1. **uint8\_t Hex\_To\_Num(char ch)**:

**Chức năng**: chuyển từ kí tự kiểu hexa nhận được (ví dụ ‘0’ – ‘9’, ‘a’ – ‘f’ hoặc ‘A’ – ‘F’) sang kí tự số (‘0’ → 0, ‘1’ → 1, … , ‘A’→ 10, ‘B’ →11). Vì khi nhận được dữ liệu trên USART thì chỉ nhận được kí tự số, không phải số.

**Input**: kí tự kiểu char

**Output:** số nguyên không dấu 8 bit

1. **uint8\_t Char\_Is\_Hex(char ch):**

**Chức năng**: Xét xem kí tự số có phải thuộc hexa hay không (nằm trong các kí tự 0 - 9, a - f hoặc A - F).

**Input**: kí tự kiểu char

**Output:** 1 nếu thuộc kiểu hexa và 0 nếu không thuộc kiểu hexa.

1. **uint32\_t Cal\_Hex\_Num(char\* ptr, uint8\_t\* count):**

**Chức năng**: Trả về kết quả là một số nguyên không dấu 32 bit từ một chuỗi (ptr) đưa vào, đồng thời biến count sẽ lưu giữ số kí số của ptr.

**Input**: char\* ptr - con trỏ chỉ tới chuỗi cần xử lý

uint8\_t\* count: con trỏ kiểu số nguyên không dấu 8 bit lưu giữ số kí số

**Output:** Kết quả theo dạng thập phân của chuỗi ptr.

Ví dụ: với ptr là con trỏ chỉ tới chuỗi “12345” => giá trị trả về từ hàm là 0x12345, count sẽ có giá trị là 5

1. **void ParseIP(char\* ip\_str, uint8\_t\* arr, uint8\_t\* cnt):**

**Chức năng:** Phân tách địa chỉ IP dưới dạng chuỗi nhận được từ USART và lưu vào mảng arr. Variable cnt dùng để tính số kí tự IP nhận được, biến này được dùng trong phân tích 1 chuỗi ký tự nhận được có chứa IP để đưa con trỏ tới vị trí cần xử lý tiếp theo sau chuỗi IP. Ví dụ với lệnh AT+CWJAP\_CUR? Response là +CWJAP\_CUR:<ssid>,<bssid>,<channel>,<rssi>. Giả sử ssid = “abc”, <bssid> = “ca:d7:19:d8:a6:44”, <channel> = 2, <rssi> = 4, ta có response là

+CWJAP\_CUR:”abc”,”ca:d7:19:d8:a6:44”,2,4. Đặt ptr là con trỏ tại vị trí a trong SSID, khi phân tích lấy được ssid => con trỏ sẽ được cộng lên 2, xử lý tiếp để bỏ qua chuỗi “,” tới vị trí MAC đầu tiên (ca), lúc này cnt sẽ lưu số ký tự của MAC là