**Project BookMark Documentation**

**Overview**

The Roaming Library (Project BookMark) is a portable file-sharing system built on NodeMCU ESP8266. It creates a wireless access point and serves a web interface for accessing digital documents and hosting discussions. The system is designed for local file sharing with a cyberpunk aesthetic.

**Hardware Requirements**

NodeMCU ESP8266

Micro SD Card Module

Micro SD Card (Max 32GB)

Power source (USB power bank or similar)

Required wiring/jumpers

**Wiring Configuration**

NodeMCU → SD Card Module

D5 (GPIO14) → SCK / CLK

D6 (GPIO12) → MISO

D7 (GPIO13) → MOSI

D8 (GPIO15) → CS

3.3V → VCC / 3.3v

GND → GND

**Features**

**1. Wireless Access Point**

No password required

Auto-redirects through captive portal

[If redirection is blocked by device, navigate to 192.168.4.1 in any browser]

Custom landing page with cyberpunk aesthetic

**2. File Management**

Supports multiple formats: PDF, EPUB, DOC, RTF, TXT, AZW, MOBI, LIB, FB2, PRC, PDB, and iBOOK

Files organized alphabetically

Collapsible file groups by first letter

File count per letter group

Upload functionality with supported formats

**3. Forum System**

Thread creation and management

Post creation within threads

1-hour auto-cleanup of forum content

Automatic thread refresh

**System Architecture**

**Core Components**

1. **Captive Portal System**

DNS and server initialization

Redirects all traffic to device IP

2. **File Management System**

File listing and categorization

Download handling

Type verification

3. **Upload System**

File validation

Storage management

Format verification

4. **Forum System**

Thread management

Post handling

JSON data storage

Automatic cleanup

**Data Flow**

Client → DNS/Captive Portal → Web Interface

File Upload → SD Card ← File Download

Forum Posts → Storage → Forum Display

**Installation Instructions**

**Required Libraries**

cppCopy

#include <ESP8266WiFi.h>

#include <ESP8266WebServer.h>

#include <DNSServer.h>

#include <SPI.h>

#include <SD.h>

#include <vector>

#include <map>

#include <algorithm>

#include <list>

#include <set>>

#include <espnow.h>

**Arduino IDE Setup**

1. Add ESP8266 Board Manager URL:

File → Preferences

Add: http://arduino.esp8266.com/stable/package\_esp8266com\_index.json Tools → Board → Boards Manager → Search "ESP8266"→ Install

2. Select Board Settings:

Board: "NodeMCU 1.0 (ESP-12E Module)"

Upload Speed: "115200"

CPU Frequency: "80 MHz"

Flash Size: "4M (1M SPIFFS)"

**SD Card Preparation**

On first boot, the firmware should implement the file structure on the sd card. If not, it should look like this:

1. Format SD card as FAT32

2. Create the directory structure:

Create main "/Alexandria" directory

Create alphabetical subdirectories "/Alexandria/A" through "/Alexandria/Z" Create "/Alexandria/0-9" for files starting with numbers

Create "/Alexandria/#@" for files starting with special characters 3. Ensure filenames only use standard characters

**Usage Guide**

**Initial Setup**

1. Power on the device

2. Look for WiFi network "B00KM4RK\_XXX" (where XXX is a random number) 3. Connect to network (no password required)

4. Captive portal will automatically open

If not, navigate to 192.168.4.1

**File Management**

1. Adding Files:

Power down device

Insert SD card into computer

Copy files to the appropriate folder within the "/Alexandria" directory: "/Alexandria/A" through "/Alexandria/Z" folders based on first letter "/Alexandria/0-9" for files starting with numbers

"/Alexandria/#@" for files starting with special characters

Reinsert SD card into module

2. Accessing Files:

Navigate to "74K3-4-F1L3"

Files are grouped by first letter

Click letter groups to expand/collapse

Click file name to download

3. Uploading Files:

Navigate to "L34V3-4-F1L3"

