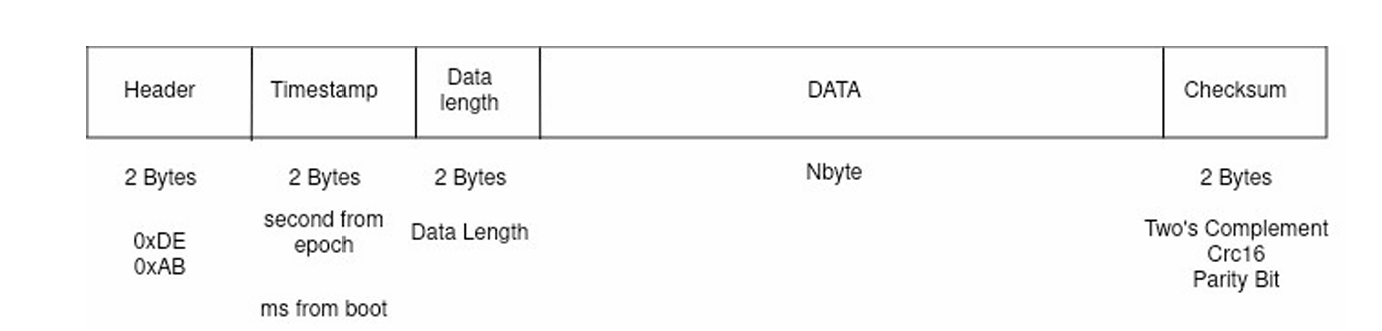
**Tài liệu thiết kế firmware**

**1 Tại sao lại sử dụng kiểu dữ liệu nhị phân thay vì mã ASCII**

* Dữ liệu nhị phân thường sẽ nhỏ gọn hơn so với mã ASCII.
* Do dữ liệu được truyền trực tiếp dưới dạng byte thô, quá trình truyền và xử lý sẽ nhanh hơn đáng kể so với việc truyền các ký tự ASCII.
* Với dữ liệu nhị phân, bạn có thể dễ dàng tích hợp các thuật toán kiểm tra độ chính xác (checksum, CRC) vào mỗi gói tin một cách hiệu quả, giúp phát hiện và sửa lỗi nhanh chóng.

**2 Khung truyền dữ liệu**

****

**Bao gồm:**

* Header để xác định điểm bắt đầu gói tin.
* Thời gian gửi gói tin (Timestamp).
* Độ dài dữ liệu (Data Length).
* Dữ liệu (DATA).
* Checksum để kiểm tra tính toàn vẹn dữ liệu.

**3 Tài liệu diễn giải**

**3.1 Layer sofware**

Application

Utils

Protocol

Driver

* **Application Layer** (Tầng ứng dụng): Tầng dùng để gửi các message.
* **Protocol Layer** (Tầng giao thức): Tầng dùng để đóng gói các message.
* **Driver/Utils Layer** (Tầng trình điều khiển và tiện ích): Tầng dùng để giao tiếp trực tiếp với các API.

**3.2 Sơ đồ tổng quát của firmware**

**3.3 Phân tích mã**

**3.3.1 Application**

Tầng này chủ yếu là sẽ gửi các message và tần số gửi message

**void** **send\_packet\_data**(packet\_t\* packet, **void**\* data, uint16\_t data\_size);

**void** **send\_date\_data**(uint32\_t days,uint32\_t month, uint32\_t year);

**void** **send\_time\_data**(uint8\_t hour, uint16\_t minute, uint16\_t second);

**void** **send\_adc\_data**(uint32\_t sample\_count, uint16\_t value);

**void** **send\_string\_data**(uint16\_t string\_len, uint8\_t \*string);

**void** **send\_button\_data**(uint8\_t button\_id, uint16\_t button\_state);

**void** **send\_temperature**(uint16\_t mcu\_temperature\_in\_c);