

# Programação de Periféricos - Trabalho 2

## Objetivo

O segundo trabalho consiste no desenvolvimento de uma aplicação embarcada controlada pelo UCX/OS RTOS. Essa aplicação será composta por um conjunto de sensores e atuadores. Os diferentes componentes da aplicação serão implementados em múltiplas threads e a coordenação dos componentes será orquestrada pelas primitivas de sincronização do RTOS (*mutex* e *pipes* / *message queues*). As seguintes tarefas deverão fazer parte de sua aplicação:

- Thread 1 - sensor de temperatura;
- Thread 2 - sensor de luminosidade;
- Thread 3 - controle de temperatura (atuador);
- Thread 4 - controle de dimerização (atuador);
- Thread 5 - gerenciamento da aplicação (controle geral);
- Thread 6 - depuração (impressão de mensagens pela UART).

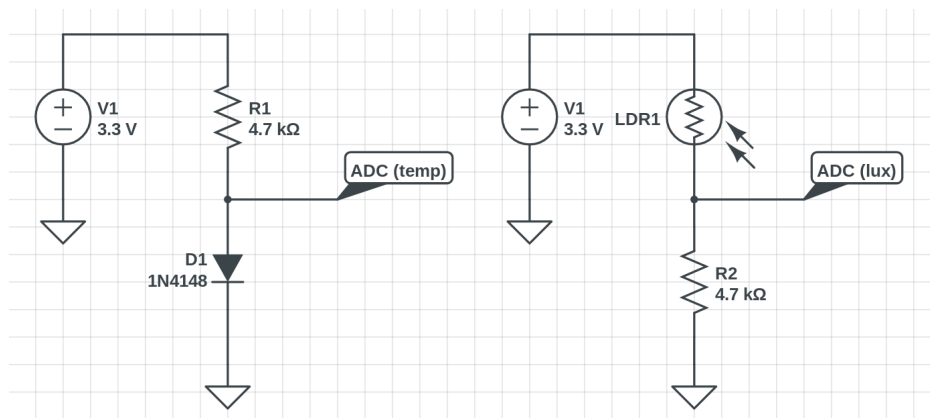
## Funcionamento

As tarefas da aplicação possuem funcionalidades específicas, que deverão ser implementadas de acordo com o seu papel na aplicação e sincronizadas em função do compartilhamento de recursos. Por exemplo, as tarefas 1 e 2 são responsáveis por realizar a leitura de sensores por meio de um conversor ADC, portanto devem acessar o recurso compartilhado de forma mutualmente exclusiva, devendo ser utilizado um *mutex* para isso. As tarefas 3 e 4 são responsáveis por configurar atuadores por meio de um timer em modo PWM, portanto devem acessar os diferentes canais do mesmo de forma mutualmente exclusiva. No gerenciamento da aplicação encontra-se a tarefa 5, que será responsável por receber dados dos sensores e enviar dados para os atuadores por meio de canais de comunicação do tipo *pipe* ou *message queue*.

Essa tarefa também poderá gerar mensagens de log e depuração, que deverão ser enviadas por meio de um *pipe* ou *message queue* para a tarefa 6.

## Sensores

Os sensores serão implementados de forma simples, sendo o primeiro um sensor de temperatura e o segundo um sensor de luminosidade, apresentados de acordo com o esquemático abaixo:



Obviamente o ideal seria utilizarmos sensores específicos para esse propósito. Na versão proposta, temos uma aproximação do comportamento de um sensor comercial, porém sua implementação simplificada é o suficiente para o desenvolvimento do trabalho. Se o grupo desejar, o sensor de temperatura poderá ser substituído por um outro, como DHT11, DHT22 ou LM35, por exemplo.

## Atuadores

Os atuadores serão implementados de forma simples, sendo o primeiro um indicativo da temperatura, composto por um resistor de 470 ohms e um LED. Esse atuador deverá indicar temperaturas entre 0 e 40 graus representando essa faixa de temperaturas por meio do brilho do LED. Temperaturas abaixo de zero deverão manter o LED desligado e acima de 40 graus ligado com brilho total. O segundo atuador será um indicativo de nível de dimerização de uma luminária, composto por um resistor de 470 ohms e um LED. O

controle de dimerização deverá atuar na faixa de 100 a 10 lux, sendo que valores acima de 100 lux deverão manter o LED desligado e abaixo de 10 lux ligado com brilho total.

Novamente, temos aqui uma simplificação da implementação de atuadores. A saída do controle de temperatura poderia ser um *driver* para um motor DC (como uma ventoinha) e a saída do o controle de dimerização poderia ser um circuito de driver para uma luminária do tipo 0-10V.

### **Template da aplicação e calibração dos sensores**

Juntamente com o enunciado estão sendo disponibilizados os recursos abaixo listados. Esses recursos são partes de sua aplicação, que deverão ser compiladas juntamente com o UCX/OS e posteriormente carregadas na placa Blackpill. Para isso, crie uma nova regra no *Makefile* principal do RTOS, adicionando os códigos fonte.

Os sensores são bastante simples e necessitam de calibração em função dos componentes utilizados pelo grupo. Sugere-se utilizar os valores nominais dos parâmetros definidos no template, e comparar com valores aferidos por um sensor de luminosidade e de temperatura (você possui acesso a esses sensores no seu celular). Caso seja necessário ajustar o sensor de temperatura, realize a medição do diodo utilizado com um multímetro, e ajuste o parâmetro F\_VOLTAGE de acordo. No caso do sensor de luminosidade, pode-se ajustar o parâmetro REF\_RESISTANCE.

### **Entrega**

Este trabalho deverá ser realizado em duplas ou individualmente e apresentado em aula no dia 28/10 (apresentação em torno de 10 minutos). Para a entrega, é esperado que apenas um dos integrantes envie pelo Moodle um arquivo *.tar.gz*, contendo:

1. Um diagrama simplificado do circuito implementado. Esse diagrama pode ser feito em qualquer software, como CircuitLab por exemplo;
2. Código fonte desenvolvido.