been sent successfully.

4. Wait for incoming data by reading from the socket using the "AT+CIPRXGET" command. Parse the received data in the desired format (e.g., as a string or a binary buffer).

5. Close the socket connection when communication is complete.

Slave:

1. Initialize the ESP8266 as a slave and set up the necessary parameters for communication. Set up the necessary parameters for communication, such as the mode, Wi-Fi credentials, and IP address.

2. Create a server socket and bind it to the desired port number by sending the "AT+CIPSERVER" command.

3. Wait for incoming connection requests by reading from the socket using the "AT+CIPSTATUS" command. If a connection request is received, accept it by sending the "AT+CIPACCEPT" command. Create a socket for communication with the master ESP8266 by sending the "AT+CIPSEND" command.

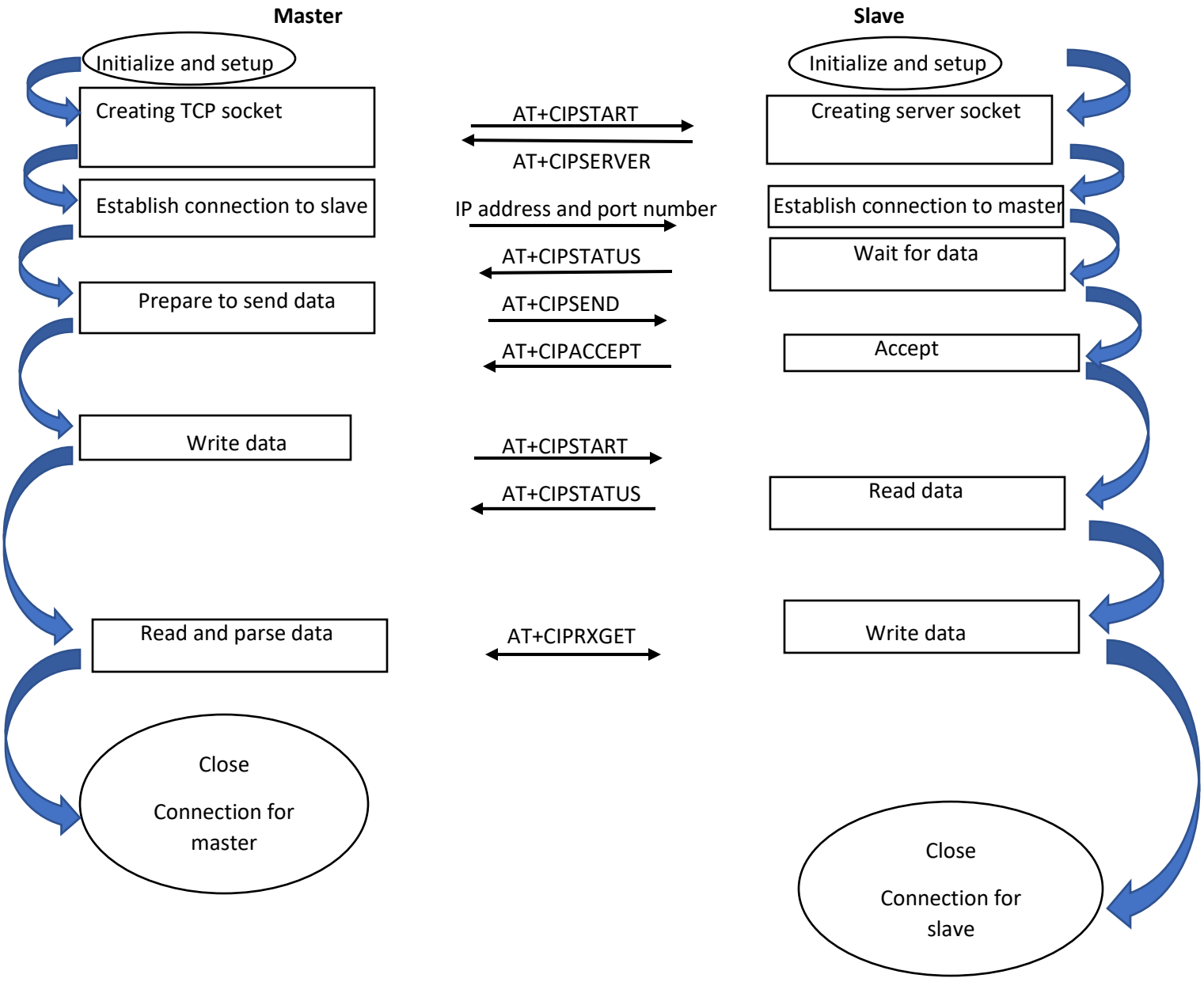
4. Accept incoming connections from the master ESP8266 device and create a socket for communication. Wait for the master ESP8266 to send the "AT+CIPSTART" command to initiate a connection. Send the "AT+CIPSTATUS" command to confirm that the connection has been established.

5. Wait for incoming data by reading from the socket using the "AT+CIPRXGET" command. Parse the received data in the desired format (e.g., as a string or a binary buffer).

6. Send data to the master ESP8266 by writing to the socket.

7. Close the socket connection when communication is complete.

FLOW CHART


Master
Slave
Initialize and setup
Initialize and setup
Creating TCP socket
Creating server socket
AT+CIPSTART
AT+CIPSERVER
Establish connection to slave
Establish connection to master
IP address and port number
AT+CIPSTATUS
Wait for data
Prepare to send data
AT+CIPSEND
AT+CIPACCEPT
Accept
Write data
AT+CIPSTART
AT+CIPSTATUS
Read data
Read and parse data
AT+CIPRXGET
Write data
Close
Connection for
master
Close
Connection for
slave