

Logging

Anforderungen

1. Loggen von Key-Value-Paaren (Key: 2 Byte, Value: 4 Byte).
2. Loggen von Text-Nachrichten (variable Länge).

API

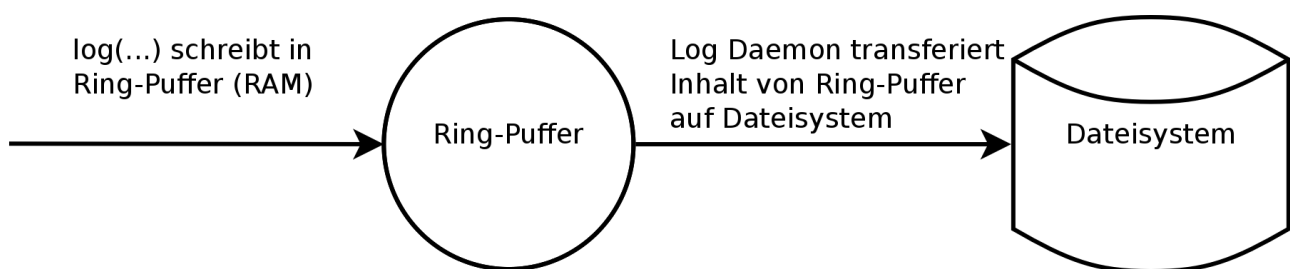
- `bool log(uint_16 key, float value)`
- `bool log(log_level_t log_level, const char *message)`

Beide Methoden arbeiten nicht-blockierend.

`log(...)`-Methoden schreiben binär in Ring-Puffer.

- `log(uint_16 key, float value)` erzeugt folgenden Eintrag:
[entry_type][timestamp][key][value]
- `log(log_level_t log_level, const char *message)` erzeugt folgenden Eintrag:
[entry_type][timestamp][thread_id][log_level][message]

Architektur



`log(...)`

- Thread-safe: Zugriff auf Ring-Puffer durch Semaphor geschützt.
- Weckt Log Daemon auf: Durch Semaphor.
- Ring-Puffer wird **nicht** überschrieben: `log(...)` schreibt erst wieder in Ring-Puffer, wenn dieser durch Log Daemon ausgelesen wurde.

Log Daemon

- Schreibt in eine Datei:
 - Binär-Datei wird zuerst nach „Entry-Types“ gefiltert. Unterschiedliche Entry-Types landen in unterschiedlichen Dateien.
 - Dann Konvertierung der Binär-Dateien nach ASCII.