Projet_VAE_Nice\terminal_patient.h

```cpp
// terminal_patient.h en code C++
#ifndef TERMINAL_PATIENT_H
#define TERMINAL_PATIENT_H

#include <Arduino.h>
#include <SPI.h>
#include <LoRa.h>
#include <TFT_eSPI.h>
#include <WiFi.h>
#include <HTTPClient.h>
#include <ArduinoJson.h>
#include <Preferences.h>

// Configuration LoRa
#define LORA_FREQUENCY 868E6
#define LORA_SYNC_WORD 0x12
#define LORA_TX_POWER 20
#define LORA_SPREADING_FACTOR 7

// Définitions des besoins
enum Besoin {
  BESOIN_DOULEUR = 1,
  BESOIN_TOILETTE = 2,
  BESOIN_REPOSITION = 3,
  BESOIN_QUESTION = 4,
  BESOIN_EAU = 5,
  BESOIN_URGENCE = 99
};

// Structure des messages LoRa
#pragma pack(push, 1)
struct LoraMessage {
  uint16_t terminalId;
  uint8_t chambreId;
  uint8_t besoin;
  uint8_t priorite;
  uint32_t timestamp;
  uint16_t checksum;
};
#pragma pack(pop)

class TerminalPatient {
public:
  TerminalPatient(uint16_t id, uint8_t chambre);
  void begin();
  void loop();

private:
  // Hardware
  uint16_t terminalId;
  uint8_t chambreId;
```

```
  TFT_eSPI tft;
  Preferences preferences;

  // Communication
  bool sendLoRaMessage(Besoin besoin, uint8_t priorite);
  bool sendHTTPBackup(const String& payload);
  uint16_t calculateChecksum(const uint8_t* data, size_t len);

  // Interface
  void drawMainMenu();
  void handleTouch();
  void handleEmergencyButton();
  void showConfirmation(Besoin besoin);
  void updateStatusLED(bool confirmed);

  // Logique
  void logEvent(const String& event);
  bool isNetworkAvailable();

  // Touch screen calibration
  int16_t touchX, touchY;
  bool screenTouched;

  // Debounce
  unsigned long lastEmergencyPress = 0;
  const unsigned long DEBOUNCE_DELAY = 500;
};

#endif

```