Introduction

A logger is a tool that helps you record messages from your program. You can use a logger to track errors, debug issues, or monitor performance. A logger usually has different levels of severity, such as info, warning, error, or fatal. You can set the level of your logger to filter out less important messages.

The Logger provides interfaces for applications to publish logging information on selected logging medium. Each of the provided logging information has its own severity level. Logging level can be changed at runtime.

Behavior

1. To be able to distinguish the logs of different application instances within a system.
2. Automatic timestamping.
3. Log to console or to file (extra: log to remote network via UDP socket).
4. Trace number of logs per application.
5. Log for builtin types (bool, uint8_t, uint16_t, uint32_t, int8_t, int16_t, int32_t, float, double).
6. Use an ostream-like syntax to write log messages.

Configuration

1. Application ID: string[5]
2. The default log level: enum LogLevel
3. The log mode: enum LogMode

Table 1: Logging levels.

lOff	No logging
lError	Error with impact to correct functionality
lWarn	Warning if correct behavior cannot be ensured
lInfo	Informational, providing high level understanding
lDebug	Detailed information for programmers

Table 2: Logging modes

mFile	Save to file
mConsole	Forward to console
mRemote	Send remotely

Log format

| TS(hh:mm:ss) |AppId | LogNum | LogLevel | LogMsg |

| 12:00:01 | ATEST | 5 | lInfo | HelloWorld! |

| 12:00:02 | ATEST | 6 | lInfo | Variable = 30 |

| 12:00:03 | BTEST | 1 | lError | App B Error |

Evaluation Criteria

Two applications will be created and will log simultaneously. Both applications will call APIs for logging, changing log level and mode.

Extra credit #1

Log to an external UDP server, for which the log user has to provide IP and port during Logger creation (default 127.0.0.1 : 8893).

Extra credit #2

Make the Logger threadsafe and allow an application with two threads to use the logger, each thread with different AppID.

APIs

CppLogger(String appId, LogLevel logLevel, LogMode logMode, …)

SetLogLevel(String appId, LogLevel logLevel)

SetLogMode(String appId, LogMode logMode, …)