Click "Choose File"

Select File

Click "Upload"

**Forum Usage**

1. Creating Threads:

Click "Enter Forum"

Select "CREATE NEW THREAD"

Enter handle and content

Submit to create thread

2. Posting Replies:

Open desired thread

Scroll to bottom

Enter handle and reply

Posts appear immediately

**Troubleshooting Guide**

**Common Issues**

1. SD Card Not Detected

Verify wiring connections

Check SD card format (must be FAT32)

Try a different CS pin configuration

Ensure the SD card is fully inserted

2. WiFi Network Not Visible

Check power supply stability

Verify AP\_SSID configuration

Reset device

Check antenna connection

3. Files Not Displaying

Verify file formats are supported

Check file naming conventions

Ensure files are in the correct directory Verify structure and SD card permissions

4. Forum Posts Not Saving

Check SD card write permissions

Verify available storage space

Ensure proper JSON formatting

Check file path accessibility

**Customization Options**

**Network Configuration**

// Modify these values in the code

const char\* AP\_SSID\_\_BASE = "B00KM4RK\_"; // WiFi network name base

IPAddress apIP(192.168.4.1); // Device IP address

const byte DNS\_PORT = 53; // DNS port

**Forum Settings**

//// AAddjjuusstt cclleeaannuupp ttiimmiinngg ((iinn mmiilllliisseeccoonnddss))

Const unsigned long CLEANUP\_INTERVAL = 3600000; // Default: 1 hour

**Visual Customization**

1. Color Scheme

// Modify CSS colors in handleRoot()

"color: #0f0;" // Text color

"background-color: #000;" // Background

"text-shadow: 0 0 5px #0f0;" // Glow effect

2. ASCII Art

// owl ASCII art in handleRoot()

html += "<pre style='color:#0f0;text-align:center;line-height:1.2;margin:20px auto;font htmml += ",\_\_\_,\n";

htmml += " (O,O)\n";

htmml += " ( v )\n";

htmml += "-==\*^\*==-\n";

htmml += "</pre>";

**Advanced Configuration Options**

**Memory Management**

// Adjust these values based on your needs

#define MAX\_POSTS\_PER\_THREAD 100 // Limit posts per thread

#define MAX\_THREADS 50 // Limit total threads

#define MAX\_FILE\_SIZE 100000000 // Maximum file size in bytes

**Alternative Hardware Setup**

1. ESP32 Adaptation

Higher processing power

More memory available

Bluetooth capabilities

Different pin configuration required

2. Different SD Card Modules

Standard SD card module

Micro SD card shield

Built-in SD card module

Consider SPI speed requirements

**Power Management**

1. Power Requirements

Operating voltage: 5V via USB

Current draw: ~200mA average

Peak current: ~350mA during WiFi operations

2. Power Supply Options

USB power bank (recommended)

Wall adapter (5V, 1A minimum)

Computer USB port

Battery pack (with voltage regulation)

**Security Considerations**

1. Network Security

Open network (no password)

Local access only

No encryption of stored data

Consider physical security of device

2. Data Privacy

Forum posts are temporary (1-hour cleanup OR on reboot) Files accessible to all users

No user authentication

Local network isolation

**Future Enhancement Possibilities**

1. Feature Additions

User authentication system

File encryption options

Multiple language support

Search functionality

File previews

Download progress indicator

2. Hardware Expansions

Multiple SD card support

E-ink display integration

Battery monitoring

Status LEDs

Real-time clock

Temperature monitoring

**Code Maintenance Guidelines**

**Version Control**

1. Keep track of code versions

2. Document all changes

3. Backup configurations

4. Test before deploying changes

**Code Structure**

1. Main setup and loop

2. WiFi and server functions

3. File handling functions

4. Forum management

5. Helper utilities

**Best Practices**

1. Comment complex code sections 2. Use consistent naming conventions 3. Handle errors gracefully

4. Log important operations

5. Validate user inputs

6. Implement timeout handlers