Task: Upgrade Codebase to AGHAF 5.0.9

Use the prompt below with an AI coding assistant on any C/C++ project upgrading from AGHAF 4.x to AGHAF 5.0.9.

```
# TASK: Upgrade Codebase to AGHAF 5.0.9

## 1. Context & Objective
You are upgrading a C/C++ project from AGHAF 4.x to **AGHAF 5.0.9**.
AGHAF is the hardware abstraction library used for automotive/diagnostic communication (CAN,

Your goal is to search the codebase for AGHAF API function calls and callback definitions, a

---

## 2. API Changes & Function Modifications

### A. Lifecycle Management (`main` function)
In AGHAF 5.0.9, the library requires explicit subsystem initialization and deinitialization.

- **Search**: Locate the application's `main` function (entry point).
- **Action**:
  1. Call `AGHAF_initialize();` before invoking any other AGHAF API function (e.g., before
  2. Call `AGHAF_uninitialize();` at application exit right before returning from `main()`.

```cpp
int main(int argc, char* argv[]) {
  // Initialize AGHAF before any other AGHAF API calls
  AGHAF_initialize();
  AGHAF_CAN_v2_useAPI();

  // ... application execution ...

  // Deinitialize AGHAF before exit
  AGHAF_uninitialize();
  return 0;
}
```

B. Device Discovery & Retrieval Functions

Several device discovery functions in AGHAF 5.0.9 changed their return type to `AGHAF_Status` and now return the device handle or count via an output pointer parameter.
```

1. `AGHAF_getDeviceBySN`
   - **Search**: Grep for calls to `AGHAF_getDeviceBySN` across the codebase.
   - **Signature Change**:
     - **Old (4.x)**: `AGHAF_Device AGHAF_getDeviceBySN(const char* productNumber, const char* serialNumber);`
     - **New (5.0.9)**: `AGHAF_Status AGHAF_getDeviceBySN(char const* productNumber, char const* serialNumber, AGHAF_Device* device);`
   - **Migration**:

```
// OLD (4.x):
AGHAF_Device deviceHandle = AGHAF_getDeviceBySN(productNo, serialNo);
if (deviceHandle == nullptr) {
  // error handling
}

// NEW (5.0.9):
AGHAF_Device deviceHandle = nullptr;
AGHAF_Status status = AGHAF_getDeviceBySN(productNo, serialNo, &deviceHandle);
if (status != AGHAF_STATUS_OK) {
  // error handling
}
```

2. `AGHAF_getDeviceByHardID`
   - **Search**: Grep for calls to `AGHAF_getDeviceByHardID`.
   - **Signature Change**:
     - **Old (4.x)**: `AGHAF_Device AGHAF_getDeviceByHardID(uint8_t const* hardwareID);`
     - **New (5.0.9)**: `AGHAF_Status AGHAF_getDeviceByHardID(uint8_t const* hardwareID, AGHAF_Device* device);`
   - **Migration**:

```
// OLD (4.x):
AGHAF_Device deviceHandle = AGHAF_getDeviceByHardID(hardwareID);
if (deviceHandle == nullptr) {
  // error handling
}

// NEW (5.0.9):
AGHAF_Device deviceHandle = nullptr;
AGHAF_Status status = AGHAF_getDeviceByHardID(hardwareID, &deviceHandle);
if (status != AGHAF_STATUS_OK) {
```

```
  // error handling
}
```

3. `AGHAF_getDeviceCount`
   - **Search**: Grep for calls to `AGHAF_getDeviceCount`.
   - **Signature Change**:
     - **Old (4.x)**: `uint32_t AGHAF_getDeviceCount(void);`
     - **New (5.0.9)**: `AGHAF_Status AGHAF_getDeviceCount(uint32_t* count);`
   - **Migration**:

```
// OLD (4.x):
uint32_t count = AGHAF_getDeviceCount();

// NEW (5.0.9):
uint32_t count = 0;
AGHAF_Status status = AGHAF_getDeviceCount(&count);
if (status != AGHAF_STATUS_OK) {
  // error handling
}
```

4. `AGHAF_getServiceVersion`
   - **Search**: Grep for calls to `AGHAF_getServiceVersion`.
   - **Signature Change**:
     - **Old (4.x)**: `uint32_t AGHAF_getServiceVersion(void);`
     - **New (5.0.9)**: `AGHAF_Status AGHAF_getServiceVersion(uint32_t* version);`
   - **Migration**:

```
// OLD (4.x):
uint32_t version = AGHAF_getServiceVersion();

// NEW (5.0.9):
uint32_t version = 0;
AGHAF_Status status = AGHAF_getServiceVersion(&version);
if (status != AGHAF_STATUS_OK) {
  // error handling
}
```

C. CAN Reception & Transmission Callbacks

In AGHAF CAN v2 (`aghaf_can_v2.h`), callback types `AGHAF_CAN_onMessageRx` and `AGHAF_CAN_onMessageTx` changed the parameter `dataLen` from `uint16_t` to `uint8_t`.

- **Search**:
  1. Grep for registrations: `AGHAF_CAN_v2_setOnMessageRxCallback` and `AGHAF_CAN_v2_setOnMessageTxCallback`.
  2. Grep for callback typedefs: `AGHAF_CAN_onMessageRx` and `AGHAF_CAN_onMessageTx`.
  3. Identify all function declarations and definitions passed as callbacks.
- **Signature Change**:
  - **Old (4.x)**:
    ```
    void onMessageRx(uint64_t timestamp, uint32_t ident, uint8_t flags,
                     uint16_t dataLen, uint8_t const* data, void* userContext);
    void onMessageTx(uint64_t timestamp, uint32_t ident, uint8_t flags,
                     uint16_t dataLen, uint8_t const* data, void* userContext);
    ```
  - **New (5.0.9)**:
    ```
    void onMessageRx(uint64_t timestamp, uint32_t ident, uint8_t flags,
                     uint8_t dataLen, uint8_t const* data, void* userContext);
    void onMessageTx(uint64_t timestamp, uint32_t ident, uint8_t flags,
                     uint8_t dataLen, uint8_t const* data, void* userContext);
    ```
- **Action**: Update both the function declarations (headers) and definitions (source files) of all registered CAN Rx and Tx callback functions, changing `uint16_t dataLen` to `uint8_t dataLen`.

---

3. Systematic Execution & Verification Procedure

1. **Search & Audit**: Run a global search for `AGHAF_` across the codebase to identify every call site and ensure no other modified APIs are missed:
   - `AGHAF_getDeviceBySN`
   - `AGHAF_getDeviceByHardID`
   - `AGHAF_getDeviceCount`
   - `AGHAF_getServiceVersion`
   - `AGHAF_CAN_v2_setOnMessageRxCallback`
   - `AGHAF_CAN_v2_setOnMessageTxCallback`
2. **Apply Code Edits**:
   - Update `main` with `AGHAF_initialize()` and `AGHAF_uninitialize()`.
   - Update device discovery calls to check for `AGHAF_STATUS_OK` and pass the handle pointer.
   - Update callback parameter types (`dataLen` -> `uint8_t`).
3. **Compilation Check**: Build the project to verify that all function signatures, return types, and callback pointer types match AGHAF 5.0.9 headers without compilation errors or warnings. “‘