UDP

UIPUDP::begin(uint16_t port):This function initializes some associated data and creates a UDP connection on the specified port.

UIPUDP::stop():This function removes a UDP connection, releases any related memory blocks, and resets the relevant variables to stop and clean up the connection.

UIPUDP::beginPacket(IPAddress ip, uint16_t port):This function builds a UDP packet for transmission to a given IP address and port. It establishes a new UDP connection or makes use of an existing one, allots memory for the departing packet, and configures the appropriate parameters for transmission.

UIPUDP::beginPacket(const char *host, uint16_t port):This function resolves a host name to an IP address by using a DNS query, and then it uses the resolved IP address and specified port number to call the other version of the "beginPacket" function.

UIPUDP::endPacket():The 'beginPacket' function was used to prepare a UDP packet, which this code completes and sends. The relevant flags and parameters are set, the packet is resized, periodic chores for the UDP connection are completed, and if the packet has a length more than zero, it is sent.

UIPUDP::write(uint8_t c)& UIPUDP::write(const uint8_t *buffer, size_t size):
These functions allow writing data into the outgoing UDP packet. The first function writes a single byte, while the second function writes multiple bytes from a buffer.

UIPUDP::available():This function checks if data is available in the incoming UDP packet by getting the packet size. It performs necessary tasks such as updating the Ethernet interface and returning the packet size.

UIPUDP::read() & UIPUDP::read(unsigned char* buffer, size_t len): These functions read data from incoming UDP packets. The first function reads a single byte, while the second reads multiple bytes from the packet into the buffer.

UIPUDP::peek():This function makes it possible to see the next byte of an incoming UDP packet without dropping it. It performs necessary tasks such as updating the Ethernet interface and checking if bytes are available to read.

UIPUDP::flush(): this function discards the current UDP packet and frees the memory allocated to it. It performs necessary tasks such as updating the Ethernet interface and setting the incoming packet to "NOBLOCK".

UIPUDP::remoteIP() & UIPUDP::remotePort():these functions retrieve information about the remote host that sent the packet to the present, such as its IP address and port number. They check if the connection has been established and return the appropriate values based on the status of the connection.

UIPUDP_appcall(void): This function handles the processing of UDP data. It allocates memory for incoming packets, copies the data into the allocated memory, and processes and sends outgoing packets.

UIPUDP::_send(uip_udp_userdata_t *data):This function handles the sending of the outgoing UDP packet. It adds the ARP header to the packet, checks if the packet is an ARP packet, and sets flags and fields to indicate the status of the outgoing packet. Finally, it calls the function to send the packet over the network.