#include "LDR.hpp"

constexpr const char* LDRSensor::TAG;

LDRSensor::LDRSensor(adc1_channel_t channel, gpio_num_t pin, uint32_t read_interval_ms): adc_channel_(channel), gpio_pin_(pin), task_handle_(nullptr), read_interval_ms_(read_interval_ms), running_(false) { adc1_config_width(ADC_WIDTH_BIT_12); // Resolutie 12 bits adc1_config_channel_atten(adc_channel_, ADC_ATTEN_DB_11); }

void LDRSensor::start() { if (!running_) { running_ = true; xTaskCreatePinnedToCore(taskWrapper, "ldr_task", 2048, this, 5, &task_handle_, tskNO_AFFINITY); } }

// void LDRSensor::stop() { // if (running) { // running_ = false; // if (taskhandle != nullptr) { // vTaskDelete(taskhandle); // taskhandle = nullptr; // } // } // }

void LDRSensor::run() { while(running_) { int val = adc1_get_raw(adc_channel_);

```
ESP_LOGI(TAG, "LDR waarde: %d", val);

vTaskDelay(read_interval_ms_ / portTICK_PERIOD_MS);
}
```

}

void LDRSensor::taskWrapper(void* pvParameters) { LDRSensor* sensor = static_cast<LDRSensor*> (pvParameters); sensor->run(); }