Program Listing for File realtime_clock.hpp

Return to documentation for file (include/holoscan/core/resources/gxf/realtime_clock.hpp)

```
/* * SPDX-FileCopyrightText: Copyright (c) 2023-2024 NVIDIA CORPORATION & AFFILIATES. All rights reserved. * SPDX-License-Identifier: Apache-2.0 * * Licensed under the Apache License, Version 2.0 (the "License"); * you may not use this file except in compliance with the License. * You may obtain a copy of the License at * * http://www.apache.org/licenses/LICENSE-2.0 * * Unless required by applicable law or agreed to in writing, software * distributed under the License is distributed on an "AS IS" BASIS, * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. * See the License for the specific language governing permissions and * limitations under the License. */ #ifndef HOLOSCAN_CORE_RESOURCES_GXF_REALTIME_CLOCK_HPP #define HOLOSCAN_CORE_RESOURCES_GXF_REALTIME_CLOCK_HPP #include <chrono> #include <string> #include "./clock.hpp" namespace holoscan { class RealtimeClock : public Clock { public: HOLOSCAN_RESOURCE_FORWARD_ARGS_SUPER(RealtimeClock, Clock) RealtimeClock() = default; RealtimeClock(const std::string& name, nvidia::gxf::RealtimeClock* component); const char* gxf_typename() const override { return "nvidia::gxf::RealtimeClock"; } void setup(ComponentSpec& spec); double time() const override; int64_t timestamp() const override; void sleep_for(int64_t duration_ns) override; void sleep_until(int64_t target_time_ns) override; void set_time_scale(double time_scale); nvidia::gxf::RealtimeClock* get() const; private: Parameter<double> initial_time_offset_; Parameter<double> initial_time_scale_; Parameter<bool> use_time_since_epoch_; }; } // namespace holoscan #endif/* HOLOSCAN_CORE_RESOURCES_GXF_REALTIME_CLOCK_HPP */
```