# MQTT Application Documentation

## Overview

This application is a Python-based MQTT client that subscribes to multiple topics and processes messages from different devices (D1, D2, D3). The system includes logging capabilities, message processing, and deployment management.

## Components

### 1. Main Application (main.py)

The entry point of the application that initializes logging and starts the MQTT handler.

### 2. MQTT Handler (mqtt\_handler.py)

* Manages MQTT client connections
* Connects to broker.hivemq.com on port 1883
* Subscribes to topics in the format TopicX/DY where:
  + X ranges from 1 to 10
  + Y is either D1, D2, or D3
* Stores message data with timestamps
* Handles connection and message processing errors

### 3. Message Processor (message\_processor.py)

* Processes messages based on device type (D1, D2, D3)
* Implements separate handling logic for each device type
* Includes error handling and logging

### 4. Logger Configuration (logger\_config.py)

* Implements colored console logging
* Maintains rotating log files (max 5MB per file, 3 backup files)
* Log levels:
  + Console: INFO and above
  + File: DEBUG and above
* Timestamps all log entries

### 5. Deployment Script (deploy.py)

* Manages code deployment
* Handles repository cloning and updates
* Implements automated code pulling
* Manages application startup

## Dependencies

* paho-mqtt >= 2.1.0
* Python >= 3.11

## Running the Application

The application can be started using:

python main.py

## Logging

Logs are stored in debug.log with the following format: - Console: HH:MM:SS [LEVEL] MESSAGE - File: YYYY-MM-DD HH:MM:SS [LEVEL] FILENAME:LINE - MESSAGE

## Error Handling

* Connection failures are logged and raised
* Message processing errors are caught and logged
* Deployment errors include fallback mechanisms

## Security Considerations

* MQTT broker connection uses default port 1883
* No authentication currently implemented
* Messages are stored in memory during runtime